Asymptotically Optimal Online Page Migration on Three Points

Algorithmica manuscript No.
(will be inserted by the editor)

Asymptotically Optimal Online Page Migration on Three Points

Akira Matsubayashi

Received: date / Accepted: date

Abstract This paper addresses the page migration problem: given online requests from nodes on a network for accessing a page stored in a node, output online migrations of the page. Serving a request costs the distance between the request and the page, and migrating the page costs the migration distance multiplied by the page size $D \geq 1$. The objective is to minimize the total sum of service costs and migration costs. Black and Sleator conjectured that there exists a 3-competitive deterministic algorithm for every graph. Although the conjecture was disproved for the case $D = 1$, whether or not an asymptotically (with respect to D) 3-competitive deterministic algorithm exists for every graph is still open. In fact, we did not know if there exists a 3-competitive deterministic algorithm for an extreme case of three nodes with $D \geq 2$. As the first step toward an asymptotic version of the Black and Sleator conjecture, we present 3- and $(3+1/D)$-competitive algorithms on three nodes with $D = 2$ and $D \geq 3$, respectively, and a lower bound of $3+\Omega(1/D)$ that is greater than 3 for every $D \geq 3$. In addition to the results on three nodes, we also derive ρ-competitiveness on complete graphs with edge-weights between 1 and $2-2/\rho$ for any $\rho \geq 3$, extending the previous 3-competitive algorithm on uniform networks.

Keywords page migration · work function algorithm · competitive analysis · server problem

1 Introduction

The problem of computing an efficient dynamic allocation of data objects stored in nodes of a network commonly arises in network applications such as memory management in a shared memory multiprocessor system and Peer-to-Peer applications on

A preliminary version appeared in the proceedings of the 10th Workshop on Approximation and Online Algorithms (WAOA 2012).

A. Matsubayashi
Division of Electrical Engineering and Computer Science, Kanazawa Univ., Kanazawa 920-1192, Japan
E-mail: mbayashi@t.kanazawa-u.ac.jp

the Internet. In this paper, we study one of the classical varieties of the problem, the *page migration problem*, in which a request issued on a node for accessing a single data object (called a *page* in this problem) must be served using unicast communication. After serving each request, we are allowed to migrate the page. Serving a request costs the distance of the communication, and migrating the page costs the migration distance multiplied by the page size $D \geq 1$. The objective is to minimize the total sum of the service and migration costs. The page migration problem has been extensively studied (e.g., [2–4, 8, 10, 13, 15]) and generalized to several settings such as *k*-page migration [3], file allocation problem, e.g., [2, 4, 13], and data management on dynamic networks, e.g, [1, 7]. See [6] for a recent survey.

1.1 Related Results

We focus on deterministic online page migration algorithms. Black and Sleator [8] first studied competitive analysis of the page migration problem and presented 3-competitive deterministic algorithms on trees, uniform networks, and Cartesian products of these networks, including grids and hypercubes. These algorithms are optimal because the deterministic lower bound is 3 for every network with at least two nodes [8, 11]. Black and Sleator conjectured that there exists a 3-competitive deterministic algorithm for every network. The first upper bound of 7 for general networks was given by Awerbuch, Bartal, and Fiat [2] and improved to 4.086 by Bartal, Charikar, and Indyk [3]. For a special case of $D = 1$, a better bound of $2 + \sqrt{2}$ is achievable [14]. For a yet restricted case of $D = 1$ and three nodes, a 3-competitive deterministic algorithm was presented in [10]. Whether or not a 3-competitive deterministic algorithm exists on three nodes for $D \geq 2$ was left open. Concerning the lower bound, Black and Sleator's conjecture was disproved by Chrobak, Larmore, Reingold, and Westbrook [10], who proved that no deterministic algorithm has the competitive ratio less than $85/27 \approx 3.148$ on special networks with $D = 1$. This bound was refined to 3.164 [14]. It is mentioned in [10] that the lower bound is larger than 3 even on four nodes. An explicit lower bound of 3.121 on five nodes was proved in [14].

1.2 Contributions of This Paper

All the previous lower bounds larger than 3 were proved only for the case $D = 1$. Therefore, an asymptotic version of the Black and Sleator conjecture with respect to D, i.e., whether or not an asymptotically 3-competitive deterministic algorithm on every network exists is still open. As the first step toward an answer for this conjecture, we present

- a $(3 + 1/D)$-competitive algorithm on three nodes with $D \geq 3$,
- a 3-competitive algorithm on three nodes with $D \leq 2$, and
- a lower bound of $3 + \Omega(1/D)$ that is greater than 3 for every $D \geq 3$.

These results thoroughly answer the open question of existence of a 3-competitive algorithm on three nodes. A summary of the results is provided in Table 1. In addition to the results on three nodes, we also derive

Table 1 Summary of Results on Three Nodes

Page size D	Upper bound	Lower bound
1	3 [10] *	3 [8]
2	3^*	3 [8]
≥ 3	$3+1/D^*$	$3+\Omega(1/D)^*$

* This paper

- ρ-competitiveness on complete graphs (of arbitrary size) with edge-weights between 1 and $2-2/\rho$ for any $\rho \geq 3$,

extending the previous 3-competitive algorithm on uniform networks [8].

1.3 Overview of Technical Ideas

Our $(3+1/D)$-competitive algorithm is a typical work function algorithm similar to algorithms for metrical task systems, e.g., [9], and k-server problems [5,12]. In general, a work function algorithm makes online decisions using information on the optimal offline cost for processing requests that have been issued so far and ending at each configuration (page node in the page migration problem). The optimal offline cost function with respect to configurations is called a work function. To prove that a work function (i.e., optimal cost) increases enough, we introduce a probably new technique of analytically dealing with the work function extended on a continuous network. In Sect. 3, we bound an extended work function from below using its derivatives. The author believes that such analysis is the technical contribution of this paper.

Since the competitive ratio on three nodes is not monotonic with respect to D, it appears to be reasonable that we need different approaches for $D = 2$ and $D \geq 3$. Our 3-competitive algorithm for $D = 2$ is based on the counter-based algorithm for uniform networks [8], which maintains a counter on each node. The counters are updated every time a request arrives so that they represent a tendency of migration. If a counter reaches a certain value, then the algorithm moves the page to the node with this counter. One can observe that the original algorithm is 3-competitive even on a complete graph with roughly the same edge-weights, and that this can be generalized to any $\rho \geq 3$. More specifically, there is a "triangle" condition on edge-weights around the page such that the original potential function used in [8] can amortize the service costs and the next migration cost. If there are three nodes, then at least one "good" node satisfies the condition. We design our algorithm by modifying the original algorithm for the page at a "bad" node. Although the modification wastes the "deposit" even worse when leaving the bad node, we can prove through careful observations that much more deposit can be saved after the possible migration to a good node or from services before the migration. The formal proof is presented in Sect. 4.

Our lower bound is based on the following observation: If there are only two nodes, then any 3-competitive algorithm must move after exactly $2D$ requests issued by a cruel adversary, which always issues a request from the other node than the

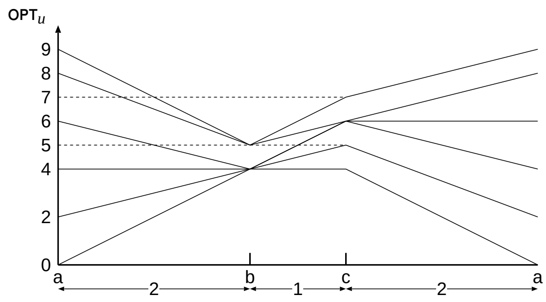

Fig. 1 Example of work functions on three nodes a, b, and c with $d_{ab} = d_{ac} = 2$ and $d_{bc} = 1$. We assume that the page of size $D = 2$ is located at a initially, and that requests are issued at b, b, b, c, and b

online page. If the adversary carefully adds a new node close to the existent request node and divides the $2D$ requests among these nodes, then no matter when or where the algorithm moves, it is too "impatient" or "tardy" to achieve the competitive ratio of 3. We explicitly design the adversary and analyze the lower bound in Sect. 5. We also demonstrate that an explicit lower bound of $3 + \frac{1}{360D+347}$ for $D \geq 3$ can be derived from our proof.

2 Preliminaries

The page migration problem can be formulated as follows: given an undirected graph $G = (V, E)$ with edge weights, $s_0, r_1, \ldots, r_k \in V$, and a positive integer D, compute $s_1, \ldots, s_k \in V$ so that the cost function $\sum_{i=1}^{k}(d_{s_{i-1}r_i} + Dd_{s_{i-1}s_i})$ is minimized, where d_{uv} is the distance between nodes u and v on G. The terms $d_{s_{i-1}r_i}$ and $Dd_{s_{i-1}s_i}$ represent the cost to serve the request from r_i by the node s_{i-1} holding the page and the cost to migrate the page from s_{i-1} to s_i, respectively. We call s_i and r_i a *server* and a *client*, respectively. An *online* page migration algorithm determines s_i without information of $r_{i+1}, \ldots, r_k$. We denote by $A(\sigma)$ the cost of a page migration algorithm A for a sequence $\sigma := r_1 \cdots r_k$. A deterministic online page migration algorithm ALG is *ρ-competitive* if there exists a constant value α such that $\text{ALG}(\sigma) \leq \rho \cdot \text{OPT}(\sigma) + \alpha$ for any σ, where OPT is an optimal offline algorithm. We denote by $\text{OPT}_u(\sigma)$, called a *work function*, the minimum (offline) cost to process σ so that $s_k = u$. Obviously, $\text{OPT}(\sigma) = \min_{u \in V}\{\text{OPT}_u(\sigma)\}$. An online algorithm that determines the server position after processing σ using the information of $\text{OPT}_u(\sigma)$ for all possible nodes u is called a *work function algorithm*. Note that $\text{OPT}_u(\sigma)$ can be computed using dynamic programming, i.e., for a request issued at r after σ, $\text{OPT}_u(\sigma r) = \min_{v \in V}\{\text{OPT}_v(\sigma) + d_{rv} + Dd_{uv}\}$ and $\text{OPT}_u(\emptyset) = Dd_{s_0 u}$ [10], where $\emptyset$ denotes an empty sequence. An example of work functions are illustrated in Fig. 1. For a node u and $k \geq 1$, we write a sequence consisting of k repetitions of u as u^k. Unless otherwise stated, we suppose that graphs considered here have a node set $V := \{a, b, c\}$ and edge weights $x := d_{ab}$, $y := d_{ac}$, and $z := d_{bc}$ for edges (a, b),

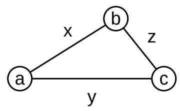

Fig. 2 Labels for nodes and edges of 3-node graphs

(a,c), and (b,c), respectively (Fig. 2). We denote $L := x+y+z$ and assume that $\max\{x,y,z\} < L/2$.

3 $(3+1/D)$-Competitive Algorithm

We consider a typical work function algorithm denoted by WFA, which moves the server located at s after processing a sequence σ of clients, to a nearest node among nodes v minimizing $\mathrm{OPT}_v(\sigma)+d_{rv}+Dd_{sv}$ after servicing a new request on r. By this definition, the destination $\hat{s}$ of the migration satisfies $\mathrm{OPT}_s(\sigma r) = \mathrm{OPT}_{\hat{s}}(\sigma)+d_{r\hat{s}}+Dd_{s\hat{s}}$. Another way of understanding the algorithm is that WFA moves the server s to $\hat{s}$ when a decline of slope D from s to $\hat{s}$ appears on the work function, i.e., $\mathrm{OPT}_s(\sigma r) - \mathrm{OPT}_{\hat{s}}(\sigma r) = Dd_{s\hat{s}}$, except when s is one of the nodes v minimizing $\mathrm{OPT}_v(\sigma)+d_{rv}+Dd_{sv}$. In Fig. 1, for example, the server initially located at a is moved to b after the last request on b. The purpose of considering such a decline on the work function as a trigger of migration is to avoid requests on $\hat{s}$ that would increase online service cost at the server s but change neither OPT_s nor $\mathrm{OPT}_{\hat{s}}$. A similar idea is used for other work function algorithms ([9,5,12]). We prove the following theorem:

Theorem 1 WFA *is* $(3+1/D)$*-competitive on three nodes.*

Our proof of Theorem 1 is divided into two parts, deriving a sufficient condition for Theorem 1 and proving the condition. In the rest of this section, we suppose that WFA locates the server on s after processing σ, and that a request is issued at $r \in V$ after σ. For a function f of σ, we use the notations $f = f(\sigma)$ and $f' = f(\sigma r)$ for simplicity.

3.1 Sufficient Condition for Theorem 1

We claim that the condition

$$Dd_{\hat{s}u} + M' \le \mathrm{OPT}'_u \text{ for any } u \in V \tag{1}$$

implies Theorem 1, where $\hat{s}$ is the server of WFA after processing σr, and $M' = M(\sigma r)$ is D times the total sum of migration distances of WFA in processing σr.

Because $|\mathrm{OPT}_u - \mathrm{OPT}_v| \le Dd_{uv}$ for any $u,v \in V$ [10], it follows that

$$\mathrm{OPT}'_s = \mathrm{OPT}_{\hat{s}} + d_{r\hat{s}} + Dd_{s\hat{s}} \ge \mathrm{OPT}_s + d_{r\hat{s}}, \text{ and} \tag{2}$$

$$\mathrm{OPT}'_s \le \mathrm{OPT}'_{\hat{s}} + Dd_{s\hat{s}}. \tag{3}$$

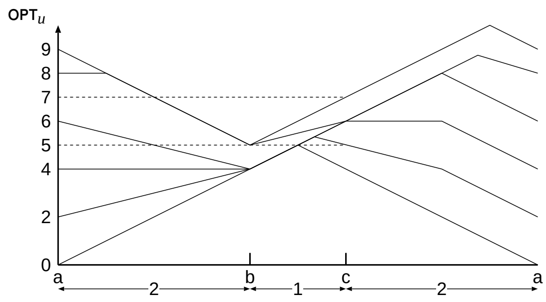

Fig. 3 Extended work functions on the same assumptions as those in Fig. 1

It follows from (2) and (3) that $d_{r\hat{s}} \le \text{OPT}'_{\hat{s}} - \text{OPT}_s + Dd_{s\hat{s}}$. Therefore, we have

$$\text{WFA}' - \text{WFA} = d_{rs} + Dd_{s\hat{s}} \le d_{r\hat{s}} + (D+1)d_{s\hat{s}} \le \text{OPT}'_{\hat{s}} - \text{OPT}_s + (2D+1)d_{s\hat{s}}. \quad (4)$$

By summing (4) overall requests in σr, we obtain $\text{WFA}' \le \text{OPT}'_{\hat{s}} + (2+1/D)M'$. Hence, if (1) is satisfied, then by choosing u minimizing OPT'_u, we have $\text{WFA}' \le \text{OPT}'_{\hat{s}} + (2+1/D)\text{OPT}' - (2D+1)d_{\hat{s}u} \le (3+1/D)\text{OPT}' - (D+1)d_{\hat{s}u}$, which completes the proof of Theorem 1.

3.2 Proof of Sufficient Condition

To prove (1), we generalize the network to a continuous loop[1] R of length L containing a, b, and c with the preserved distances. Specifically, we define R as an interval $\{p \mid 0 \le p < L\}$ modulo L, i.e., any real number p is equivalent to $p - \lfloor p/L \rfloor \cdot L$. We define an extended work function at a point $p \in R$ as

$$w'_p := \min_{q \in R}\{w_q + d_{rq} + Dd_{pq}\} \text{ and } w_p(\emptyset) := Dd_{s_0 p}.$$

An example of extended work functions are illustrated in Fig. 3. One of the important properties of extended work functions is that $\hat{p} \in V$ for any $p \in R$ with $\hat{p} \ne p$, where $\hat{p}$ is a nearest point to $p \in R$ among points $q \in R$ minimizing $w_q + d_{rq} + Dd_{pq}$. This implies that $w'_p = \min_{q \in V \cup \{p\}}\{w_q + d_{rq} + Dd_{pq}\}$, and hence, $w_u = \text{OPT}_u$ for any $u \in V$. Another property is that one-sided derivatives at any point are integers between $-D$ and D. These properties will formally be proved later in Lemma 5.

We denote the farthest point of p on R by $\bar{p}$. For $p, q \in R$, we define $[p,q]$ as the closed interval of length d_{pq} between p and q on R if $d_{pq} < L/2$. If $d_{pq} = L/2$, then we define $[p,q]$ as the whole set R, not an interval between p and q. Notations $(p,q]$, $[p,q)$, and (p,q) are used to denote the intervals obtained from $[p,q]$ by excluding p, q, and both p and q, respectively. Lemmas 1–4 below state basic properties of w_p that will be used in the subsequent lemmas.

[1] One might expect that a continuous tree instead of a continuous loop would be preferable in terms of scalability of the network. However, this idea would fail because such a tree has the center, i.e., a point near to three nodes, which makes a work function extended on the continuous tree smaller than the original work function at some nodes.

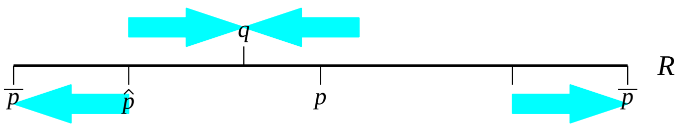

Fig. 4 Range in which $\hat{q}$ may exist on R. Upper and lower arrows represent $d_{q\hat{q}} \leq d_{q\hat{p}}$ and $d_{p\hat{p}} \leq d_{p\hat{q}}$, respectively

Lemma 1 *For any $p, q \in R$, it follows that $w_q - w_p \leq Dd_{pq}$.*

Proof The lemma clearly holds if $\sigma = \emptyset$. Otherwise, it follows from the minimality of w'_q that $w'_q \leq w_{\hat{p}} + d_{r\hat{p}} + Dd_{q\hat{p}} = w'_p - Dd_{p\hat{p}} + Dd_{q\hat{p}} \leq w'_p + Dd_{pq}$. □

Lemma 2 *For any $p \in R$ and $q \in (p, \hat{p}]$, it follows that $\hat{q} = \hat{p}$.*

Proof It follows from the minimality of w'_p that

$$w'_p = w_{\hat{p}} + d_{r\hat{p}} + Dd_{p\hat{p}} \leq w_{\hat{q}} + d_{r\hat{q}} + Dd_{p\hat{q}}. \tag{5}$$

Substituting $d_{p\hat{p}} = d_{pq} + d_{q\hat{p}}$, we obtain

$$w_{\hat{p}} + d_{r\hat{p}} + Dd_{q\hat{p}} \leq w_{\hat{q}} + d_{r\hat{q}} + D(d_{p\hat{q}} - d_{pq}) \leq w_{\hat{q}} + d_{r\hat{q}} + Dd_{q\hat{q}} = w'_q. \tag{6}$$

By the minimality of w'_q, (6) holds with equality. This means that (5) also holds with equality. Therefore, $\hat{p}$ minimizes $w_{\hat{p}} + d_{r\hat{p}} + Dd_{q\hat{p}}$ (i.e., $\hat{p} \in \arg\min_{t \in R}\{w_t + d_{rt} + Dd_{qt}\}$), and $\hat{q}$ minimizes $w_{\hat{q}} + d_{r\hat{q}} + Dd_{p\hat{q}}$ (i.e., $\hat{q} \in \arg\min_{t \in R}\{w_t + d_{rt} + Dd_{pt}\}$). By the minimalities of $d_{q\hat{q}}$ and $d_{p\hat{p}}$, it follows that $d_{q\hat{q}} \leq d_{q\hat{p}}$ and $d_{p\hat{p}} \leq d_{p\hat{q}}$. Because $q \in (p, \hat{p}]$, $\hat{q}$ exists only at $\hat{p}$ (Fig. 4). □

Lemma 3 *For any $p \in R$ and $q \in [p, \hat{p})$, it follows that $w_q - w_{\hat{p}} > (D-1)d_{\hat{p}q}$.*

Proof Because q is nearer to p than $\hat{p}$ is, it follows that $w_{\hat{p}} + d_{r\hat{p}} + Dd_{p\hat{p}} < w_q + d_{rq} + Dd_{pq}$. Thus, because $d_{p\hat{p}} = d_{pq} + d_{q\hat{p}}$, we have $w_q - w_{\hat{p}} > d_{r\hat{p}} - d_{rq} + D(d_{p\hat{p}} - d_{pq}) \geq (D-1)d_{\hat{p}q}$. □

Lemma 4 *For any $p \in R$ and $q \in [r, \hat{p}]$, it follows that $w_{\hat{p}} - w_q \leq (D-1)d_{\hat{p}q}$.*

Proof It follows from the minimality of w'_p that $w_{\hat{p}} + d_{r\hat{p}} + Dd_{p\hat{p}} \leq w_q + d_{rq} + Dd_{pq}$. Thus, because $d_{r\hat{p}} = d_{rq} + d_{q\hat{p}}$, we have $w_{\hat{p}} - w_q \leq d_{rq} - d_{r\hat{p}} + D(d_{pq} - d_{p\hat{p}}) \leq (D-1)d_{\hat{p}q}$. □

To prove (1), we utilize a relation between the increased amount of the work function and its one-sided derivatives, which are defined as

$$m_{p-0} := \lim_{q \to p-0} \frac{w_q - w_p}{d_{pq}} \text{ and } m_{p+0} := \lim_{q \to p+0} \frac{w_q - w_p}{d_{pq}} \text{ for any } p \in R.$$

It should be noted that m_{p-0} is a negated value of standard one-sided derivative. The following lemma guarantees that $w_u = \mathrm{OPT}_u$ for any $u \in V$, the derivatives exist and are integers, and that w_p can be strictly convex only on an interval containing a node of V.

Lemma 5 *The following claims hold.*

1. *For any $p \in R$ with $\hat{p} \neq p$, it follows that $\hat{p} \in V$.*
2. *For any $p \in R$, m_{p-0} and m_{p+0} are integers with $-D \leq m_{p\pm 0} \leq D$.*
3. *For any $p \in R \setminus V$, it follows that $m_{p-0} + m_{p+0} \leq 0$, i.e., w_p is concave on any interval not containing a node in V.*

Proof We prove the lemma by induction on σ. If $\sigma = \emptyset$, then $m_{s_0-0} = m_{s_0+0} = D$, $m_{\bar{s_0}-0} = m_{\bar{s_0}+0} = -D$, and $\{m_{p-0}, m_{p+0}\} = \{-D, D\}$ for $p \in R \setminus \{s_0, \bar{s_0}\}$. These equations imply Claims 2 and 3. Assume that Claims 2 and 3 hold for a sequence σ.

We first prove Claim 1 for σ. Let $p \in R$ with $\hat{p} \neq p$. The claim is immediate if $\hat{p} = r$. We assume $\hat{p} \neq r$. Let $q_1 \in (p, \hat{p})$ and $q_2 \in (r, \hat{p})$. It follows that $r \notin (p, \hat{p})$, for otherwise, by Lemma 1 and $d_{p\hat{p}} = d_{pr} + d_{r\hat{p}}$, we have $w'_p = w_{\hat{p}} + d_{r\hat{p}} + Dd_{p\hat{p}} > w_r - Dd_{r\hat{p}} + Dd_{p\hat{p}} = w_r + Dd_{pr}$, contradicting the minimality of w'_p. Therefore, we have $\hat{p} \in (q_1, q_2)$. Thus, by Lemmas 3 and 4 we have

$$m_{\hat{p}-0} + m_{\hat{p}+0} = \lim_{q_1 \to \hat{p}} \frac{w_{q_1} - w_{\hat{p}}}{d_{\hat{p}q_1}} + \lim_{q_2 \to \hat{p}} \frac{w_{q_2} - w_{\hat{p}}}{d_{\hat{p}q_2}} > (D-1) - (D-1) = 0.$$

By Claim 3 of induction hypothesis, this means $\hat{p} \notin R \setminus V$, and hence, $\hat{p} \in V$.

We then prove Claim 2 for σr. I.e., we prove that for any $p \in R$, $\lim_{q \to p}(w'_q - w'_p)/d_{pq}$ is an integer in $[-D, D]$. By Lemma 2, if $p \neq \hat{p}$, then any point $q \in (p, \hat{p})$ has $\hat{q}$ with $q \neq \hat{q} = \hat{p}$. Therefore, $I := \{q \in R \mid q \neq \hat{q}\}$ is a union of disjoint intervals $[i, j)$ with $j = \hat{i}$, or (i, j) with $j \neq \hat{i}$ such that any point $q \in (i, j)$ has $\hat{q} = j$. It should be noted that i is not contained in the latter interval for two cases. One case is that $w_q + d_{rq} + Dd_{iq}$ is minimized at both $q = i$ and $q = j$. In this case, $i = \hat{i}$ and hence $i \notin I$. The other case is that $w_q + d_{rq} + Dd_{iq}$ is minimized at $q = j$ and $q = \hat{i} \notin [i, j]$ with $d_{i\hat{i}} \leq d_{ij}$. In this case, $[i, \hat{i})$ is also a subset of I. Conversely, for any interval $[i, \hat{i}) \subseteq I$, there exists an interval $(i, j) \subseteq I$ with $j \neq \hat{i}$ and $d_{i\hat{i}} \leq d_{ij}$. For otherwise, an infinite number of points $i' \notin [i, \hat{i})$ sufficiently close to i has $\hat{i'} = i'$, implying $\hat{i} = i$ by continuity of w'_q.

For any such interval $[i, j)$ or (i, j) of I, and for any point $p \in [i, j]$ and $q \in (i, j)$, it follows that $w'_p = w_j + d_{rj} + Dd_{pj}$ and $w'_q = w_j + d_{rj} + Dd_{qj}$. Therefore, we have

$$\frac{w'_q - w'_p}{d_{pq}} = \frac{D(d_{qj} - d_{pj})}{d_{pq}} = \pm D. \tag{7}$$

The set $R \setminus I$ is a union of disjoint intervals $[i, j]$ (with not necessarily distinct end-points i and j) such that any $p \in [i, j]$ has $\hat{p} = p$. Therefore, for $q \neq p$ in (i, j), it follows that

$$\frac{w'_q - w'_p}{d_{pq}} = \frac{(w_q + d_{rq}) - (w_p + d_{rp})}{d_{pq}} = \frac{w_q - w_p}{d_{pq}} + \frac{d_{rq} - d_{rp}}{d_{pq}}. \tag{8}$$

This approaches an integer as $q \to p$ because the first term approaches an integer by Claim 2 of induction hypothesis, and because the second term approaches ± 1. The absolute value of (8) is at most D by Lemma 1. Because $\hat{q} \in V$ for any $q \in R$ by Claim 1, I consists of finite disjoint intervals. Therefore, $R \setminus I$ also consists of finite disjoint intervals. If p is an end-point of an interval of I or of $R \setminus I$, and if q not in the interval is sufficiently close to p, then q resides in an interval adjacent to the interval. Thus, we have Claim 2 for σr by (7) and (8).

We finally prove Claim 3 for σr. Let $p \in R \setminus V$. If $m'_{p-0} \leq m_{p-0}$ and $m'_{p+0} \leq m_{p+0}$, then the claim holds by induction hypothesis. Otherwise, assume without loss of generality that $m'_{p-0} > m_{p-0}$. There are two such cases from the proof of Claim 2.

One case is that m'_{p-0} becomes D. I.e., for some interval $[i,j)$ or (i,j) in I with $i < j$ such that any q in the interval has $\hat{q} = j$, p is contained in (i,j) and $(w'_q - w'_p)/d_{pq} = D(d_{qj} - d_{pj})/d_{pq} = D$ for any q with $i < q < p$. It should be noted that $p \neq j$ because $p \notin V$. Then, for any q with $p < q < j$, it follows that $(w'_q - w'_p)/d_{pq} = D(d_{qj} - d_{pj})/d_{pq} = -D$, and hence $m'_{p+0} = -D$.

The other case is that $m'_{p-0} = m_{p-0} + 1$. I.e., for some interval $[i,j]$ in $R \setminus I$ with $i < j$, p is contained in $(i,j]$ and $(d_{rq_1} - d_{rp})/d_{pq_1} \to 1$ as $q_1 \to p$ with $i < q_1 < p < r < p + L/2$. It should be noted that $p \neq r$ by $p \notin V$. If $p < j$, then we have $(d_{rq_2} - d_{rp})/d_{pq_2} \to -1$ as $q_2 \to p$ with $p < q_2 < \min\{j,r\}$, which means $m'_{p+0} = m_{p+0} - 1$. If $p = j$, then $p = j$ is an end-point of an interval (j,j') in I with $j < j'$ such that any $q \in (j,j')$ has $\hat{q} = j'$. It should be noted that j cannot be $\hat{q}$ for any point $q \neq j$ by $j = p \notin V$. Therefore, for any q with $p < q < j'$, it follows from (7) that $(w'_q - w'_p)/d_{pq} = D(d_{qj'} - d_{pj'})/d_{pq} = -D$, and hence $m'_{p+0} = -D$. Because $m'_{p-0} \leq D$ by Claim 2, we have Claim 3 for σr. □

We define

$$m_{s\to u} := \lim_{\substack{q\to u \\ q\in[s,u)}} \frac{w_q - w_u}{d_{uq}} \text{ for } u \in V \setminus \{s\},$$

and $m_s := \min\{m_{s\to u} \mid u \in V \setminus \{s\}\}$. Now we state our main lemma, which claims (1) together with two other claims.

Lemma 6 *The following claims hold.*

1. *For $\{p,q\} := V \setminus \{s\}$, $w_p \geq D(L - d_{sp}) + M$, or $w_q \geq D(L - d_{sq}) + M$, or $w_p + w_q \geq m_s d_{pq} + DL + 2M$.*
2. *For any $u \in V$, $w_u + w_{\bar{u}} \geq w_s + \frac{DL}{2} + M$.*
3. *For any $u \in V$, $w_u \geq D d_{su} + M$.*

Proof Sketch We describe a proof sketch prior to our formal proof. Through the extension of networks and work functions to continuous ones, we see that Claim 3 is implied by Claim 2. Actually, if Claim 2 holds, then it follows that $w_u \geq w_s - w_{\bar{u}} + \frac{DL}{2} + M \geq -D d_{s\bar{u}} + \frac{DL}{2} + M = -D(\frac{L}{2} - d_{su}) + \frac{DL}{2} + M = D d_{su} + M$. Here, we have used the fact $w_s - w_{\bar{u}} \geq -D d_{s\bar{u}}$ (Lemma 1). We will prove Claim 2 by induction on events of services and migrations of WFA for requests. The inductive proof for a WFA's migration is easy, because a WFA's migration of distance d decreases w_s by Dd, increases M by Dd, and does not change the left hand side of the inequality in Claim 2. As for the proof for a WFA's service, Claim 2 can inductively be proved for most cases using basic properties of w_u (Lemmas 1–5), some of which are properties of w_u's slope defined using one-sided derivatives. However, there is one exception for which Claim 2 cannot be proved inductively. As shown in Fig. 5, for example, if $\hat{u} = u \neq s$ and a decline from $\bar{u}$ to the request node $r \in V \setminus \{s,u\}$ has slope D, then w_u increases by d_{ur}, whereas $w_{\bar{u}}$ does not increase. Therefore, if $\hat{s} = s$ and $d_{sr} > d_{ur}$,

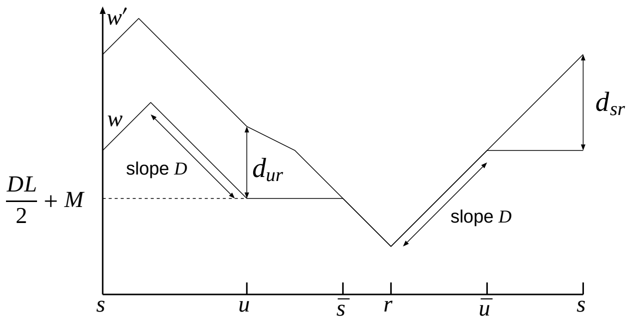

Fig. 5 Situation for which Claim 2 cannot be proved inductively. It follows that $w'_u + w'_{\bar{u}} = w_u + d_{ur} + w_{\bar{u}} < M + \frac{DL}{2} + d_{sr} + w_s = w'_s + \frac{DL}{2} + M$, whereas $w_u + w_{\bar{u}} = w_s + \frac{DL}{2} + M$

then it is the case that the increased amount d_{ur} of $w_u + w_{\bar{u}}$ is less than the increased amount d_{sr} of w_s.

To prove Claim 2 even for such a case, we need Claim 1. The first and second inequalities in Claim 1 imply that w_p or w_q is already large enough, and therefore, the inequality in Claim 2 is satisfied for p or q,[2] respectively. Actually, if the first inequality holds, then it follows that $w_p + w_{\bar{p}} \geq D(L - d_{sp}) + M + w_s - Dd_{s\bar{p}} = w_s + \frac{DL}{2} + M$. Here, we have used the fact $w_{\bar{p}} - w_s \geq -Dd_{s\bar{p}}$ (Lemma 1). The parameter m_s in the third inequality of Claim 1 is the smaller slope at w_p toward s and at w_q toward s. Roughly speaking, m_s is increased by requests from p or q and becomes D in a situation for which Claim 2 cannot be proved inductively. Actually, $m_s = D$ in Fig. 5. However, the third inequality of Claim 1 with $m_s = D$ implies Claim 2 because $w_p + w_q \geq Dd_{pq} + DL + 2M = D(2L - d_{sp} - d_{sq}) + 2M$, implying the first or second inequality of Claim 1. Claim 1 is proved inductively, together with induction hypothesis of Claim 3, and hence that of Claim 2. Thus, Claims 1–3 are proved simultaneously in the formal proof.

Formal Proof Claim 2 implies Claim 3 as described in the proof sketch. We prove Claims 1 and 2 by induction on events of services and migrations of WFA for requests in σ. If $\sigma = \emptyset$, then the claims hold. This is because $w_p + w_q - m_s d_{pq} - 2M = D(d_{sp} + d_{sq}) + Dd_{pq} = DL$, and because $w_u + w_{\bar{u}} - w_s - M = D(d_{su} + d_{s\bar{u}}) = \frac{DL}{2}$ for any $u \in V$. Assume that Claims 1–3 hold for all events in σ. We suppose that w and m are updated to w' and m', respectively, in the service of WFA for a request issued at r after σ, and that M is updated to M' in the subsequent migration of WFA.

We first prove Claim 1 for WFA's service for r. If $w_p \geq D(L - d_{sp}) + M$ or $w_q \geq D(L - d_{sq}) + M$, then the claim holds for the event because $w'_p \geq w_p$ and $w'_q \geq w_q$. Therefore, we assume that $w_p + w_q \geq m_s d_{pq} + DL + 2M$.

Case 1.1: $\hat{p} = s$. Then, $m'_{s \to p} = -D$, and hence $m'_s = -D \leq m_s$. This means that $w'_p + w'_q - m'_s d_{pq} \geq w_p + w_q - m_s d_{pq} \geq DL + 2M$ by induction hypothesis.

[2] To be accurate, we should prove the inequality in Claim 2 for both p and q. Although we do not mention the reason here, we note that one of the first and second inequalities of Claim 1 suffices.

Case 1.2: $\hat{p} = q$. Then, it follows from Claim 3 in induction hypothesis that $w'_p \geq w_q + Dd_{pq} \geq Dd_{sq} + M + Dd_{pq} = D(L - d_{sp}) + M$.

Case 1.3: $\hat{q} \in \{s, p\}$. Similar to the case $\hat{p} \in \{s, q\}$.

Case 1.4: $\hat{p} = p$ and $\hat{q} = q$. If $m'_s \leq m_s + 1$, then $w'_p + w'_q - m'_s d_{pq} \geq w_p + d_{rp} + w_q + d_{rq} - (m_s + 1)d_{pq} \geq w_p + w_q - m_s d_{pq} \geq DL + 2M$ by induction hypothesis. If $m'_s > m_s + 1$, then $m_{s\to p}$ or $m_{s\to q}$, say, $m_{s\to p}$ increases by more than 1. By (the proof of) Lemma 5, this means that $m_{s\to p} < D - 1$, $m'_{s\to p} = D$, and that there exists $i \in (s, p)$ with $p \in (i, \hat{i}]$. It follows from Lemma 2 that $p = \hat{p} = \hat{i}$. Therefore, it follows from Lemma 3 that $w_j - w_p > (D - 1)d_{pj}$ for any $j \in (i, p)$, which contradicts $m_{s\to p} < D - 1$.

Second, we prove Claim 2 for WFA's service for r. Because $w_{\bar{s}} = w_s + w_{\bar{s}} - w_s \geq \frac{DL}{2} + M$ by induction hypothesis, it follows that $w'_s + w'_{\bar{s}} - w'_s \geq w_{\bar{s}} \geq \frac{DL}{2} + M$. Therefore, without loss of generality, it suffices to prove that $w'_p + w'_{\bar{p}} \geq w'_s + \frac{DL}{2} + M$.

Case 2.1: $\hat{p} = s$. Then, $\hat{s} = \hat{p} = s$ by Lemma 2. Therefore, it follows that $w'_s = w_s + d_{rs}$. Moreover, $w'_p = w_s + d_{rs} + Dd_{sp} \geq w_p + d_{rs}$ by Lemma 1. Thus, we have $w'_p + w'_{\bar{p}} - w'_s \geq w_p + d_{rs} + w_{\bar{p}} - (w_s + d_{rs}) \geq \frac{DL}{2} + M$ by induction hypothesis.

Case 2.2: $\hat{p} = q$. Then, $w'_p \geq D(L - d_{sp}) + M$ as shown in Case 1.2. Moreover, $w'_{\bar{p}} \geq w'_s - Dd_{s\bar{p}} = w'_s - D(\frac{L}{2} - d_{sp})$ by Lemma 1. Thus, we have $w'_p + w'_{\bar{p}} \geq D(L - d_{sp}) + M + w'_s - D(\frac{L}{2} - d_{sp}) = w'_s + \frac{DL}{2} + M$.

Case 2.3: $\hat{p} = p$. The proof for the case $\hat{\bar{p}} = s$ is similar to that for the case $\hat{p} = s$. If $\hat{\bar{p}} = p$, then it follows from Claim 3 in induction hypothesis that $w'_{\bar{p}} = w_p + d_{rp} + Dd_{p\bar{p}} \geq Dd_{sp} + M + \frac{DL}{2}$. Moreover, $w'_p \geq w'_s - Dd_{sp}$ by Lemma 1. Thus, we have $w'_p + w'_{\bar{p}} \geq w'_s + M + \frac{DL}{2}$. If $\hat{\bar{p}} = \bar{p}$, then it follows from the minimality of w'_s that $w'_s = w_{\hat{s}} + d_{r\hat{s}} + Dd_{s\hat{s}} \leq w_s + d_{rs}$. Thus, by induction hypothesis, we have $w'_p + w'_{\bar{p}} - w'_s \geq w_p + d_{rp} + w_{\bar{p}} + d_{r\bar{p}} - (w_s + d_{rs}) \geq M + \frac{DL}{2}$. Assume the remaining case $\hat{\bar{p}} = q$. Then, $w_{\bar{p}} - w_q > (D-1)d_{\bar{p}q}$ by Lemma 3. This means $m_{s\to q} = D$ because $m_{s\to q}$ is an integer at most D by Lemma 5, and because there is no node of V between $\bar{p}$ and q, and therefore, no convex point in $(\bar{p}, q)$ by Lemma 5.

Case 2.3.1: $m_{s\to p} = D$. Then, it follows from Claim 1 in induction hypothesis that $w_p \geq D(L - d_{sp}) + M$, or $w_q \geq D(L - d_{sq}) + M$, or $w_p + w_q \geq Dd_{pq} + DL + 2M$. The third inequality implies the first or second inequality. Therefore, it follows that $w'_p \geq w_p \geq D(L - d_{sp}) + M$, or that $w'_{\bar{p}} = w_q + d_{rq} + Dd_{q\bar{p}} \geq D(L - d_{sq}) + M + d_{rq} + Dd_{q\bar{p}} \geq M + D(L - d_{s\bar{p}})$. Both cases can be proved using similar arguments for Case 2.2.

Case 2.3.2: $m_{s\to p} \le D-1$. This means $w_{\bar{q}} - w_p \le (D-1)d_{p\bar{q}}$ because there is no node of V between $\bar{q}$ and p, and therefore, no convex point in $(\bar{q}, p)$ by Lemma 5. Therefore, it follows that $w'_p + w'_{\bar{p}} = w_p + d_{rp} + w_q + d_{rq} + Dd_{q\bar{p}} \ge w_{\bar{q}} - (D-1)d_{p\bar{q}} + d_{rp} + w_q + d_{rq} + Dd_{q\bar{p}} = w_q + w_{\bar{q}} + d_{p\bar{q}} + d_{rp} + d_{rq} \ge w_s + \frac{DL}{2} + M + \frac{L}{2}$ by induction hypothesis. Because $w'_s \le w_s + d_{rs} \le w_s + \frac{L}{2}$ by the minimality of w'_s, we have $w'_p + w'_{\bar{p}} \ge w'_s + \frac{DL}{2} + M$.

Finally, we prove Claims 1 and 2 for WFA's migration from s to another node, say, p after the service for r. It follows that

$$w'_s - w'_p = Dd_{sp}. \tag{9}$$

Therefore, it follows that $m'_p = -D$. Moreover, it follows from Claims 2 and 3 (for the event of WFA's service) that

$$w'_u + w'_{\bar{u}} \ge w'_s + \frac{DL}{2} + M \text{ for any } u \in V, \text{ and} \tag{10}$$

$$w'_p \ge Dd_{sp} + M. \tag{11}$$

Furthermore, because $\bar{q} \in (s, p)$, it follows that

$$w'_s - w'_{\bar{q}} = Dd_{s\bar{q}} = D\left(\frac{L}{2} - d_{sq}\right). \tag{12}$$

We obtain $w'_s \ge 2Dd_{sp} + M$ from (9) and (11), and $w'_q \ge D(L - d_{sq}) + M$ from (10) with $u = q$ and (12). Thus, we have $w'_s + w'_q - m'_p d_{sq} \ge 2Dd_{sp} + M + D(L - d_{sq}) + M + Dd_{sq} = DL + 2(Dd_{sp} + M) = DL + 2M'$. Moreover, it follows from (9) and (10) that $w'_u + w'_{\bar{u}} - w'_p \ge \frac{DL}{2} + M + Dd_{sp} = \frac{DL}{2} + M'$ for any $u \in V$. □

By Lemma 6, we have (1), and hence Theorem 1.

4 Counter-Based Algorithm

In this section we design a counter-based algorithm called CBA and prove the following theorems:

Theorem 2 CBA *is* 3*-competitive on three nodes if* $D \le 2$.

We define and analyze CBA in three stages. In Sect. 4.1, we review a 3-competitive algorithm, called COUNT, for uniform networks presented in [8][3] and prove that COUNT in fact has generalized competitiveness as follows:

Theorem 3 COUNT *is* ρ*-competitive on complete graphs with edge-weights between* 1 *and* $2 - 2/\rho$ *for any* $\rho \ge 3$.

We define CBA for three nodes by extending COUNT in Sect. 4.2, and analyze CBA in Sect. 4.3.

[3] Although the algorithm described here is slightly modified, it is essentially same as the original version.

4.1 Algorithm for Restricted Edge-Weights

In this subsection we consider graphs of arbitrary size. COUNT maintains a counter $C_v \geq 0$ for each node v so that $\sum_{v\in V} C_v = 2D$, and that the server of COUNT always has a positive counter. Initially, the server has a counter of $2D$, and the other nodes have counters of 0. If a request is issued on a node other than the server, then COUNT decrements a positive counter of a node by 1 and increments the counter of the request node by 1. If a counter becomes $2D$, then COUNT moves the server to the node with this counter. The 3-competitiveness of COUNT is proved by verifying that for each event of COUNT's migration, OPT's migration, and services of COUNT and OPT for a request,

$$f := \Delta\text{COUNT} + \Delta\Phi - \rho\Delta\text{OPT} \leq 0 \tag{13}$$

is satisfied for $\rho = 3$. Here, Φ is a *potential function* of counters and the servers s and t of CBA and OPT, respectively, and defined as follows:

$$\Phi := \frac{\rho}{2}\sum_{v\in V} C_v d_{tv} + \left(\frac{\rho}{2} - 1\right)\sum_{v\in V} C_v d_{sv}.$$

ΔCOUNT, ΔOPT, $\Delta\Phi$ are the amounts of change of COUNT's cost, OPT's cost, and Φ in the event, respectively. Since $\Phi \geq 0$, by summing (13) overall events, we can prove that COUNT is ρ-competitive.

Theorem 3 will be proved by verifying that for the service event of COUNT and OPT for a request on r, if COUNT decrements the counter of a node $u \neq s$ with $d_{sr} \leq (1-2/\rho)d_{su} + d_{ur}$, then (13) is satisfied. If $u = s$, then (13) is satisfied from the original proof. As for the migration event of COUNT or OPT, (13) is satisfied regardless of the structure of the network because COUNT always moves the server from a node of counter 0 to a node with counter $2D$. Therefore, if the server is located at a node s satisfying

$$d_{sv} \leq (1 - \tfrac{2}{\rho})d_{su} + d_{uv} \text{ for any distinct } u, v \in V \setminus \{s\}, \tag{14}$$

then (13) is satisfied for any event considered here. We formally prove this in Lemmas 7–9 below.

Lemma 7 *Suppose that* COUNT *and* OPT *serve a request issued at $r \in V$ with the servers on s and t, respectively. If (14) is satisfied, then $f \leq 0$.*

Proof Obviously, ΔCOUNT $= d_{rs}$ and ΔOPT $= d_{rt}$ for the services of COUNT and OPT, respectively. If $r = s$, then no counters are changed. Therefore, $\Delta\Phi = 0$, and hence, $f = 0 + 0 - \rho d_{rt} \leq 0$. Otherwise, the amount of 1 is moved from the counter of a node u to the counter of r. If $u \neq s$, then it follows that $\Delta\Phi = \frac{\rho}{2}(d_{tr} - d_{tu}) + (\frac{\rho}{2} - 1)(d_{sr} - d_{su})$. Therefore, we have

$$\begin{aligned} f &= d_{rs} + \tfrac{\rho}{2}(d_{tr} - d_{tu}) + (\tfrac{\rho}{2} - 1)(d_{sr} - d_{su}) - \rho d_{rt} \\ &= \tfrac{\rho}{2}(d_{rs} - d_{tr} - d_{tu}) - (\tfrac{\rho}{2} - 1)d_{su} \leq \tfrac{\rho}{2}\left(d_{rs} - d_{ru} - (1 - \tfrac{2}{\rho})d_{su}\right) \leq 0. \end{aligned}$$

If $u = s$, then it follows that $\Delta\Phi = \frac{\rho}{2}(d_{tr} - d_{ts}) + (\frac{\rho}{2} - 1)d_{sr}$. Therefore, we have

$$f = d_{rs} + \tfrac{\rho}{2}(d_{tr} - d_{ts}) + (\tfrac{\rho}{2} - 1)d_{sr} - \rho d_{rt} = \tfrac{\rho}{2}(d_{rs} - d_{rt} - d_{st}) \leq 0.$$

□

Lemma 8 *If* OPT *moves the server from t to q, then $f \le 0$.*

Proof Obviously, ΔCOUNT $= 0$ and ΔOPT $= Dd_{tq}$ for OPT's migration. Moreover, $\Delta\Phi = \frac{\rho}{2}\sum_{v\in V} C_v(d_{qv} - d_{tv})$. Therefore, we have

$$\begin{aligned} f &= 0 + \frac{\rho}{2}\sum_{v\in V} C_v(d_{qv} - d_{tv}) - \rho D d_{tq} = \frac{\rho}{2}\sum_{v\in V} C_v(d_{qv} - d_{tv}) - \frac{\rho}{2}\sum_{v\in V} C_v d_{tq} \\ &= \frac{\rho}{2}\sum_{v\in V} C_v(d_{qv} - d_{tv} - d_{tq}) \le 0. \end{aligned}$$

□

Lemma 9 *Suppose that* COUNT *moves the server from s to p. If $\rho \ge 3$, then $f \le 0$.*

Proof Obviously, ΔCOUNT $= Dd_{sp}$ and ΔOPT $= 0$ for COUNT's migration. Because p has the counter of $2D$ and all the other nodes have counters of 0, it follows that $\Delta\Phi = (\frac{\rho}{2} - 1)\sum_{v\in V} C_v(d_{pv} - d_{sv}) = (\frac{\rho}{2} - 1)C_p(-d_{sp}) = -D(\rho - 2)d_{sp}$. Therefore, we have

$$f = Dd_{sp} - D(\rho - 2)d_{sp} - 0 = -D(\rho - 3)d_{sp} \le 0.$$

□

If a complete graph has edges of weights between 1 and $2 - 2/\rho$, then (14) is satisfied for every node s. Therefore, we have Theorem 3.

4.2 Algorithm for Three Nodes

If the server is located at a node s not satisfying (14), then it may be the case that $f > 0$. We shall amortize the excessive debt. Let A be the set of nodes satisfying (14) and B be the set of nodes not contained in A. In the rest of this section, we consider graphs with three nodes and labels as shown in Fig. 2. Moreover, we assume $\rho = 3$ for simplicity, and $y \ge \max\{x, z\}$ without loss of generality. Then, it follows that $b \in A$, and hence, $B \subseteq \{a, c\}$. This is because $x \le y \le (1 - \frac{2}{\rho})z + y$ and $z \le y \le (1 - \frac{2}{\rho})x + y$.

We design our algorithm CBA by introducing the following policy to COUNT. If the server, say a, is in B, then CBA always decrements a's counter for a request on b or c and increments the counter of the request node. With this policy, (13) is satisfied for any service event. However, this policy may cause a situation that the counters of both b and c are less than $2D$ when a's counter becomes 0. This situation forces CBA to move the server to b or c, because a has no counter to be decremented for further requests on b or c. This migration may cause $f > 0$. Precisely, f depends on the position of the server t of OPT and distribution of values of the counters. If the counter of c is sufficiently large, then the excessive debt for the migration from a to c can entirely be amortized by the sum of f associated with service events between the previous and current migrations. Otherwise, although the excessive debt for the migration from a to b may still remain unpaid through the previous service events, it can be amortized by the sum of f associated with service events and a possible OPT's migration between the current and next migrations of CBA. CBA determines the destination of the migration by estimating the excessive debt for the migration and the amount that can amortize the debt.

Now we formally define CBA. We divide the input sequence of clients into phases so that a migration of CBA ends the current phase. When a new phase begins, CBA sets the counter of the previous server to 0. We define a function $\Psi_{st} \le 0$ of counters of the servers s and t of CBA and OPT, respectively, at the end of a phase, i.e., just after the migration of CBA to s. If $B = \emptyset$, then $\Psi_{st} := 0$ for any s and t. Otherwise,

$$\Psi_{st} := 0 \text{ if } s \in \{a,c\}, \text{ or } s = b \text{ and } t \neq v,$$
$$\Psi_{bv} := \max\left\{C_{\bar{v}}(-\tfrac{1}{2}d_{b\bar{v}} - \tfrac{3}{2}(d_{v\bar{v}} - d_{bv})), \tfrac{3}{2}C_b(d_{b\bar{v}} - d_{vb} - d_{v\bar{v}})\right\},$$

where $\{v,\bar{v}\} = \{a,c\}$ with $C_v = 0$.

If a request is issued at a node r, then CBA performs the following procedure unless $r = s$.

1. If $s \in A$ and there exists unique $\bar{r} \in V \setminus \{s,r\}$ with $C_{\bar{r}} \ge 1$, then $C_{\bar{r}}{-}{-}$ and $C_r{+}{+}$. Otherwise, $C_s{-}{-}$ and $C_r{+}{+}$.
2. If $C_s = 0$, then move the server as follows:
   (a) If $s \in A$, then move the server to r. Step 1 implies $C_r = 2D$ in this case.
   (b) If $s \in B$ and $F_b \le F_{\bar{s}}$ (F is defined later), then move the server to b, where $\{\bar{s}\} = V \setminus \{s,b\}$. It should be noted that $\{s,\bar{s}\} = \{a,c\}$.
   (c) If $s \in B$ and $F_b > F_{\bar{s}}$, then move the server to $\bar{s}$, and set $C_b := 0$ and $C_{\bar{s}} := 2D$.

Here, for $p \in \{b,\bar{s}\}$,

$$F_p := \max_{t,q \in V}\{M_{pq} + S_q + \Psi_{pq} - \Psi'_{st}\},$$
$$M_{bq} := C_{\bar{s}}(\tfrac{L}{2} - d_{s\bar{s}}) \text{ for } q \in V,$$
$$M_{\bar{s}q} := C_b\left(\tfrac{1}{2}(d_{s\bar{s}} - d_{sb}) + \tfrac{3}{2}(d_{\bar{s}q} - d_{bq})\right) \text{ for } q \in V,$$
$$S_s := 0, \text{ and}$$
$$S_q := \max\left\{-3C_{\bar{s}}(\tfrac{L}{2} - d_{s\bar{s}}), -3C_b(\tfrac{L}{2} - d_{sb}), -3C_b(\tfrac{L}{2} - d_{b\bar{s}})\right\} \text{ for } q \in \{b,\bar{s}\}.$$

We have used Ψ' to denote Ψ associated with the previous phase and migration. If the current phase is the first phase, then Ψ' is defined using the initial server and counters. Moreover, Ψ_{pq} is associated with the current phase and migration. It should be noted that Ψ_{pq} can be computed just before the migration of CBA to p using counters at this point. This is because CBA changes no counters if $p = b$, and because $\Psi_{aq} = \Psi_{cq} = 0$.

The intuitions of Ψ, F, M, and S are as follows: S and M are corrections of Φ in the current phase, i.e., upper bounds of increase of (CBA's cost) $+ \Phi - \rho$(OPT's cost) for services and migration of CBA, respectively. Since M may be positive and $S \le 0$, M may yield the excessive debt of the current phase and be amortized by S. The debt actually remains unpaid if $p = b$, whereas S is enough if $p \neq b$. In the next phase after CBA moves the server to b, in particular, we can save sufficient deposit to amortize the remaining debt of the current phase, as well as the debt of the next phase. Ψ is introduced to transfer such deposit from the next phase to the current phase. F is the total debt of a phase taking into account Ψ. Our goal is to prove that F_b or $F_{\bar{s}}$ is at most 0.

4.3 Analysis of CBA

For any event e, let $\Delta\mathrm{CBA}(e)$ and $\Delta\mathrm{OPT}(e)$ be the costs of CBA and OPT for e, respectively. Moreover, let $\Delta\Phi(e)$ be the amount of change of Φ for e. Furthermore, let $f(e) := \Delta\mathrm{CBA}(e) + \Delta\Phi(e) - \rho\Delta\mathrm{OPT}(e)$. We will omit e in the notations if e is clear from the context.

Lemmas 10–12 below are detailed statements of Lemmas 7–9, respectively, except that CBA's migration in Step 2b or 2c is included in Lemma 12. These lemmas imply that we can save some deposit (as Ψ and S), and will be used to prove that the deposit can entirely amortize the excessive debt (M) for the migration in Step 2b or 2c.

Lemma 10 *Suppose that* CBA *and* OPT *serve a request issued at $r \in V$ with the servers on s and t, respectively. If $r = s$, then $f = -3d_{rt} \le 0$. If $r \ne s$, $s \in A$, and $C_{\bar{r}} \ge 1$, then $f \le \frac{3}{2}(d_{rs} - d_{r\bar{r}}) - \frac{1}{2}d_{s\bar{r}} \le 0$, where $\{\bar{r}\} = V \setminus \{s,r\}$. Otherwise, $f = \frac{3}{2}(d_{rs} - d_{rt} - d_{st}) \le 0$.*

Proof By the definition of CBA, if $r \ne s$, $s \in A$, and $C_{\bar{r}} \ge 1$, then the amount of 1 is moved from $C_{\bar{r}}$ to C_r. Otherwise, the amount of 1 is moved from C_s to C_r. Therefore, we have the lemma by the proof of Lemma 7. □

Lemma 11 *If* OPT *moves the server from t to q, then $f = \frac{3}{2}\sum_{v\in V} C_v(d_{qv} - d_{tv} - d_{tq}) \le 0$.*

Proof The lemma is directly obtained from the proof of Lemma 8. □

Lemma 12 *Suppose that* CBA *moves the server from s to p. If the server is moved in Step 2a, then $f = 0$. If the server is moved in Step 2b or 2c, then $f = M_{pq}$, where q is the server of* OPT *at the migration of* CBA*. In particular, if $C_p = 2D$, then $f = 0$ for any case.*

Proof Obviously, $\Delta\mathrm{CBA} = Dd_{sp}$ and $\Delta\mathrm{OPT} = 0$ for CBA's migration. If CBA moves the server in Step 2a or 2b, then no counters are changed in the steps and $C_s = 0$. Therefore, $\Delta\Phi = \frac{1}{2}\sum_{v\in V} C_v(d_{pv} - d_{sv}) = \frac{1}{2}(-C_p d_{sp} + C_{\bar{p}}(d_{p\bar{p}} - d_{s\bar{p}}))$, where $\{\bar{p}\} = V \setminus \{s,p\}$. Thus, we have

$$
\begin{aligned}
f &= Dd_{sp} + \tfrac{1}{2}(-C_p d_{sp} + C_{\bar{p}}(d_{p\bar{p}} - d_{s\bar{p}})) - 0\\
&= Dd_{sp} + \tfrac{1}{2}(-(2D - C_{\bar{p}})d_{sp} + C_{\bar{p}}(d_{p\bar{p}} - d_{s\bar{p}}))\\
&= \tfrac{1}{2}C_{\bar{p}}(d_{sp} + d_{p\bar{p}} - d_{s\bar{p}}),
\end{aligned}
$$

which equals 0 if $C_p = 2D$, implied by Step 2a. This is because $C_p = 2D$ implies $C_{\bar{p}} = 0$. For Step 2b, $f = M_{pq}$ because $s \in \{a,c\}$, $p = b$, and $\bar{p} = \bar{s}$.

If CBA moves the server in Step 2c, then C_p and $C_{\bar{p}}$ are set to $2D$ and 0, respectively, after the migration. Moreover, $C_s = 0$ during the migration. Therefore, $\Delta\Phi = \frac{3}{2}((2D - C_p)d_{qp} + (0 - C_{\bar{p}})d_{q\bar{p}}) + \frac{1}{2}(-C_p d_{sp} - C_{\bar{p}} d_{s\bar{p}})$. Thus, we have

$$
\begin{aligned}
f &= Dd_{sp} + \tfrac{3}{2}\left((2D - C_p)d_{qp} + (0 - C_{\bar{p}})d_{q\bar{p}}\right) + \tfrac{1}{2}(-C_p d_{sp} - C_{\bar{p}} d_{s\bar{p}}) - 0\\
&= Dd_{sp} + \tfrac{3}{2}C_{\bar{p}}(d_{qp} - d_{q\bar{p}}) + \tfrac{1}{2}(-(2D - C_{\bar{p}})d_{sp} - C_{\bar{p}} d_{s\bar{p}})\\
&= C_{\bar{p}}\left(\tfrac{1}{2}(d_{sp} - d_{s\bar{p}}) + \tfrac{3}{2}(d_{pq} - d_{\bar{p}q})\right),
\end{aligned}
$$

which equals M_{pq} because $s \in \{a,c\}$, $p = \bar{s}$, and $\bar{p} = b$ for Step 2c. Obviously $f = 0$ if $C_p = 2D$, implying $C_{\bar{p}} = 0$. □

Fix a phase, and let ϕ be the sequence of events in the phase consisting of services of CBA and OPT for a request, migrations of OPT, and a migration of CBA. Suppose that CBA and OPT locate the servers at s and t, respectively, at the beginning of the phase, and at p and q, respectively, at the end of the phase. We will prove $g := \sum_{e\in\phi} f(e) + \Psi_{pq} - \Psi'_{st} \le 0$. If this holds, then because both Φ and Ψ can be bounded from below independently of the number of requests, we can prove that CBA is 3-competitive by summing up the inequalities overall phases. In what follows, C_v denotes the counter of $v \in V$ just before CBA moves the server to p. This means that $C_s = 0$.

If $B = \emptyset$ or $s \in \{a,c\} \cap A$, then $C_p = 2D$ as mentioned in Step 2a of the definition of CBA, and $\Psi'_{st} = 0$. Therefore, $g \le 0$ by $\Psi_{pq} \le 0$ and Lemmas 10–12. To prove Theorem 2, it remains to prove that $g \le 0$ for the case $B \ne \emptyset$ and $s \in \{b\} \cup B$.

Lemma 13 *If $s = b$, then $g \le 0$.*

Proof Let C'_v be the value of counter of $v \in V$ at the beginning of the phase, i.e., just after the previous migration of CBA to $s = b$. Because CBA moved the server from $u \in \{a,c\}$ to b in the previous migration, $C'_u = 0$ by the definition of CBA. We prove the lemma for the case $u = a$ and omit a proof for the case $u = c$, which can be obtained with a similar argument. Because $b \in A$, $C_p = 2D$ by the definition of CBA. If $p = c$, then by Lemma 12, $f = 0$ for the event of the migration of CBA to c. Therefore, $\sum_{e\in\phi} f(e) \le 0$ by Lemmas 10 and 11. If $p = a$, then an amount at least C'_c must be moved from c's counter to a's counter in the phase. This means that at least C'_c requests on a move the amount of C'_c from c's counter to a's counter. It should be noted that CBA never increases the server's counter. Therefore, it follows from Lemma 10 that $\sum_{e\in\phi} f(e) \le C'_c(\frac{3}{2}(x-y) - \frac{1}{2}z)$. Thus, we can obtain $g \le \sum_{e\in\phi} f(e) - \Psi'_{bt} \le 0$ if $t \in \{b,c\}$ or $p = a$.

We assume that $t = a$ and $p = c$. An amount at least C'_b must be moved from b's counter to c's counter in the phase. If a situation that c's counter becomes 0 occurs in the phase, then the amount at least C'_c must be moved from c's counter to a's counter, and hence, we can prove $g \le 0$ as in the case $p = a$. We assume that no such situation occurs. Then, C'_b requests on c moves the amount of C'_b from b's counter to c's counter when a's counter is 0. It should be noted that CBA never decreases the counter of a server in A unless one of the other nodes has the counter of 0. Therefore, if OPT does not move the server throughout the phase, then $\sum_{e\in\phi} f(e) \le \frac{3}{2}C'_b(z-x-y)$ by Lemma 10 and the above analysis that $f = 0$ for the migration of CBA to c. Thus, we can obtain $g \le \sum_{e\in\phi} f(e) - \Psi'_{ba} \le 0$.

It remains to prove the lemma for the case that $t = a$, $p = c$, and that OPT moves the server in the phase. Because we have assumed that c has a positive counter throughout the phase, no amount moves from b's counter to a's counter directly. Therefore, if $\lambda \le C'_b$ is the amount moving from b's counter to c's counter before the first migration of OPT, and if δ is the smaller value of $C'_c + \lambda$ and the number of requests issued at a before the OPT's migration, then b and c have the counters $C'_b - \lambda$ and at least $C'_c + \lambda - \delta$, respectively, at the point of the migration of OPT. For the events of services of CBA and OPT for the δ requests on a and the λ requests on c,

$f \le \delta(\frac{3}{2}(x-y)-\frac{1}{2}z)+\frac{3}{2}\lambda(z-y-x)$ by Lemma 10. If OPT moves the server from a to c, then by Lemma 11, $f \le \frac{3}{2}(C'_c+\lambda-\delta)(d_{cc}-d_{ac}-d_{ac}) = -3(C'_c+\lambda-\delta)y$ for the event. Moreover, $f=0$ for CBA's migration from b to c. Therefore, it follows that

$$\begin{aligned}\sum_{e\in\phi} f(e) &\le \delta\left(\tfrac{3}{2}(x-y)-\tfrac{1}{2}z\right)+\tfrac{3}{2}\lambda(z-y-x)-3(C'_c+\lambda-\delta)y\\ &= \delta\left(\tfrac{3}{2}(x+y)-\tfrac{1}{2}z\right)+\tfrac{3}{2}\lambda(z-3y-x)-3C'_cy\\ &\le (C'_c+\lambda)\left(\tfrac{3}{2}(x+y)-\tfrac{1}{2}z\right)+\tfrac{3}{2}\lambda(z-3y-x)-3C'_cy\\ &= C'_c\left(\tfrac{3}{2}(x-y)-\tfrac{1}{2}z\right)+\lambda(z-3y) \le C'_c\left(\tfrac{3}{2}(x-y)-\tfrac{1}{2}z\right)\end{aligned}$$

Thus, we can obtain $g \le \sum_{e\in\phi} f(e) - \Psi'_{ba} \le 0$. If OPT moves the server to b, then by Lemma 11, $f \le \frac{3}{2}((C'_b-\lambda)(d_{bb}-d_{ab}-d_{ab})+(C'_c+\lambda-\delta)(d_{bc}-d_{ac}-d_{ab})) = \frac{3}{2}(-2C'_bx+\lambda(z-y+x)+(C'_c-\delta)(z-y-x))$ for the event. Therefore, it follows that

$$\begin{aligned}\sum_{e\in\phi} f(e) &\le \delta\left(\tfrac{3}{2}(x-y)-\tfrac{1}{2}z\right)+\tfrac{3}{2}\lambda(z-y-x)\\ &\quad+\tfrac{3}{2}\left(-2C'_bx+\lambda(z-y+x)+(C'_c-\delta)(z-y-x)\right)\\ &= \delta(3x-2z)+3\lambda(z-y)+\tfrac{3}{2}\left(-2C'_bx+C'_c(z-y-x)\right)\end{aligned} \tag{15}$$

If $3x \ge 2z$, then the last expression of (15) is at most

$$\begin{aligned}&(C'_c+\lambda)(3x-2z)+3\lambda(z-y)+\tfrac{3}{2}\left(-2C'_bx+C'_c(z-y-x)\right)\\ &= C'_c(3x-2z)+\tfrac{3}{2}C'_c(z-y-x)+3(\lambda-C'_b)x+\lambda(z-3y)\\ &\le C'_c\left(\tfrac{3}{2}(x-y)-\tfrac{1}{2}z\right).\end{aligned}$$

If $3x < 2z$, then the last expression of (15) is at most

$$\tfrac{3}{2}(-2C'_bx) \le \tfrac{3}{2}\left(-C'_bx-C'_b(y-z)\right) = \tfrac{3}{2}C'_b(z-x-y).$$

Thus, we can obtain $g \le \sum_{e\in\phi} f(e) - \Psi'_{ba} \le 0$. □

We prove $g \le 0$ for the remaining case $s \in \{a,c\} \cap B$ in Lemmas 14 and 15 below.

Lemma 14 *If $s \in B$, then $\sum_{e\in\phi} f(e) \le M_{pq}+S_q$.*

Proof We prove the lemma for the case $s=a$ and omit a proof for the case $s=c$, which can be obtained with a similar argument. For the event of CBA's migration to p, $f=M_{pq}$ by Lemma 12. Moreover, $\sum_{e\in\phi'} f(e) \le 0 = S_a$ by Lemmas 10 and 11, where ϕ' is the sequence of events obtained from ϕ by removing the last event of CBA's migration. Therefore, it suffices to prove that $\sum_{e\in\phi'} f(e) \le S_q = \max\{-3C_c(\frac{L}{2}-y), -3C_b(\frac{L}{2}-x), -3C_b(\frac{L}{2}-z)\}$ for $q \in \{b,c\}$.

Let δ_b and δ_c be the numbers of requests issued at b and c in the phase, respectively, before the point that OPT locates the server on q and keeps it until the end of the phase. Then, $\delta_b \le C_b$, $\delta_c \le C_c$, and b and c have the counters of δ_b and δ_c at the point, respectively. This is because CBA sets the server's counter to $2D$ after it moves the server to a node in B, and hence $C'_a = 2D$ and $C'_b = C'_c = 0$, and because CBA decreases only the server's counter when the server is in B. Therefore, $C_b-\delta_b$ and

$C_c - \delta_c$ requests are issued on b and c after that point, respectively. For the events of the services of CBA and OPT for the $C_{\bar{q}} - \delta_{\bar{q}}$ requests on unique $\bar{q} \in \{b,c\} \setminus \{q\}$, $f \le (C_{\bar{q}} - \delta_{\bar{q}}) \cdot \frac{3}{2}(d_{a\bar{q}} - d_{\bar{q}q} - d_{aq})$ by Lemma 10. If OPT keeps the server on q throughout the phase, i.e., $\delta_b = \delta_c = 0$, then

$$\sum_{e \in \phi'} f(e) \le \tfrac{3}{2} C_{\bar{q}}(d_{a\bar{q}} - d_{\bar{q}q} - d_{aq}) \le \max\left\{3C_b(x - \tfrac{L}{2}), 3C_c(y - \tfrac{L}{2})\right\}.$$

If OPT moves the server from a to q at the point that b and c have the counters of δ_b and δ_c, then $f \le \frac{3}{2}(\delta_q(d_{qq} - d_{aq} - d_{aq})) + \delta_{\bar{q}}(d_{q\bar{q}} - d_{a\bar{q}} - d_{aq})) = \frac{3}{2}(-2\delta_q d_{aq} + \delta_{\bar{q}}(d_{q\bar{q}} - d_{a\bar{q}} - d_{aq}))$ by Lemma 11. Combining this event and the events for $C_b - \delta_b$ and $C_c - \delta_c$ requests on b and c, respectively, we have

$$\begin{aligned}
\sum_{e \in \phi'} f(e) &\le \tfrac{3}{2}\left(-2\delta_q d_{aq} + \delta_{\bar{q}}(d_{q\bar{q}} - d_{a\bar{q}} - d_{aq})\right) + (C_{\bar{q}} - \delta_{\bar{q}}) \cdot \tfrac{3}{2}(d_{a\bar{q}} - d_{\bar{q}q} - d_{aq}) \\
&= \tfrac{3}{2}\left(C_{\bar{q}}(d_{a\bar{q}} - z - d_{aq}) - 2\delta_q d_{aq} - 2\delta_{\bar{q}}(d_{a\bar{q}} - z)\right) \\
&\le \tfrac{3}{2} C_{\bar{q}}(|d_{a\bar{q}} - z| - d_{aq}) \qquad [\text{by } \delta_{\bar{q}} \le C_{\bar{q}}] \\
&\le \max\left\{3C_c(y - \tfrac{L}{2}), 3C_b(x - \tfrac{L}{2}), 3C_b(z - \tfrac{L}{2})\right\}.
\end{aligned}$$

If OPT moves the server from $\bar{q}$ to q at the point that b and c have the counters of δ_b and δ_c, then by analyzing this event with Lemma 11, we have

$$\begin{aligned}
\sum_{e \in \phi'} f(e) &\le \tfrac{3}{2}\left(\delta_q(d_{qq} - d_{\bar{q}q} - d_{\bar{q}q}) + (2D - \delta_q - \delta_{\bar{q}})(d_{qa} - d_{\bar{q}a} - d_{\bar{q}q})\right) \\
&= \tfrac{3}{2}\left((2D - \delta_{\bar{q}})(d_{qa} - d_{\bar{q}a} - d_{\bar{q}q}) - \delta_q(d_{qa} - d_{\bar{q}a} + d_{\bar{q}q})\right) \\
&\le \tfrac{3}{2}(2D - \delta_{\bar{q}})(d_{qa} - d_{\bar{q}a} - d_{\bar{q}q}) \\
&\le \tfrac{3}{2} C_q(d_{qa} - d_{\bar{q}a} - z) \le \max\left\{3C_b(x - \tfrac{L}{2}), 3C_c(y - \tfrac{L}{2})\right\}.
\end{aligned}$$

Here, we have used the fact that $2D - \delta_{\bar{q}} \ge 2D - C_{\bar{q}} = C_q$. □

Lemma 15 *If $D \le 2$ and $s \in \{a,c\} \cap B$, then $F_b \le 0$ or $F_{\bar{s}} \le 0$.*

Proof We prove the lemma for the case $s = a$ and omit a proof for the case $s = c$, which can be obtained with a similar argument.

We first estimate F_b. Because $\Psi'_{at} = 0$, $\Psi_{bb} = \Psi_{bc} = 0$, $S_a = 0$, $S_b = S_c$, and $M_{ba} = M_{bb} = M_{bc}$, we have $F_b = \max_{t,q \in V}\{M_{bq} + S_q + \Psi_{bq} - \Psi'_{at}\} = M_{ba} + \max\{\Psi_{ba}, S_b\}$. By the definitions of M_{ba}, S_q, and Ψ_{ba},

$$\begin{aligned}
M_{ba} &= C_c(\tfrac{L}{2} - y), \\
S_b &= \max\left\{3C_c(y - \tfrac{L}{2}), 3C_b(x - \tfrac{L}{2}), 3C_b(z - \tfrac{L}{2})\right\}, \text{ and} \\
\Psi_{ba} &= \max\left\{C_c\left(\tfrac{3}{2}(x - y) - \tfrac{1}{2}z\right), \tfrac{3}{2}C_b(z - x - y)\right\}.
\end{aligned}$$

If $\Psi_{ba} = C_c(\frac{3}{2}(x - y) - \frac{1}{2}z)$, then $F_b/C_c \le (\frac{L}{2} - y) + \frac{3}{2}(x - y) - \frac{1}{2}z = 2(x - y) \le 0$. Moreover, if $S_b = 3C_c(y - \frac{L}{2})$ and $\Psi_{ba} \le S_b$, then $F_b/C_c \le (\frac{L}{2} - y) + 3(y - \frac{L}{2}) = 2y -$

$L \le 0$. Thus, the lemma holds for these cases. We assume the remaining cases. Then, by $\frac{1}{2}(z-x-y) = z - \frac{L}{2}$, we have

$$\begin{aligned} F_b &\le C_c(\tfrac{L}{2} - y) + \max\left\{3C_b(x - \tfrac{L}{2}), 3C_b(z - \tfrac{L}{2})\right\} \\ &= (2D - C_b)(\tfrac{L}{2} - y) + 3C_b\left(\max\{x,z\} - \tfrac{L}{2}\right). \end{aligned}$$

Therefore, if $C_b \ge D/2$, then $F_b \le 3C_b(\max\{x,z\} - y) \le 0$. If $C_b < D/2 \le 1$, i.e., $C_b = 0$, then M_{cq}, S_q, Ψ_{cq}, and Ψ'_{at} are all equal to 0 for any $t,q \in V$. Thus, we have $F_c = \max_{t,q\in V}\{M_{cq} + S_q + \Psi_{cq} - \Psi'_{at}\} = 0$. □

By Lemmas 13–15, we have $g \le 0$ for every case. Therefore, the proof of Theorem 2 is completed.

5 Lower Bound

In this section we prove the following theorem:

Theorem 4 *If a deterministic page migration algorithm is ρ-competitive on three nodes, then $\rho = 3 + \Omega(1/D)$. In particular, $\rho > 3$ for any $D \ge 3$.*

5.1 Adversary

To prove Theorem 4, we design a 3-node network and an *adversary*, i.e., a strategy to generate an arbitrarily costly sequence σ of clients against any deterministic online page migration algorithm ALG on the network so that $\text{ALG}(\sigma) > \rho \cdot \text{OPT}(\sigma)$ for some $\rho = 3 + \Omega(1/D)$ with $D \ge 3$. By using such a strategy, we obtain a lower bound of ρ, i.e., $\text{ALG}(\sigma) \ge \rho \cdot \text{OPT}(\sigma) + \alpha$ for any α independent of the number of clients because σ can be arbitrarily costly. Broadly, our strategy repeatedly generates a sequence ϕ of clients so that ALG returns the server to the initial position s_0 after processing each ϕ, and that $\text{ALG}(\phi) > (3 + \Omega(1/D))\text{OPT}_{s_0}(\phi)$. The sequence ϕ begins with a sequence τ such that $\text{ALG}(\tau) > (3 + \Omega(1/D))\text{OPT}(\tau)$, or that ALG moves the server too early to achieve a competitive ratio $3 + o(1/D)$. If ALG locates the server at s_0 after processing τ and has $\text{ALG}(\tau) > (3 + \Omega(1/D))\text{OPT}_{s_0}(\tau)$, then τ is actually a desired sequence ϕ. Otherwise, a subsequent sequence τ' enforces enough separation between costs of ALG and OPT if necessary, and leads ALG to return the server to s_0 with preserving part of the separation, so that $\text{ALG}(\tau\tau') > (3 + \Omega(1/D))\text{OPT}_{s_0}(\tau\tau')$.

In this section we assume without loss of generality that $y \ge x \ge z$. We call a sequence χ a *v-forcing sequence*, denoted by χ_v, if ALG leaves the server on a node v after processing χ. The following Lemma 16 is a tool to enforce enough separation between costs of ALG with too early migration and OPT.

Lemma 16 *Let $P \subseteq V$, $Q := V \setminus P$, and let $p \in P$ and $q \in Q$ be joined by an edge with the minimum weight w overall edges joining P and Q. If there exist $\rho > 3$ and a q-forcing sequence χ of clients such that $(\rho - 1)\text{OPT}_p(\chi) + \text{OPT}_q(\chi) - \text{ALG}(\chi) + (\rho - 5)Dw < 0$, then there exists a p-forcing sequence χ' with $\text{ALG}(\chi\chi') > \rho \cdot \text{OPT}_p(\chi\chi')$ or a q-forcing sequence χ'' with $\text{ALG}(\chi\chi'') > \rho \cdot \text{OPT}_q(\chi\chi'')$.*

Proof We prove that $\chi' := p^{k_1}q^{\ell_1}\cdots p^{k_{i-1}}q^{\ell_{i-1}}p^{k_i}$ or $\chi'' := p^{k_1}q^{\ell_1}\cdots p^{k_i}q^{\ell_i}$ is a desired sequence for some i. Here, k_j (resp. ℓ_j) $(1 \le j \le i)$ is the minimum positive integer such that ALG moves the server from a node of Q (resp. P) to a node P (resp. Q) after processing $\chi p^{k_1}q^{\ell_1}\cdots p^{k_{j-1}}q^{\ell_{j-1}}p^{k_j}$ (resp. $\chi p^{k_1}q^{\ell_1}\cdots p^{k_j}q^{\ell_j}$).

Assume for contradiction that $\mathrm{ALG}(\chi\chi') \le \rho\cdot\mathrm{OPT}_p(\chi\chi')$ and $\mathrm{ALG}(\chi\chi'') \le \rho\cdot\mathrm{OPT}_q(\chi\chi'')$. Because ALG incurs a cost at least w to serve a request in χ' or χ'' and a cost at least Dw to migrate between P and Q, it follows that

$$\mathrm{ALG}(\chi\chi') \ge \mathrm{ALG}(\chi) + (K_i + Di + L_{i-1} + D(i-1))w, \text{ and}$$
$$\mathrm{ALG}(\chi\chi'') \ge \mathrm{ALG}(\chi) + (K_i + Di + L_i + Di)w,$$

where $K_j := \sum_{h=1}^{j} k_h$ and $L_j := \sum_{h=1}^{j} \ell_h$ for $1 \le j \le i$, and $L_0 := 0$. Moreover, an offline algorithm that locates and keeps the server at p (resp. q) after processing χ can process $\chi\chi'$ (resp. $\chi\chi''$) with a cost of $\mathrm{OPT}_p(\chi) + L_{i-1}w$ (resp. $\mathrm{OPT}_q(\chi) + K_i w$). Therefore, it follows that $\mathrm{OPT}_p(\chi\chi') \le \mathrm{OPT}_p(\chi) + L_{i-1}w$, and $\mathrm{OPT}_q(\chi\chi'') \le \mathrm{OPT}_q(\chi) + K_i w$. By the inequalities observed above, we have

$$\mathrm{ALG}(\chi) + (K_i + Di + L_{i-1} + D(i-1))w \le \rho(\mathrm{OPT}_p(\chi) + L_{i-1}w), \text{ and}$$
$$\mathrm{ALG}(\chi) + (K_i + Di + L_i + Di)w \le \rho(\mathrm{OPT}_q(\chi) + K_i w),$$

which yield the inequalities

$$K_i \le (\rho-1)L_{i-1} - D(2i-1) + A \text{ and } L_i \le (\rho-1)K_i - 2Di + B \text{ for } i \ge 1,$$

where $A := (\rho\cdot\mathrm{OPT}_p(\chi) - \mathrm{ALG}(\chi))/w$ and $B := (\rho\cdot\mathrm{OPT}_q(\chi) - \mathrm{ALG}(\chi))/w$. Thus, we have the recurrence

$$K_i \le (\rho-1)^2K_{i-1} - 2\rho Di + (2\rho-1)D + A + (\rho-1)B \text{ for } i \ge 2,$$

which is equivalent to

$$K_i - \frac{2Di}{\rho-2} - \frac{\frac{\rho D}{\rho-2} - A - (\rho-1)B}{\rho(\rho-2)} \le \left\{K_{i-1} - \frac{2D(i-1)}{\rho-2} - \frac{\frac{\rho D}{\rho-2} - A - (\rho-1)B}{\rho(\rho-2)}\right\}(\rho-1)^2.$$

Therefore, it follows that

$$K_i \le \left\{K_1 - \frac{2D}{\rho-2} - \frac{\frac{\rho D}{\rho-2} - A - (\rho-1)B}{\rho(\rho-2)}\right\}(\rho-1)^{2(i-1)} + \frac{2Di}{\rho-2} + \frac{\frac{\rho D}{\rho-2} - A - (\rho-1)B}{\rho(\rho-2)}$$

by $K_1 \le A - D$

$$\le \left\{-\frac{\rho(\rho-1)D}{\rho-2} + (\rho-1)A + B\right\}\frac{(\rho-1)^{2i-1}}{\rho(\rho-2)} + \frac{2Di}{\rho-2} + \frac{\frac{\rho D}{\rho-2} - A - (\rho-1)B}{\rho(\rho-2)}$$
$$= \left\{-\frac{\rho(\rho-1)D}{\rho-2} + (\rho-1)A + B\right\}\cdot\Theta\left((\rho-1)^{2i}\right) + O(i).$$

The factor of $\Theta((\rho-1)^{2i})$ can be estimated as

$$-\frac{\rho(\rho-1)D}{\rho-2} + (\rho-1)A + B = \frac{\rho}{w}\left\{(\rho-1)\mathrm{OPT}_p(\chi) + \mathrm{OPT}_q(\chi) - \mathrm{ALG}(\chi) - \frac{\rho-1}{\rho-2}Dw\right\},$$

which is negative by $-\frac{\rho-1}{\rho-2} \leq \rho - 5$ for $\rho \geq 3$ and by the assumption of the lemma. Therefore, K_i decreases as i grows sufficiently large, but it is impossible by definition. □

Lemmas 17 and 18 below are tools to generate τ' for ALG with $\mathrm{ALG}(\tau) > \rho \cdot \mathrm{OPT}(\tau)$ and with too early migration, respectively.

Lemma 17 *Let $p := a$ and $q := b$, or $p := b$ and $q := c$. Let $w := d_{pq}$. If there exist $\rho > 3$, $\beta > 0$, and a q-forcing sequence χ of clients such that $\mathrm{ALG}(\chi) > \rho \cdot \mathrm{OPT}_q(\chi)$ and $\mathrm{OPT}_q(\chi) \geq \beta Dw$, then there exists a sequence χ' that is a p-forcing sequence with $\mathrm{ALG}(\chi\chi') > \rho' \cdot \mathrm{OPT}_p(\chi\chi')$ or an arbitrarily costly sequence with $\mathrm{ALG}(\chi\chi') > \rho' \cdot \mathrm{OPT}(\chi\chi')$, where $\rho' := \frac{\beta}{\beta+4}(\rho-3)+3$.*

Proof We define χ' as follows:

1. Let ψ^0 be an empty sequence and $j := 1$.
2. ALG have processed $\chi\psi^0 \cdots \psi^{j-1}$ and locates the server on q. Then, we generate requests at p repeatedly until ALG locates the server on p. Let i be the number of the requests on p.
3. If $i \geq ((\beta+1)\rho' - \beta\rho - 1)D$, then set $\chi' := \psi^0 \cdots \psi^{j-1} p^i$, and quit the procedure.
4. Otherwise, we estimate costs of ALG and OPT for the clients p^i with the server initially at q. Wherever ALG moves the server between q and $u \notin \{p,q\}$ during the requests, ALG incurs a cost at least $(i+D)w$. This is because $w \leq d_{pu}$ by $y \geq x \geq z$. An offline algorithm that keeps the server at q can process p^i with a cost of iw. Moreover, an offline algorithm that moves the server from q to p first and keeps the server at p can process p^i with a cost of Dw. Thus, we have

$$\begin{aligned}
&(\rho'-1)\mathrm{OPT}_q(p^i) + \mathrm{OPT}_p(p^i) - \mathrm{ALG}(p^i) + (\rho'-5)Dw \\
&\quad \leq (\rho'-1)iw + Dw - (i+D)w + (\rho'-5)Dw \\
&\quad < \left\{(\rho'-2)((\beta+1)\rho' - \beta\rho - 1) + \rho' - 5\right\}Dw \\
&\quad = \left\{(\beta+1)\rho'^2 - (\beta\rho + 2(\beta+1))\rho' + 2\beta\rho - 3\right\}Dw \\
&\quad = (\beta+1)\left(\rho' - A(\rho)\right)\left(\rho' - B(\rho)\right)Dw < 0,
\end{aligned} \tag{16}$$

where

$$A(\rho) := 1 + \frac{\beta\rho + \sqrt{(\beta\rho - 2(\beta+1))^2 + 12(\beta+1)}}{2(\beta+1)}, \text{ and}$$

$$B(\rho) := 1 + \frac{\beta\rho - \sqrt{(\beta\rho - 2(\beta+1))^2 + 12(\beta+1)}}{2(\beta+1)}.$$

The last inequality of (16) can be proved by verifying that for $\rho \geq 3$,

$$\begin{aligned}
A(\rho) &> \tfrac{d}{d\rho}A(3) \cdot (\rho - 3) + A(3) \qquad \left[\text{by } \tfrac{d^2}{d\rho^2}A(\rho) > 0\right] \\
&= \rho', \text{ and} \\
B(\rho) &< 1 + \tfrac{\beta\rho - |\beta\rho - 2(\beta+1)|}{2(\beta+1)} \leq 2 < \rho'.
\end{aligned}$$

Therefore, by applying Lemma 16 with $P := \{p\}$ and $Q := \{q,u\}$, we can obtain a sequence ψ^j beginning with p^i that is a p-forcing sequence with $\mathrm{ALG}(\psi^j) > \rho'\mathrm{OPT}_p(\psi^j)$ or a q-forcing sequence with $\mathrm{ALG}(\psi^j) > \rho'\mathrm{OPT}_q(\psi^j)$.

5. If ψ^j is a p-forcing sequence, then set $\chi' := \psi^0 \cdots \psi^j$, and quit the procedure. Otherwise, set $j := j+1$, and repeat the process from Step 2.

By definition, χ' is a p-forcing sequence or arbitrarily costly. If the procedure ends in Step 3, then it follows that

$$\begin{aligned}
\text{ALG}(\chi\chi') - \rho' \cdot \text{OPT}_p(\chi\chi') &\geq \text{ALG}(\chi) + \sum_j \text{ALG}(\psi^j) + \text{ALG}(p^i) \\
&\quad - \rho' \left\{ \text{OPT}_q(\chi) + \sum_j \text{OPT}_q(\psi^j) + \text{OPT}_p(p^i) \right\} \\
&> (\rho - \rho')\text{OPT}_q(\chi) + \big((\beta+1)\rho' - \beta\rho\big) Dw - \rho' Dw \\
&= (\rho - \rho')\left(\text{OPT}_q(\chi) - \beta Dw\right) \geq 0.
\end{aligned}$$

If the procedure ends in Step 5, then it follows that

$$\begin{aligned}
\text{ALG}(\chi\chi') - \rho' \cdot \text{OPT}_p(\chi\chi') &\geq \text{ALG}(\chi) + \sum_{h<j} \text{ALG}(\psi^h) + \text{ALG}(\psi^j) \\
&\quad - \rho' \left\{ \text{OPT}_q(\chi) + \sum_{h<j} \text{OPT}_q(\psi^h) + \text{OPT}_p(\psi^j) \right\} \\
&> (\rho - \rho')\text{OPT}_q(\chi) > 0.
\end{aligned}$$

Otherwise, we can similarly prove $\text{ALG}(\chi\chi') - \rho' \cdot \text{OPT}(\chi\chi') > 0$. □

Lemma 18 *Let $\{p,q\} := \{a,b\}$ and $w := d_{pq}$. If there exist $\rho > 3$, $\beta > 0$, and a q-forcing sequence χ of clients such that $(\rho-1)\text{OPT}_p(\chi) + \text{OPT}_q(\chi) - \text{ALG}(\chi) + (\rho-5)Dw < 0$ and $\text{OPT}_q(\chi) \geq \beta Dw$, then there exists a sequence χ' that is an a-forcing sequence with $\text{ALG}(\chi\chi') > \rho' \cdot \text{OPT}_a(\chi\chi')$ or an arbitrarily costly sequence with $\text{ALG}(\chi\chi') > \rho' \cdot \text{OPT}(\chi\chi')$, where $\rho' := \frac{\beta}{\beta+4}(\rho-3)+3$.*

Proof Let $P := \{a\}$ and $Q := \{b,c\}$ if $p = a$, $P := \{b,c\}$ and $Q := \{a\}$ otherwise. By applying Lemma 16 with such P and Q, we can obtain a sequence ψ that is an a-forcing sequence with $\text{ALG}(\chi\psi) > \rho \cdot \text{OPT}_a(\chi\psi)$ or a b-forcing sequence with $\text{ALG}(\chi\psi) > \rho \cdot \text{OPT}_b(\chi\psi)$. If ψ is an a-forcing sequence, then we have obtained a desired sequence. Otherwise, by Lemma 17, there exists a sequence ψ' that is an a-forcing sequence with $\text{ALG}(\chi\psi\psi') > \rho' \cdot \text{OPT}_a(\chi\psi\psi')$ or an arbitrarily costly sequence with $\text{ALG}(\chi\psi\psi') > \rho' \cdot \text{OPT}(\chi\psi\psi')$. Therefore, $\psi\psi'$ is a desired sequence. □

We set the initial server $s_0 := a$. Our strategy to generate σ is defined using a state machine as shown in Fig. 6. In this state machine, a transition represents a server position selected by ALG, together with optional conditions on the number of requests generated in the source state. The parameter $1 \leq \lambda \leq D/3$ will be defined later. A state with the form of u^k (i.e., b^h, a^j, and c^i) represents a sequence of requests that are issued on u until the server position of ALG and the number k of the issued requests meet those associated with one of the outgoing arcs from the state. For example, we generate requests on b at the state b^h and transit to a^+ if ALG moves the server from a to b or c after at most λ requests, while we transit to c^i if ALG keeps the server at

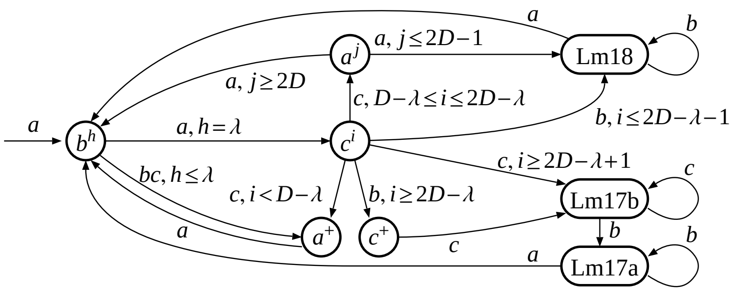

Fig. 6 Strategy to generate σ

a during λ requests on b. At the state a^j, for another example, we generate requests on a until ALG locates the server at a, and transit to Lm18 if the number of generated requests on a is less than $2D$, b^h otherwise. A state with the form of u^+ (i.e., a^+ and c^+) represents a sequence of requests on u until ALG locates the server on u. The states Lm17b and Lm17a represent sequences of requests obtained by applying Lemma 17 with $p := b$ and $q := c$, and with $p := a$ and $q := b$, respectively. The state Lm18 represents a sequence of requests obtained by applying Lemma 18 with $p \in \{a,b\} \setminus \{s\}$ and $q := s$, where $s \in \{a,b\}$ is the server of ALG at the beginning of the state.

5.2 Analysis

Now we prove Theorem 4. Suppose that $y = x + \delta$ and $z = \gamma\delta$ with $\delta > 0$ and $3 \le \gamma \le x/\delta$. We will choose γ and δ later. We divide σ into phases so that entering the state b^h begins a new phase. ALG locates the server on a at the beginning of each phase. Therefore, Theorem 4 is proved if for each a-forcing phase ϕ, $\mathrm{ALG}(\phi) > \rho \cdot \mathrm{OPT}_a(\phi)$ with the server initially at a, and if for an arbitrarily costly phase ϕ, $\mathrm{ALG}(\phi) > \rho \cdot \mathrm{OPT}(\phi)$ with the server initially at a.

Case 1: $\phi = b^h_{bc} a^+$ with $h \le \lambda$. It follows that $\mathrm{ALG}(\phi) > (h+2D)x$ and $\mathrm{OPT}_a(\phi) \le hx$ (cost of keeping the server at a). Thus, we have $\frac{\mathrm{ALG}(\phi)}{\mathrm{OPT}_a(\phi)} > \frac{h+2D}{h} \ge 1 + \frac{2D}{\lambda} \ge 7$.

Case 2: $\phi = \tau\tau'$, where $\tau := b^\lambda_a c^i_b$ with $i \le 2D - \lambda - 1$, and τ' is the sequence of clients generated in the state Lm18. It follows that $\mathrm{ALG}(\tau) = (\lambda + D)x + iy$, $\mathrm{OPT}_a(\tau) = \lambda x + iy$ (cost of keeping the server at a), and $\mathrm{OPT}_b(\tau) \le Dx + iz$ (cost of moving the server to b first and keeping it at b). Thus, we have

$$
\begin{aligned}
&(\rho - 1)\mathrm{OPT}_a(\tau) + \mathrm{OPT}_b(\tau) - \mathrm{ALG}(\tau) + (\rho - 5)Dx \\
&\quad \le (\rho - 1)(\lambda x + iy) + Dx + iz - ((\lambda + D)x + iy) + (\rho - 5)Dx \\
&\quad \le \rho\{(3D-1)x + (2D - \lambda - 1)\delta\} - \{(9D-2)x + (2D - \lambda - 1)(2 - \gamma)\delta\}.
\end{aligned}
$$

Therefore, if $(\rho-1)\text{OPT}_a(\tau)+\text{OPT}_b(\tau)-\text{ALG}(\tau)+(\rho-5)Dx \geq 0$, then we obtain

$$\rho \geq 3+\frac{x-(2D-\lambda-1)(1+\gamma)\delta}{(3D-1)x+(2D-\lambda-1)\delta},$$

which is $3+\frac{\varepsilon}{O(D)}$ with $0<\varepsilon<1$ by setting $\gamma=O(1)$ and

$$\delta \leq \frac{(1-\varepsilon)x}{(2D-\lambda-1)(\gamma+1)} = O\left(\frac{x}{D}\right). \tag{17}$$

This means that there exists $\rho=3+\Omega(1/D)$ such that $(\rho-1)\text{OPT}_a(\tau)+\text{OPT}_b(\tau)-\text{ALG}(\tau)+(\rho-5)Dx<0$. Because $\text{OPT}_b(\tau)\geq Dx$, by Lemma 18, there exists $\rho'=3+\Omega(1/D)$ such that ϕ is an a-forcing sequence with $\text{ALG}(\phi)>\rho'\cdot\text{OPT}_a(\phi)$ or an arbitrarily costly sequence with $\text{ALG}(\phi)>\rho'\cdot\text{OPT}(\phi)$.

Case 3: $\phi=\tau\tau'$, where $\tau=b_a^{\lambda}c_b^{i}c^{+}$ with $i\geq 2D-\lambda$, and τ' is the sequence of clients generated in the states Lm17b and Lm17a. It follows that $\text{ALG}(\tau)\geq(\lambda+D)x+iy+(1+D)z$ and $\text{OPT}_c(\tau)\leq Dy+\lambda z$ (cost of moving the server to c first and keeping it at c). Thus, we have

$$\frac{\text{ALG}(\tau)}{\text{OPT}_c(\tau)} \geq \frac{(\lambda+D)x+iy+(1+D)z}{Dy+\lambda z} \geq \frac{3Dx+\{(2D-\lambda)+(1+D)\gamma\}\delta}{Dx+(D+\lambda\gamma)\delta}$$
$$=3+\frac{\{(\gamma-1)D+\gamma-\lambda(3\gamma+1)\}\delta}{Dx+(D+\lambda\gamma)\delta},$$

which is $3+\frac{\varepsilon}{\Theta(D)}$ with $0<\varepsilon<1$ by setting

$$\gamma := 4+3\varepsilon = O(1), \tag{18}$$
$$\lambda := \left\lfloor \frac{(\gamma-1-\varepsilon)D+\gamma}{3\gamma+1} \right\rfloor = \Theta(D), \text{ and} \tag{19}$$

$\delta=\Theta(x/D)$. It should be noted that $1\leq\lambda\leq D/3$ for $D\geq 3$. Because $\text{OPT}_c(\tau)\geq Dy$ and $\text{OPT}_b(\tau)\geq Dx$, by Lemma 17, there exists $\rho'=3+\Theta(1/D)$ such that ϕ is an a-forcing sequence with $\text{ALG}(\phi)>\rho'\cdot\text{OPT}_a(\phi)$ or an arbitrarily costly sequence with $\text{ALG}(\phi)>\rho'\cdot\text{OPT}(\phi)$.

Case 4: $\phi=b_a^{\lambda}c_c^{i}a^{+}$ with $i<D-\lambda$. It follows that $\text{ALG}(\phi)\geq\lambda x+(i+D+1+D)y=\lambda x+(i+2D+1)y$ and $\text{OPT}_a(\phi)\leq\lambda x+iy$ (cost of keeping the server at a). Thus, we have

$$\frac{\text{ALG}(\phi)}{\text{OPT}_a(\phi)} \geq \frac{\lambda x+(i+2D+1)y}{\lambda x+iy} \geq 1+\frac{(2D+1)y}{\lambda x+iy} > 1+\frac{2D+1}{D} = 3+\frac{1}{D}.$$

Case 5: $\phi = \tau\tau'$ where $\tau = b_a^{\lambda} c_c^i a_a^j$ with $D-\lambda \le i \le 2D-\lambda$ and $j \le 2D-1$, and τ' is the sequence of clients generated in the state Lm18. If ALG keeps the server at c during a^j, then the cost for a^j is $(j+D)y$. If ALG moves the server from c to b after the j'th request of a^j, then the cost for a^j is at least $j'y+Dz+(j-j'+D)x = jy+D(\gamma\delta+x)-(j-j')\delta$. Because $\gamma \ge 3$ and $j-j' < 2D$, this is at least $jy+D(3\delta+x)-2D\delta = jy+D(\delta+x) = (j+D)y$. Therefore, it follows that $\text{ALG}(\tau) \ge \lambda x+(i+D+j+D)y = \lambda x+(i+j+2D)y$. Moreover, $\text{OPT}_a(\tau) \le \lambda x + iy$ (cost of keeping the server at a), and $\text{OPT}_b(\tau) \le Dx+iz+jx = (j+D)x+iz$ (cost of moving the server to b first and keeping it at b). Thus, we have

$$\begin{aligned}
&(\rho-1)\text{OPT}_b(\tau)+\text{OPT}_a(\tau)-\text{ALG}(\tau)+(\rho-5)Dx \\
&\quad \le (\rho-1)((j+D)x+iz)+\lambda x+iy-(\lambda x+(i+j+2D)y)+(\rho-5)Dx \\
&\quad \le \rho\{(4D-1)x+(2D-\lambda)\gamma\delta\}-\{(12D-2)x+(4D-1+(2D-\lambda)\gamma)\delta\}.
\end{aligned}$$

To derive the second inequality, we have bounded j by $2D-1$ because j is multiplied by $(\rho-1)x-y \ge 2x-y \ge x+z-y > 0$ for $\rho \ge 3$. Therefore, if $(\rho-1)\text{OPT}_b(\tau)+\text{OPT}_a(\tau)-\text{ALG}(\tau)+(\rho-5)Dx \ge 0$, then we obtain

$$\rho \ge 3+\frac{x+((4D-1)-2(2D-\lambda)\gamma)\delta}{(4D-1)x+(2D-\lambda)\gamma\delta},$$

which is $3+\frac{\varepsilon}{O(D)}$ with $0<\varepsilon<1$ by setting $\gamma = O(1)$ and

$$\delta \le \frac{(1-\varepsilon)x}{2(2D-\lambda)\gamma-(4D-1)} = O\left(\frac{x}{D}\right). \tag{20}$$

This means that there exists $\rho = 3+\Omega(1/D)$ such that $(\rho-1)\text{OPT}_b(\tau)+\text{OPT}_a(\tau)-\text{ALG}(\tau)+(\rho-5)Dx < 0$. Because $\text{OPT}_a(\tau) > Dx$, by Lemma 18, there exists $\rho' = 3+\Omega(1/D)$ such that ϕ is an a-forcing sequence with $\text{ALG}(\phi) > \rho'\cdot\text{OPT}_a(\phi)$ or an arbitrarily costly sequence with $\text{ALG}(\phi) > \rho'\cdot\text{OPT}(\phi)$.

Case 6: $\phi = b_a^{\lambda} c_c^i a_a^j$ with $D-\lambda \le i \le 2D-\lambda$ and $j \ge 2D$. If ALG keeps the server at c during a^j, then the cost for a^j is $(j+D)y \ge 3Dy$. If ALG moves the server from c to b after the j'th request of a^j, then the cost for a^j is at least $j'y+Dz+(j-j'+D)x \ge jx+D(\gamma\delta+x)$. Because $\gamma \ge 3$ and $j \ge 2D$, this is at least $3D(\delta+x) = 3Dy$. Therefore, it follows that $\text{ALG}(\phi) \ge \lambda x+(i+D+3D)y = \lambda x+(i+4D)y$ and $\text{OPT}_a(\phi) \le \lambda x+iy$ (cost of keeping the server at a). Thus, we have

$$\begin{aligned}
\frac{\text{ALG}(\phi)}{\text{OPT}_a(\phi)} &\ge \frac{\lambda x+(i+4D)y}{\lambda x+iy} = 1+\frac{4Dy}{\lambda x+iy} \ge 1+\frac{4D(x+\delta)}{2Dx+(2D-\lambda)\delta} \\
&= 3+\frac{2\lambda\delta}{2Dx+(2D-\lambda)\delta},
\end{aligned}$$

which is $3+\Theta(1/D)$ by setting $\lambda = \Theta(D)$ and $\delta = \Theta(x/D)$.

Case 7: $\phi = \tau\tau'$, where $\tau = b_a^\lambda c_c^i$ with $i \geq 2D - \lambda + 1$, and τ' is the sequence of clients generated in the states Lm17b and Lm17a. It follows that $\text{ALG}(\tau) \geq \lambda x + (i+D)y$ and $\text{OPT}_c(\tau) \leq Dy + \lambda z$ (cost of moving the server to c first and keeping it at c). Thus, we have

$$\frac{\text{ALG}(\tau)}{\text{OPT}_c(\tau)} \geq \frac{\lambda x + (i+D)y}{Dy + \lambda z} \geq \frac{(3D+1)x + (3D-\lambda+1)\delta}{Dx + (D+\lambda\gamma)\delta} = 3 + \frac{x - ((3\gamma+1)\lambda - 1)\delta}{Dx + (D+\lambda\gamma)\delta},$$

which is $3 + \frac{\varepsilon}{O(D)}$ with $0 < \varepsilon < 1$ by setting $\gamma = O(1)$, $\lambda = \Theta(D)$, and

$$\delta \leq \frac{(1-\varepsilon)x}{(3\gamma+1)\lambda - 1} = O\left(\frac{x}{D}\right). \tag{21}$$

Because $\text{OPT}_c(\tau) \geq Dy$ and $\text{OPT}_b(\tau) \geq Dx$, by Lemma 17, there exists $\rho' = 3 + \Omega(1/D)$ such that ϕ is an a-forcing sequence with $\text{ALG}(\phi) > \rho' \cdot \text{OPT}_a(\phi)$ or an arbitrarily costly sequence with $\text{ALG}(\phi) > \rho' \cdot \text{OPT}(\phi)$.

By setting γ as in (18), λ as in (19), and δ so that (17), (20), (21), and $\delta \leq x/\gamma$ are satisfied, we can obtain a desired sequence ϕ. Thus, the proof of Theorem 4 is completed.

If we set $\varepsilon := 1/3$, $\gamma := 5$, $\lambda := \lfloor \frac{11D+15}{48} \rfloor$, and $\delta := \frac{x}{24D}$, then we can lower-bound $\frac{\text{ALG}(\tau)}{\text{OPT}_c(\tau)}$ by $3 + \frac{1}{72D+8}$ in Case 3. By applying Lemma 17 with $\beta = y/z = \frac{24D+1}{5}$ for the state Lm 17b, and then with $\beta = 1$ for the state Lm 17a, we obtain $\rho' > 3 + (360D + 340 + \frac{500}{24D+1})^{-1} > 3 + \frac{1}{360D+347}$ for $D \geq 3$, which is the smallest lower bound over all Cases 1–7.

6 Future Work

It would be interesting to answer whether or not there exists an asymptotically 3-competitive deterministic algorithm on a broader class of networks. Unfortunately, even 4-node ring networks do not allow WFA as it is to have such a competitive ratio. In fact, our proof of Theorem 1 depends on the fact that an extended work function is concave on the interval between two nodes on a continuous loop with three nodes (Claim 3 of Lemma 5). However, this fact does not follow on four nodes. On the other hand, there might exist a lower bound of $3 + \Theta(1)$ on general networks. For such a lower bound, however, we would need at least four nodes and have to overcome the difficulty of designing and analyzing a much more complicated adversary mainly due to increase of nodes. In any case, improving the currently best upper bound of 4.086 on general networks is still an important open problem.

References

1. Awerbuch, B., Bartal, Y., Fiat, A.: Distributed paging for general networks. J. Algorithms **28**(1), 67–104 (1998)
2. Awerbuch, B., Bartal, Y., Fiat, A.: Competitive distributed file allocation. Inf. Comput. **185**(1), 1–40 (2003)
3. Bartal, Y., Charikar, M., Indyk, P.: On page migration and other relaxed task systems. Theoret. Comput. Sci. **268**(1), 43–66 (2001)

4. Bartal, Y., Fiat, A., Rabani, Y.: Competitive algorithms for distributed data management. J. Comput. Sys. Sci. **51**(3), 341–358 (1995)
5. Bein, W.W., Chrobak, M., Larmore, L.L.: The 3-server problem in the plane. Theoret. Comput. Sci. **289**, 335–354 (2002)
6. Bienkowski, M.: Migrating and replicating data in networks. Comput. Sci. Res. Dev. **27**(3), 169–179 (2012)
7. Bienkowski, M., Byrka, J., Korzeniowski, M., Meyer auf der Heide, F.: Optimal algorithms for page migration in dynamic networks. J. Discrete Algorithms **7**(4), 545–569 (2009)
8. Black, D.L., Sleator, D.D.: Competitive algorithms for replication and migration problems. Tech. Rep. CMU-CS-89-201, Department of Computer Science, Carnegie Mellon University (1989)
9. Borodin, A., El-Yaniv, R.: Online Computation and Competitive Analysis. Cambridge University Press, Cambridge (1998)
10. Chrobak, M., Larmore, L.L., Reingold, N., Westbrook, J.: Page migration algorithms using work functions. J. Algorithms **24**(1), 124–157 (1997)
11. Karlin, A.R., Manasse, M.S., Rudolph, L., Sleator, D.D.: Competitive snoopy caching. Algorithmica **3**(1), 79–119 (1988)
12. Koutsoupias, E., Papadimitriou, C.H.: On the k-server conjecture. J. ACM **42**(5), 971–983 (1995)
13. Lund, C., Reingold, N., Westbrook, J., Yan, D.: Competitive on-line algorithms for distributed data management. SIAM J. Comput. **28**(3), 1086–1111 (1999)
14. Matsubayashi, A.: Uniform page migration on general networks. Int. J. Pure Appl. Math. **42**(2), 161–168 (2008)
15. Westbrook, J.: Randomized algorithms for multiprocessor page migration. SIAM J. Comput. **23**(5), 951–965 (1994)