3点上のオンラインページ移動問題に対する新しい上下界

松林 昭

金沢大学大学院自然科学研究科電子情報科学専攻
〒920-1192 金沢市角間町

概要　ページ移動問題とは，辺重み $w : E \to \mathbb{R}^+$ を持つグラフ $G = (V, E)$，正整数 D，点列 $s_0, r_1, \ldots, r_k \in V$ が与えられ，$\sum_{i=1}^{k}(d_{s_{i-1}r_i} + D \cdot d_{s_{i-1}s_i})$ を最小化するような点列 $s_1, \ldots, s_k \in V$ を求める問題である．ただし，d_{uv} は u と v を結ぶパスの辺の重みの最小和である．本報告ではこの問題に対する決定的なオンラインアルゴリズムについて考える．すべてのグラフと D に対して 3 より小さい競合比を持つアルゴリズムは存在しないことが知られている．また，3 点上で $D = 1$ の場合に対する 3-競合アルゴリズムが知られている．しかし，3 点上で $D \geq 2$ の場合に対する 3-競合オンラインアルゴリズムが存在するか否かは，著者が知る限り知られていない．本稿では，3 点上で $D = 2$ の場合には 3-競合アルゴリズムが存在するが，$D \geq 3$ の場合には 3-競合アルゴリズムは存在しないことを示す．また任意の D に対する 3.1467-競合オンラインアルゴリズムを示す．

New Bounds for Online Page Migration on Three Points

Akira Matsubayashi

Division of Electrical Engineering and Computer Science, Kanazawa University,
Kanazawa 920-1192, Japan

Abstract　The page migration problem is as follows: given a graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}^+$, a positive integer D, and nodes $s_0, r_1, \ldots, r_k \in V$, to compute $s_1, \ldots, s_k \in V$ so that the cost function $\sum_{i=1}^{k}(d_{s_{i-1}r_i} + D \cdot d_{s_{i-1}s_i})$ is minimized, where d_{uv} is the minimum sum of weights of edges of a path connecting u and v. We consider deterministic online algorithms for the problem. It is known that for any G and D, there exists no online algorithm with a competitive ratio less than 3. It is also known that there exists a 3-competitive algorithm on three points for $D = 1$. In this report, we prove that there exists a 3-competitive algorithm on three points if $D = 2$, while there exists no 3-competitive algorithm if $D \geq 3$. We also present a 3.1467-competitive algorithm on three points for any D.

1　Introduction

The problem of computing an efficient dynamic allocation of data objects stored in nodes of a network so that the costs to serve requests for the data is minimized commonly arises in network applications such as memory management in a shared memory multiprocessor system and Peer-to-Peer applications on the Internet. In this report, we study one of the variations of the problem, called *the page migration problem*, in which requests are to be served using unicast communication, and we are allowed to migrate data objects, i.e, no replication is allowed. The objective function to be minimized is the total sum of the service cost for each request, which is the distance between the server and client nodes, and of the management cost for each migration, which is the migration distance multiplied by the data size.

We consider deterministic online page migration algorithms. Randomized algorithms have been investigated in, e.g., [2, 4, 5, 7]. Black and Sleator presented 3-competitive deterministic algorithms for trees, uniform networks, and products of those networks, including grids and hypercubes [3]. The tightness of the competitive ratio of 3 was also shown in [3] by proving that no deterministic algorithm has a competitive ratio less than 3 even for one link networks. The current best upper and lower bounds for general networks are 4.086 [1] and 3.1639 [6], respectively. The upper bound was improved to $2 + \sqrt{2}$ for the case that the data size D is 1. It was mentioned in [4] that the lower bound is more than 3 even for 4-node networks. A lower bound of 3.1213 for five node networks was presented in [6]. We can obtain an explicit lower bound of 3.1062 for 4-node networks using a similar argument. For

3-node networks, a 3-competitive algorithm was presented in [4] for the case of $D = 1$. To the author's knowledge, however, it has not been known whether there exists a 3-competitive algorithm on 3-node networks for $D \geq 2$. In this report, we answer the question by providing a lower bound of $3 + \Omega(\frac{1}{D^2})$ for any $D \geq 3$ and a 3-competitive algorithm for $D \leq 2$. The algorithm achieves a competitive ratio of 3.1467 for any D.

2 Preliminaries

We suppose that graphs considered here have nodes $V = \{a, b, c\}$ and edge weights x, y, and z for edges (a, b), (a, c), and (b, c), respectively. We assume without loss of generality that $y \geq \max\{x, z\}$. The distance between two nodes u and v, denoted by d_{uv}, is the minimum sum of weights of edges of a path connecting u and v.

The page migration problem is as follows: given a graph G, a node s_0 of G that initially holds a page of size D, and a sequence $r_1, \ldots, r_k$ of nodes of G that issue a request for access to the page, to compute a sequence $s_1, \ldots, s_k$ of nodes of G that hold the page so that the cost function $\sum_{i=1}^{k} d_{s_{i-1}r_i} + D \cdot d_{s_{i-1}s_i}$ is minimized. We call nodes s_i and r_i a *server* and a *client*. An *online* page migration algorithm determines s_i without knowing $r_{i+1}, \ldots, r_k$. We denote by $A(\sigma)$ the cost of a page migration algorithm A for a sequence $\sigma = (r_1, \ldots, r_k)$. An online page migration algorithm ALG is *ρ-competitive* if there exists a value α independent of k such that $\text{ALG}(\sigma) \leq \rho \cdot \text{OPT}(\sigma) + \alpha$ for an optimal offline algorithm OPT and any σ.

3 Lower Bound

In this section we show the following theorem:

Theorem 1 *There exists no deterministic ρ-competitive page migration algorithm on 3-node networks if $\rho = 3 + o(\frac{1}{D^2})$ and $D \geq 3$.*

Assume in this section that $x \geq z$. Let ALG be a deterministic online page migration algorithm. Let $\text{OPT}_u(\sigma)$ be the optimal offline cost to leave the last server on u after processing a sequence σ of clients. We write σ as σ_v if ALG leaves the last server on v after processing σ.

Lemma 1 *Let U and V be disjoint sets of nodes, and $u \in U$ and $v \in V$ be joined by an edge with the minimum weight w overall edges joining U and V. If there exist $\rho > 3$ and a sequence σ_v of clients such that $(\rho - 1)\text{OPT}_u(\sigma_v) + \text{OPT}_v(\sigma_v) - \text{ALG}(\sigma_v) + (\rho - 5)Dw < 0$, then there exists a sequence σ' with $\text{ALG}(\sigma_v\sigma'_u) > \rho\text{OPT}_u(\sigma_v\sigma')$ or $\text{ALG}(\sigma_v\sigma'_v) > \rho\text{OPT}_v(\sigma_v\sigma')$.*

Proof We prove that $\sigma'_u := u^{k_1}v^{l_1}\ldots u^{k_{i-1}}v^{l_{i-1}}u^{k_i}u^+$ or $\sigma'_v := u^{k_1}v^{l_1}\ldots u^{k_i}v^{l_i}v^+$ is a desired sequence for some i. Here, u^k (resp. v^l) is a sequence of k (resp. l) requests from u (resp. v) after which ALG moves the server from a node of V (resp. U). In addition, u^+ (resp. v^+) is a sequence of requests from u (resp. v) until ALG locates the server on u (resp. v).

Assume for contradiction that $\text{ALG}(\sigma_v\sigma'_u) \leq \rho\text{OPT}_u(\sigma_v\sigma'_u)$ and $\text{ALG}(\sigma_v\sigma'_v) \leq \rho\text{OPT}_v(\sigma_v\sigma'_v)$. It follows that $\text{ALG}(u^{k_i}) \geq (k_i + D)w$, $\text{ALG}(v^{l_i}) \geq (l_i + D)w$, $\text{OPT}_u(\sigma_v\sigma'_u) \leq \text{OPT}_u(\sigma_v) + L_{i-1}w$, and $\text{OPT}_v(\sigma_v\sigma'_v) \leq \text{OPT}_v(\sigma_v) + K_iw$, where $L_{i-1} := \sum_{j=1}^{i-1} l_j$ and $K_i := \sum_{j=1}^{i} k_j$. By the inequalities, we have $\text{ALG}(\sigma_v) + (K_i + Di)w + (L_{i-1} + D(i-1)) \leq \rho(\text{OPT}_u(\sigma_v) + L_{i-1}w)$ and $\text{ALG}(\sigma_v) + (K_i + Di)w + (L_i + Di)w \leq \rho(\text{OPT}_v(\sigma_v) + K_iw)$. Therefore, it follows that $K_i \leq (\rho - 1)L_{i-1} - D(2i - 1) + A$ and $L_i \leq (\rho - 1)K_{i-1} - 2Di + B$, where $A := \frac{\rho\text{OPT}_u(\sigma_v) - \text{ALG}(\sigma_v)}{w}$ and $B := \frac{\rho\text{OPT}_v(\sigma_v) - \text{ALG}(\sigma_v)}{w}$. It follows from the recurrences that $K_i \leq (\rho - 1)^2K_{i-1} - 2\rho Di + (2\rho - 1)D + A + (\rho - 1)B \leq \frac{(\rho-1)^{2i-1}}{\rho(\rho-2)}(-\frac{\rho(\rho-1)}{\rho-2}D + (\rho - 1)A + B) + \frac{2Di}{\rho-2} + \frac{\frac{\rho}{\rho-2}D - A - (\rho-1)B}{\rho(\rho-2)}$. The coefficient of $\frac{(\rho-1)^{2i-1}}{\rho(\rho-2)}$ can be estimated as follows: $-\frac{\rho(\rho-1)}{\rho-2}D + (\rho - 1)A + B = \frac{\rho}{w}((\rho - 1)\text{OPT}_u(\sigma_v) + \text{OPT}_v(\sigma_v) - \text{ALG}(\sigma_v) - \frac{\rho-1}{\rho-2}Dw)$, 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. □

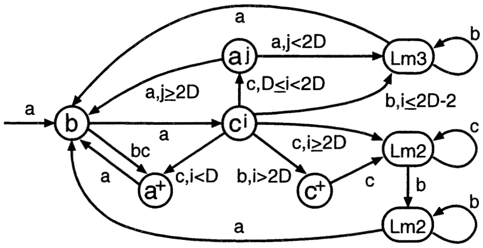

Figure 1: Strategy for σ.

Lemma 2 *Let $u := a$ and $v := b$, or $u := b$ and $v := c$. Let w be the weight of the edge (u, v). If there exist $\rho > 3$, $\beta > 0$, and a sequence σ_v of clients such that $\text{ALG}(\sigma_v) > \rho\text{OPT}_v(\sigma_v)$ and $\text{OPT}_v(\sigma_v) \geq \beta Dw$, then there exists a sequence σ' such that $\text{ALG}(\sigma_v\sigma'_u) > \rho'\text{OPT}_u(\sigma_v\sigma')$, or σ' is an arbitrarily long sequence with $\text{ALG}(\sigma_v\sigma') > \rho'\text{OPT}(\sigma_v\sigma')$, where $\rho' = 1 + \frac{\beta\rho + \sqrt{\beta^2\rho^2 - 4(\beta+1)(\beta\rho-\beta-4)}}{2(\beta+1)} > \frac{\beta}{\beta+4}(\rho - 3) + 3$.*

Proof We define σ' as follows:

1. Let $\tau^0 := \epsilon$, i.e., an empty sequence, and $j = 1$.
2. Input u^i to ALG, i.e., generate i requests from u until ALG locates the server on u. It should be noted that wherever ALG moves the server during the requests, $\text{ALG}(u^i) \geq (i + D)w$. This is because w is at most the weight between u and $p \notin \{u, v\}$ by $y \geq x \geq z$.
3. If $i \geq ((\beta + 1)\rho' - \beta\rho - 1)D$, then set $\sigma' := \tau^0 \cdots \tau^{j-1}u^i$, and quit the procedure.
4. Otherwise, because $\text{ALG}_u(u^i) \geq (i + D)w$, $\text{OPT}_u(u^i) \leq Dw$, and $\text{OPT}_v(u^i) \leq iw$, it follows that $(\rho' - 1)\text{OPT}_v(u^i) + \text{OPT}_u(u^i) - \text{ALG}(u^i) + (\rho' - 5)Dw \leq (\rho' - 1)iw + Dw - (i + D)w + (\rho' - 5)Dw < \{(\rho' - 2)((\beta + 1)\rho' - \beta\rho - 1) + \rho' - 5\}Dw = \{(\beta + 1)\rho'^2 - (\beta\rho + 2(\beta + 1))\rho' + 2\beta\rho - 3\}Dw$, which equals 0 by the definition of ρ'. Therefore, by Lemma 1, there exists a sequence τ^j containing u^i such that $\text{ALG}(\tau^j_u) > \rho'\text{OPT}_u(\tau^j)$ or $\text{ALG}(\tau^j_v) > \rho'\text{OPT}_v(\tau^j)$.
5. If $\tau^j = \tau^j_u$, then set $\sigma' := \tau^0 \cdots \tau^j$, and quit the procedure. Otherwise, set $j := j + 1$, and repeat the process from Step 2.

By definition, σ' is σ'_u or arbitrarily long. Because $((\beta+2)\rho - 2(\beta+1))^2 - (\beta^2\rho^2 - 4(\beta+1)(\beta\rho - \beta - 4)) = 4(\beta + 1)(\rho + 1)(\rho - 3) > 0$, it follows that $\rho' < 1 + \frac{\beta\rho + (\beta+2)\rho - 2(\beta+1)}{2(\beta+1)} = \rho$. Thus, if the procedure ends in Step 2, then it follows that $\text{ALG}(\sigma_v\sigma'_u) - \rho'\text{OPT}(\sigma_v\sigma') \geq \{\text{ALG}(\sigma_v) + \sum_{j\geq 0}\text{ALG}(\tau^j) + \text{ALG}(u^i)\} - \{\text{OPT}_v(\sigma_v) + \sum_{j\geq 0}\text{OPT}_v(\tau^j) + \text{OPT}_u(u^i)\} > (\rho - \rho')\text{OPT}_v(\sigma_v) + ((\beta + 1)\rho' - \beta\rho)Dw - \rho' Dw \geq (\rho - \rho')\beta Dw + \beta(\rho' - \rho)Dw = 0$. Similarly, if the procedure ends in Step 5, then $\text{ALG}(\sigma_v\sigma'_u) - \rho'\text{OPT}(\sigma_v\sigma') > (\rho - \rho')\text{OPT}_v(\sigma_v) > 0$. □

Lemma 3 *Let $\{u, v\} := \{a, b\}$, and w be the weight of the edge (u, v). If there exist $\rho > 3$, $\beta > 0$, and a sequence σ_v of clients such that $(\rho - 1)\text{OPT}_u(\sigma_v) + \text{OPT}_v(\sigma_v) - \text{ALG}(\sigma_v) + (\rho - 5)Dw < 0$ and $\text{OPT}_v(\sigma_v) \geq \beta Dw$, then there exists a sequence σ' such that $\text{ALG}(\sigma_v\sigma'_a) > \rho'\text{OPT}_a(\sigma_v\sigma')$, or σ' is an arbitrarily long sequence with $\text{ALG}(\sigma_v\sigma') > \rho'\text{OPT}(\sigma_v\sigma')$, where $\rho' = \frac{\beta}{\beta+4}(\rho - 3) + 3$.*

Proof By Lemma 1, there exists a sequence τ with $\text{ALG}(\sigma_v\tau_a) > \rho\text{OPT}_a(\sigma_v\tau)$ or $\text{ALG}(\sigma_v\tau_b) > \rho\text{OPT}_b(\sigma_v\tau)$. If $\tau = \tau_a$, then we have obtained a desired sequence. Otherwise, by Lemma 2, there exists a sequence τ' such that $\text{ALG}(\sigma_v\tau_b\tau'_a) > \rho'\text{OPT}_a(\sigma_v\tau_b\tau')$, or τ' is an arbitrarily long sequence with $\text{ALG}(\sigma_v\tau_b\tau') > \rho'\text{OPT}(\sigma_v\tau_b\tau')$. Therefore, $\tau\tau'$ is a desired sequence. □

Suppose that $y = x + \delta$ and $z = \gamma\delta$ with $3 \leq \gamma < \frac{x}{\delta}$. To prove Theorem 1, we carefully choose ρ, γ, and δ, and design a strategy to generate an arbitrarily long sequence σ with $\text{ALG}(\sigma) > \rho\text{OPT}(\sigma)$. This proves that $\text{ALG}(\sigma) \geq \rho\text{OPT}(\sigma) + \alpha$ for any α independent of the number of clients. The strategy is illustrated using an automaton as shown in Fig. 1. Here, a transition and a state represent a server selected by ALG and a sequence of clients given to ALG after the selection, respectively. We set the initial server on a and generate the first request from b. If ALG moves the server to b or c after it served the first request, then we generate a^+. Otherwise, we generate c^i, which is a sequence of requests from c after which ALG moves the server from a to b or c. Outgoing arcs from c^i have additional conditions of i to be transited. For example, if ALG moves the server from a to c after it served less than i requests from c, then we generate a^+. The state a^j represents a sequence of requests from a until ALG locates the server on a. It should be noted that i in c^i is the number to force ALG to *move* the server somewhere else, while j in a^j is the number to force ALG to *locates* the server on a. The state Lm3 represents a sequence obtained by applying Lemma 3, which forces ALG to move the server to a or is an arbitrarily long. Similarly, two states Lm2 represents a sequence obtained by applying Lemma 3 twice, which forces ALG to move the server to b, and then to a, or is arbitrarily long.

Let σ be a sequence of clients that ends with the initial state or transits to a state of Lm2 or Lm3.

Case 1: $\sigma = b_{bc}a^+$. It follows that $\text{ALG}(\sigma) \geq 2(1+D)x$ and $\text{OPT}_a(\sigma) = x$. Therefore, $\frac{\text{ALG}(\sigma)}{\text{OPT}_a(\sigma)} \geq D+2 \geq 5$.

Case 2: $\sigma = b_a c_b^{i \leq 2D-2}$. It follows that $\text{ALG}(\sigma) = (1+D)x + iy$, $\text{OPT}_a(\sigma) = x + iy$, and $Dx \leq \text{OPT}_b(\sigma) \leq Dx + iz$. Thus, it follows that $(\rho-1)\text{OPT}_a(\sigma) + \text{OPT}_b(\sigma) - \text{ALG}(\sigma) + (\rho-5)Dx \leq \rho((D+1)x + (2D-2)y) - ((5D+2)x + 2(2D-2)y - (2D-2)z)$. Hence, if $(\rho-1)\text{OPT}_a(\sigma) + \text{OPT}_b(\sigma) - \text{ALG}(\sigma) + (\rho-5)Dx \geq 0$, then we can obtain $\rho \geq \frac{(5D+2)x+2(2D-2)y-(2D-2)z}{(D+1)x+(2D-2)y} = 3 + \frac{x-(1+\gamma)(2D-2)\delta}{(3D-1)x+(2D-2)\delta}$, which can be greater than $3+\Omega(\frac{1}{D})$ by setting δ to a sufficiently small value such that $\gamma\delta = \Theta(\frac{1}{D})$. By Lemma 3, this means that there exists a sequence τ such that $\text{ALG}(\sigma\tau_a) \geq \rho\text{OPT}_a(\sigma\tau)$, or τ is an arbitrarily long sequence with $\text{ALG}(\sigma\tau) \geq \rho\text{OPT}(\sigma\tau)$ with $\rho = 3 + \Omega(\frac{1}{D})$.

Case 3: $\sigma = b_a c_b^{i \geq 2D-1} c^+$. It follows that $\text{ALG}(\sigma) \geq (1+D)x + iy + (1+D)z$ and $Dy \leq \text{OPT}_c(\sigma) \leq Dy + z$. Therefore, we have $\frac{\text{ALG}(\sigma)}{\text{OPT}_c(\sigma)} \geq \frac{(1+D)x+(2D-1)y+(1+D)z}{Dy+z} = 3 + \frac{(\gamma(D-2)-(D+1))\delta}{Dx+(D+\gamma)\delta}$, which can be greater than $3+\Omega(\frac{1}{D})$ by setting $\frac{D+1}{D-2} < \gamma := \Theta(1)$ and $\delta := \Theta(\frac{1}{D})$. By applying Lemma 2 twice, we can obtain a sequence τ such that $\text{ALG}(\sigma\tau_a) \geq \rho\text{OPT}_a(\sigma\tau)$, or τ is an arbitrarily long sequence with $\text{ALG}(\sigma\tau) \geq \rho\text{OPT}(\sigma\tau)$ with $\rho = 3 + \Omega(\frac{1}{D})$.

Case 4: $\sigma = b_a c_c^{i \leq D-1} a^+$. It follows that $\text{ALG}(\sigma) \geq x + (i+D+1+D)y = x + (i+2D+1)y$ and $\text{OPT}_a(\sigma) \leq x + iy$. Therefore, $\frac{\text{ALG}(\sigma)}{\text{OPT}_a(\sigma)} \geq \frac{x+(i+2D+1)y}{x+iy} \geq 1 + \frac{(2D+1)y}{x+iy} > 1 + \frac{2D+1}{D} = 3 + \frac{1}{D}$.

Case 5: $\sigma = b_a c_c^{D \leq i \leq 2D-1} a_a^{j \leq 2D-1}$. If ALG moves the server from c to b in the j'th request of a^j, then the cost for a^j is at least $(j-j')y + Dz + (j'+D)x = jy + D(\gamma\delta + x) - j'\delta$. Because $\gamma \geq 3$ and $j' < 2D$, this is at least $jy + D(3\delta + x) - 2D\delta = jy + D(\delta + x) = (j+D)y$, which is the cost that ALG moves the server from c to a after a^j. Therefore, it follows that $\text{ALG}(\sigma) \geq x + (i+D+j+D)y = x + (i+j+2D)y$, $Dy < \text{OPT}_a(\sigma) \leq x + iy$, and $\text{OPT}_b(\sigma) \leq Dx + iz + jx = (j+D)x + iz$. Thus, it follows that $(\rho-1)\text{OPT}_b(\sigma) + \text{OPT}_a(\sigma) - \text{ALG}(\sigma) + (\rho-5)Dx \leq \rho((4D-1)x + (2D-1)z) - ((8D-1)x + (4D-1)y + (2D-1)z)$. Hence, if $(\rho-1)\text{OPT}_b(\sigma) + \text{OPT}_a(\sigma) - \text{ALG}(\sigma) + (\rho-5)Dx \geq 0$, then we can obtain $\rho \geq \frac{(8D-1)x+(4D-1)y+(2D-1)z}{(4D-1)x+(2D-1)z} = 3 + \frac{x+((4D-1)-2\gamma(2D-1))\delta}{(4D-1)x+\gamma(2D-1)\delta}$, which can be greater than $3+\Omega(\frac{1}{D})$ by setting δ to a sufficiently small value such that $\gamma = \Theta(1)$ and $\delta = \Theta(\frac{1}{D})$. By Lemma 3, this means that there exists a sequence τ such that $\text{ALG}(\sigma\tau_a) \geq \rho\text{OPT}_a(\sigma\tau)$, or τ is an arbitrarily long sequence with $\text{ALG}(\sigma\tau) \geq \rho\text{OPT}(\sigma\tau)$ with $\rho = 3 + \Omega(\frac{1}{D})$.

Case 6: $\sigma = b_a c_c^{D \leq i \leq 2D-1} a_a^{j \geq 2D}$. If ALG moves the server from c to b in the j'th request of a^j, then the cost for a^j is at least $(j-j')y + Dz + (j'+D)x = jy + D(\gamma\delta + x) - j'\delta \geq jx + D(\gamma\delta + x)$. Because $\gamma \geq 3$ and $j \geq 2D$, this is at least $3D(\delta + x) = 3Dy$. which is the cost that ALG moves the server from c to a before a^j. Therefore, it follows that $\text{ALG}(\sigma) \geq x + (i+D+3D)y = x + (i+4D)y$ and $\text{OPT}_a(\sigma) \leq x + iy$. Thus, it follows that $\frac{\text{ALG}(\sigma)}{\text{OPT}_a(\sigma)} \geq 1 + \frac{4Dy}{x+iy} \geq 3 + \frac{\delta}{2Dx+(2D-1)\delta}$, which can be greater than $3+\Omega(\frac{1}{D^2})$ by setting $\delta = \Theta(\frac{1}{D})$.

Case 7: $\sigma = b_a c_c^{i \geq 2D}$. It follows that $\textsc{Alg}(\sigma) \geq x+(i+D)y$ and $Dy \leq \textsc{Opt}_c(\sigma) \leq Dy+z$. Therefore, we have $\frac{\textsc{Alg}(\sigma)}{\textsc{Opt}_c(\sigma)} \geq \frac{x+(i+D)y}{Dy+z} \geq \frac{x+3Dy}{Dy+z} = 3+\frac{x-3\gamma\delta}{Dx+(D+\gamma)\delta}$, which can be greater than $3+\Omega(\frac{1}{D})$ by setting δ to a sufficiently small value such that $\gamma\delta = \Theta(1)$. By applying Lemma 2 twice, we can obtain a sequence τ such that $\textsc{Alg}(\sigma\tau_a) \geq \rho\textsc{Opt}_a(\sigma\tau)$, or τ is an arbitrarily long sequence with $\textsc{Alg}(\sigma\tau) \geq \rho\textsc{Opt}(\sigma\tau)$ with $\rho = 3 + \Omega(\frac{1}{D})$.

Therefore, the proof of Theorem 1 is completed.

4 Algorithm

In this section we show the following theorems by providing a desired algorithm:

Theorem 2 *There exists a* 3.1467*-competitive deterministic page migration algorithm on* 3*-node networks.*

Theorem 3 *There exists a* 3*-competitive deterministic page migration algorithm on* 3*-node networks if* $D \leq 2$.

4.1 Definition

To describe our algorithm, called PM3, we need some notations as follows: Let $\rho \geq 3$ and $L = x+y+z$. Let A be the set of nodes v incident to edges with weights w and w' such that $w \leq \frac{L}{2} - \frac{w'}{\rho}$ and $w' \leq \frac{L}{2} - \frac{w}{\rho}$, and $B := V \setminus A$. By the assumption that $y \geq \{x, z\}$, it follows that $b \in A$. This is because that for $\{e, f\} = \{x, z\}$, $e - (\frac{L}{2} - \frac{f}{\rho}) = e - (\frac{L}{2} - \frac{L-e-y}{\rho}) \leq (1 - \frac{2}{\rho})(e - \frac{L}{2}) \leq 0$. We assume without loss of generality that $B \in \{\emptyset, \{a\}, \{a, b\}\}$. PM3 has a counter $C_v \geq 0$ for each node v so that $\sum_{v \in V} C_v = 2D$. We define a function Φ of counters and the servers of PM3 and $\textsc{Opt}$ on s and t, respectively, so that

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

We divide the input sequence of clients into phases so that a migration of PM3 ends the current phase. When a new phase begins, PM3 sets counter of the previous server to 0. We define another function Ψ of a phase beginning with the servers of PM and $\textsc{Opt}$ on s and t, as follows: If $B = \emptyset$, then $\Psi_{st} := 0$ for $s, t \in V$. Otherwise,

- $\Psi_{st} := 0$ for $\{s, t\} = \{a, c\}$,
- $\Psi_{bt} := \max\{C_{\bar{t}}(-(\frac{\rho}{2} - 1)d_{b\bar{t}} - \frac{\rho}{2}(d_{t\bar{t}} - d_{bt})), C_b\frac{\rho}{2}(d_{b\bar{t}} - d_{tb} - d_{t\bar{t}}) - (\rho - 3)Dd_{b\bar{t}})\}$, and
- $\Psi_{bb} := \Psi_{b\bar{t}} := \max\{C_{\bar{t}}(-(\frac{\rho}{2} - 1)d_{b\bar{t}} - \frac{\rho}{2}(d_{t\bar{t}} - d_{bt})), -(\rho - 3)Dd_{b\bar{t}})\}$ for $\{t, \bar{t}\} = \{a, c\}$ with $C_t = 0$.

If a request is issued from a node r, then PM3 serves the request and performs as follows:

Algorithm PM3

1. If $r = s$, i.e., the current server, then PM3 performs nothing.
2. If $r \neq s$, $s \in A$, and $C_{\bar{r}} \geq 1$, then $C_{\bar{r}}{-}{-}$ and $C_r{+}{+}$ for $\bar{r} \in V - \{s, r\}$. Otherwise, $C_s{-}{-}$ and $C_r{+}{+}$.
3. If $C_s = 0$, then:
   (a) If $s \in A$, then $C_r = 2D$ and move the server to r.
   (b) Otherwise, let $\bar{s} \in V - \{s, b\}$. Recall that $b \notin B$, $\{s, \bar{s}\} = \{a, c\}$, and $y = d_{s\bar{s}}$.
   (c) Move the server to b if $F_b \leq F_{\bar{s}}$.
   (d) Move the server to $\bar{s}$ if $F_b > F_{\bar{s}}$, and set counters of b and $\bar{s}$ to 0 and $2D$, respectively.

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

- $F_p := \max_{t,q\in V}\{M_{pq} + S_q + \Psi_{pq} - \Psi'_{st}\}$,

- $M_{bq} := -(\rho-3)Dd_{sb} + (\rho-2)C_{\bar{s}}(\frac{L}{2} - d_{s\bar{s}})$ for $q \in V$,
- $M_{\bar{s}q} := -(\rho-3)Dd_{s\bar{s}} + C_b((\frac{\rho}{2}-1)(d_{s\bar{s}} - d_{sb}) + \frac{\rho}{2}(d_{\bar{s}q} - d_{bq}))$ for $q \in V$,
- $S_s := 0$, and
- $S_q := \max\{-\rho C_{\bar{s}}(\frac{L}{2} - d_{s\bar{s}}), -\rho C_b(\frac{L}{2} - d_{sb}), -\rho C_b(\frac{L}{2} - d_{b\bar{s}})\}$ for $q \in \{b, \bar{s}\}$.

We use Ψ' to denote it is associated with counters at the beginning of the current phase. It should be noted that Ψ_{pq}, which will possibly be that of the next phase, can be computed just before a migration. This is because PM3 does not change counters if $p = b$, and because $\Psi_{\bar{s}q}$ depends on no counters.

4.2 Competitiveness

For any event e, let ΔPM3(e) and ΔOPT(e) be the costs of PM3 and OPT, respectively, for e. Moreover, let $\Delta\Phi(e)$ be the amount of change of Φ for e. Furthermore, let $f(e) := \Delta\text{PM3}(e) + \Delta\Phi(e) - \rho\Delta\text{OPT}(e)$. We will omit e in the notations if e is clear from the context.

By definition, we can observe following facts:

Lemma 4

1. *If PM3 moves the server to p in Step 3(a) or 3(c), then $f = -(\rho-3)Dd_{sp} + (\rho-2)C_{\bar{p}}(\frac{L}{2} - d_{s\bar{p}})$, where $\bar{p} \in V \setminus \{s, p\}$. If PM3 moves the server from s to p in Step 3(d), then $f = -(\rho-3)Dd_{sp} + C_{\bar{p}}((\frac{\rho}{2}-1)(d_{sp} - d_{s\bar{p}}) + \frac{\rho}{2}(d_{pq} - d_{\bar{p}q}))$. In particular, if $C_p = 2D$ (i.e., $C_{\bar{p}} = 0$), then $f = -(\rho-3)Dd_{sp}$.*
2. *If OPT moves the server from t to q, then $f = \frac{\rho}{2}\sum_{v\in V} C_v(d_{qv} - d_{tv} - d_{tq}) \le 0$.*
3. *Suppose that $r \in V$ issues a request, and that PM3 and OPT locates the servers on s and t, respectively. If $r = s$, then $f = -\rho d_{rt} \le 0$. If $r \ne s$, $s \in A$, and $C_{\bar{r}} \ge 1$, then $f = \rho d_{rs} + d_{\bar{r}s} - \rho\frac{L}{2} - \frac{\rho}{2}(d_{rt} + d_{\bar{r}t} - d_{r\bar{r}}) \le 0$. Otherwise, $f = \frac{\rho}{2}(d_{rs} - d_{rt} - d_{st}) \le 0$.* □

Fix a phase. Suppose that PM3 and OPT locates the servers s and t, respectively, at the beginning of the phase, and on p and q, respectively, at the end of the phase. We will prove $g := f + \Psi_{pq} - \Psi'_{st} \le 0$ for the phase. If this holds, then by summing up the inequality overall the phases, we can prove that PM3 is ρ-competitive.

If $B = \emptyset$ or $s = c \notin B$, then $s \in A$ and $\Psi_{st} = 0$, and hence, $g = f \le 0$ for $\rho \ge 3$ by Lemma 4. In what follows, C_v is the value of counter of $v \in V$ just before PM3 moves the server to p. This means that $C_s = 0$.

Lemma 5 *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. By the definition of PM3, C'_a or C'_c is 0. We may assume without loss of generality that $C'_a = 0$. Because $b \in A$, either C_a or C_c is $2D$, i.e., $p \in \{a, c\}$. If $p = c$, then $f \le -(\rho-3)Dz$ by Lemma 4. If $p = a$, then an amount of at least C'_c of c's counter must move to a in the phase. Thus, $f \le C'_c(\frac{\rho}{2}x - (\frac{\rho}{2}-1)z - \frac{\rho}{2}y)$ by Lemma 4. Therefore, $g \le f - \Psi'_{bt} \le 0$ if $t \in \{b, c\}$ or $p = a$.

It remains to prove the lemma for the case of $t = a$ and $p = c$. An amount of at least C'_b of b's counter must move to c by the definition of PM3. Thus, if OPT does not move the server during the phase, then $f \le C'_b\frac{\rho}{2}(z - x - y) - (\rho-3)Dz$ by Lemma 4. Therefore, $g \le f - \Psi'_{ba} \le 0$. If OPT moves the server from a to c after an amount $\delta \le C'_c$ of c's counter moved to a, then $f \le \delta(-(\frac{\rho}{2}-1)z - \frac{\rho}{2}(y-x)) - \rho(C'_c - \delta)y = \delta(-(\frac{\rho}{2}-1)z + \frac{\rho}{2}(x+y)) - \rho C'_c y \le C'_c(-(\frac{\rho}{2}-1)z - \frac{\rho}{2}(y-x))$. Therefore, $g \le f - \Psi'_{bt} \le 0$ for $t \in V$. If OPT moves the server from a to b after an amount δ of b's and/or c's counters moved to a, then $f \le \delta(-(\frac{\rho}{2}-1)z - \frac{\rho}{2}(y-x)) \le C'_c(-(\frac{\rho}{2}-1)z - \frac{\rho}{2}(y-x))$ if $\delta \ge C'_c$. If $\delta < C'_c$, then $f \le \delta(-(\frac{\rho}{2}-1)z - \frac{\rho}{2}(y-x)) + \frac{\rho}{2}(-2C'_b x + (C'_c - \delta)(z+y-x)) - (\rho-3)Dz = \delta(-(\rho-1)z + \rho x) + \frac{\rho}{2}(-(2C'_b + C'_c)x - C'_c(y-z)) - (\rho-3)Dz$, which is at most $C'_c(-(\frac{\rho}{2}-1)z - \frac{\rho}{2}(y-x))$ if $z \le \frac{\rho x}{\rho-1}$, at most $\frac{\rho}{2}(-(2C'_b + C'_c)x - C'_c(y-z)) - (\rho-3)Dz \le \frac{\rho}{2}(-(C'_b + C'_c)(y - z + x)) - (\rho-3)Dz = 2D\frac{\rho}{2}((z - x - y)) - (\rho-3)Dz$ if $z > \frac{\rho x}{\rho-1}$. Therefore, $g \le f - \Psi'_{bt} \le 0$ for $t \in V$. □

Lemma 6 *If $s \in B$, then $f \leq M_{pq} + S_q$ for $p \in \{b, \bar{s}\}$ and $t, q \in V$.*

Proof We may assume without loss of generality that $s = a$. Let f^M and f^S be the amounts of change of f in the move of PM3 and the other events, respectively, in the phase. It follows from Lemma 4 that $f^M = M_{pq}$ and $f^S \leq 0 = S_a$. Thus, it suffices to prove that $f^S \leq S_q = \max\{-\rho C_c(\frac{L}{2} - y), -\rho C_b(\frac{L}{2} - x), -\rho C_b(\frac{L}{2} - z)\}$ for $q \in \{b, c\}$.

Suppose that OPT moves the server from u to $v \neq u$ after amounts of δ_b and δ_c of a's counter to b and c, respectively. It should be noted that counters of b and c never decrease in the phase by the definition of PM3.

Let f_{uv} be the amount of change of f for the move of OPT. By Lemma 4,

- $f_{av} = \frac{\rho}{2}(-2\delta_v d_{av} + \delta_{\bar{v}}(z - x - y))$ for $\{v, \bar{v}\} = \{b, c\}$,
- $f_{uv} = \frac{\rho}{2}(-2\delta_v z + (2D - \delta_b - \delta_c)(d_{av} - d_{au} - z)) = \frac{\rho}{2}((2D - \delta_u)(d_{av} - d_{au} - z) - \delta_v(d_{av} - d_{au} + z)) \leq \frac{\rho}{2}(2D - \delta_u)(d_{av} - d_{au} - z) \leq -\rho C_v(\frac{L}{2} - d_{av})$ for $\{u, v\} = \{b, c\}$, and
- $f_{ua} = \frac{\rho}{2}(-2(2D - \delta_b - \delta_c)d_{au} + \delta_{\bar{u}}(d_{a\bar{u}} - z - d_{au})) = \frac{\rho}{2}((2D - \delta_u)(d_{a\bar{u}} - z - d_{au}) - (2D - \delta_b - \delta_c)(d_{a\bar{u}} - z + d_{au})) \leq -\rho C_{\bar{u}}(\frac{L}{2} - d_{a\bar{u}})$ for $\{u, \bar{u}\} = \{b, c\}$.

Moreover, if OPT resides at q after amounts of δ_b and δ_c of a's counter moved to b and c, respectively, then the amount f'_q of change of f for services of remaining requests is at most $(C_{\bar{q}} - \delta_{\bar{q}})\frac{\rho}{2}(d_{a\bar{q}} - z - d_{aq})$ by Lemma 4, where $\bar{q} \in \{b, c\} \setminus \{q\}$.

If $t = a$, then OPT moves the server from a to q, and then does not move it until the end of the phase, or OPT moves the server from $\bar{q} \in \{b, c\} - \{q\}$ to q in the phase. In the former case, $f^S \leq f_{aq} + f'_q \leq \frac{\rho}{2}(C_{\bar{q}}(d_{a\bar{q}} - z - d_{aq}) - 2\delta_q d_{aq} - 2\delta_{\bar{q}}(d_{a\bar{q}} - z) \leq \frac{\rho}{2}(C_{\bar{q}}(|d_{a\bar{q}} - z| - d_{aq})) \leq S_q$. In the latter case, $f^S \leq f_{\bar{q}q} \leq -\rho C_q(\frac{L}{2} - d_{aq}) \leq S_q$. If $t \in \{b, c\}$, then OPT moves the server from t to $q \neq t$, or locates the server on $q = t$ throughout the phase. In the former case, $f^S \leq f_{tq} \leq -\rho C_q(\frac{L}{2} - d_{aq}) \leq S_q$. In the latter case, $f^S \leq f'_t|_{\delta_{\bar{t}}=0} \leq -\rho C_{\bar{t}}(\frac{L}{2} - d_{a\bar{t}}) \leq S_q$. □

Lemma 7 *If $s \in B$, then $\{F_b, F_{\bar{s}}\} \leq 0$ if $\rho \geq 3.1467$.*

Proof We may without loss of generality that $s = a$. By Lemma 6 and $\Psi_{cq} = \Psi'_{at} = 0$, we will prove that $\min\{F_b, F_c\} = \min\{\max_{q \in V}\{M_{bq} + S_q + \Psi_{bg}\}, \max_{q \in V}\{M_{cq} + S_q\}\} \leq 0$ for $y \geq \frac{L}{2}$, $y \geq x$, and $0 \leq C_b, C_c \leq 2D$.

We first estimate F_b. It follows that $(\rho - 2)(\frac{L}{2} - y) + (-(\frac{\rho}{2} - 1)z - \frac{\rho}{2}(y - x)) = (\rho - 1)(x - y) \leq 0$. Thus, it follows that $F_b = -(\rho - 3)Dx + (\rho - 2)C_c(\frac{L}{2} - y) + \max\{C_c(-(\frac{\rho}{2} - 1)z - \frac{\rho}{2}(y - x)), C_b\frac{\rho}{2}(z - x - y) - (\rho - 3)Dz, -\rho C_c(\frac{L}{2} - y) - (\rho - 3)Dz, -\rho C_b(\frac{L}{2} - x) - (\rho - 3)Dz\} \leq -(\rho - 3)D(L - y) + \max\{0, (\rho - 2)C_c(\frac{L}{2} - y) + \max\{-\rho C_b(x + y - \frac{L}{2}), -\rho C_b(\frac{L}{2} - x)\}\} = \max\{0, D(L - (\rho - 1)y) + \max\{-2C_b(y - \frac{L}{2} + \frac{\rho}{2}x), -C_b((\rho - 1)L - \rho x - (\rho - 2)y)\}\}$. If $C_b \geq \frac{D}{2}$, then for $\rho \geq 3$, this is at most $\max\{0, D(L - 2y) + \max\{-D(y - \frac{L}{2} + \frac{3}{2}x), -D(L - \frac{3}{2}x - \frac{y}{2})\}\} = \max\{0, -3D(y - \frac{L}{2} + \frac{x}{2}), -\frac{3}{2}D(y - x)\} = 0$. Therefore, the lemma holds if $C_b \geq \frac{D}{2}$.

We next show that $F_c = M_{ca} + S_a = M_{ca}$ if $C_b \leq D$, i.e., $C_b \leq C_c$. Because $\Psi_{cb} = \Psi_{cc}$ and $M_{cc} - M_{cb} = -2\rho C_b z \leq 0$, $F_c = \max\{M_{ca}, M_{cb} + \Psi_{cb}\}$. It follows that $M_{cb} + \Psi_{cb} - M_{ca} = \rho C_b(\frac{L}{2} - y) + \max\{-\rho C_c(\frac{L}{2} - y), -\rho C_b(\frac{L}{2} - x), -\rho C_b(\frac{L}{2} - z)\} \leq \rho C_b \max\{0, x - y, z - y\} = 0$.

Therefore, it suffices to prove that $F_c = -(\rho - 3)Dy + C_b(\rho - 1)(y - x)$ or $F_b = \max\{F_b^1, F_b^2\}$ is at most 0 for $C_b \leq \frac{D}{2}$ and $\rho \geq 3.1467$, where $F_b^1 := D(L - (\rho - 1)y) - 2C_b(y - \frac{L}{2} + \frac{\rho}{2}x) = (D + C_b)L + (-(\rho - 1)D - 2C_b)y - \rho C_b x$ and $F_b^2 := D(L - (\rho - 1)y) - C_b((\rho - 1)L - \rho x - (\rho - 2)y) = (D - (\rho - 1)C_b)L + (-(\rho - 1)D + (\rho - 2)C_b)y + \rho C_b x$. We provide a proof of this in Appendix. □

We have proved that if $\rho \geq 3.1467$, then $g \leq 0$ for every case. Therefore, the proof of Theorem 2 is completed. Moreover, it follows from the proof of Lemma 7 that if $C_b \geq \frac{D}{2}$ of $C_b = 0$, then $g \leq 0$ for $\rho = 3$. Therefore, if $D \leq 2$, then $g \leq 0$ for $\rho = 3$, which proves Theorem 3.

References

[1] Y. Bartal, M. Charikar, and P. Indyk. On page migration and other relaxed task systems. *Theoretical Computer Science*, 268(1):43–66, 2001.

[2] Y. Bartal, A. Fiat, and Y. Rabani. Competitive algorithms for distributed data management. *J. Computer and System Sciences*, 51(3):341–358, 1995.

[3] D. L. Black and D. D. Sleator. Competitive algorithms for replication and migration problems. Technical Report CMU-CS-89-201, Department of Computer Science, Carnegie Mellon University, 1989.

[4] M. Chrobak, L. L. Larmore, N. Reingold, and J. Westbrook. Page migration algorithms using work functions. *J. Algorithms*, 24(1):124–157, 1997.

[5] C. Lund, N. Reingold, J. Westbrook, and D. Yan. Competitive on-line algorithms for distributed data management. *SIAM J. Comput.*, 28(3):1086–1111, 1999.

[6] A. Matsubayashi. Uniform page migration on general networks. *International Journal of Pure and Applied Mathematics*, 42(2):161–168, 2007.

[7] J. Westbrook. Randomized algorithms for multiprocessor page migration. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 7, pages 135–150, 1992.

Appendix

We show $\min\{F_c, \max\{F_b^1, F_b^2\}\} \le 0$ by proving both $\min\{F_c, F_b^1\} \le 0$ and $\min\{F_c, F_b^2\} \le 0$.

Lemma 8 $\min\{F_c, F_b^1\} \le 0$ *if* $y \ge \frac{L}{2} - \frac{x}{\rho}$ *and* $\rho \ge 3.1467$.

Proof It suffices to show the lemma for $F_c = \{-(\rho-3)+(\rho-1)k\}y - (\rho-1)kx$ and $F_b^1 = 2(1+k) + (-(\rho-1)-2k)y - \rho kx$ with $0 \le k \le \frac{1}{2}$ and $y \ge 1 - \frac{x}{\rho}$. We will prove $F_{\max}^1 := \max_{x,y,k} \min\{F_c, F_b^1\} \le 0$. It is easy to verify that if $y = y_1 := \frac{2(1+k)-kx}{2+(\rho+1)k}$, then $F_c = F_b^1$, and that $F_{\max}^1 = \max_{x,k} F_c|_{y=y_1}$. Because $\frac{dF_c|_{y=y_1}}{dx} = \frac{((\rho-3)-(\rho-1)k)k}{2+(\rho+1)k} - (\rho-1)k < 0$, $F_c|_{y=y_1}$ is maximized when $x = x_1$ with $\frac{2(1+k)-kx_1}{2+(\rho+1)k} = 1 - \frac{x_1}{\rho}$. Thus, we have $F_{\max}^1 = \max_k F_c|_{y=y_1, x=x_1}$, where $x_1 := \frac{(\rho-1)\rho k}{2+k}$ and $y_1 := 1 - \frac{(\rho-1)k}{2+k}$. Because $\frac{dF_c|_{y=y_1,x=x_1}}{dk} = -\frac{\rho-1}{(k+2)^2}\{(\rho^2-2)k^2 + 4(\rho^2-2)k - 2(\rho-1)\}$, $F_c|_{y=y_1,x=x_1}$ is maximized when $k = -2 + \sqrt{\frac{4\rho^2+2\rho-10}{\rho^2-2}}$. Therefore, $F_{\max}^1 = 4\rho^3 - 3\rho^2 - 11\rho + 12 - 2(\rho-1)\sqrt{2(2\rho^2+\rho-5)(\rho^2-2)}$, which is negative if $\rho \ge 3.1467$. □

Lemma 9 $\min\{F_c, F_b^2\} \le 0$ *if* $y \ge \frac{L}{2} - \frac{x}{\rho}$ *and* $\rho \ge 3.114$.

Proof The proof proceeds in just a similar way as Lemma 8. Assume $F_c = \{-(\rho-3)+(\rho-1)k\}y - (\rho-1)kx$ and $F_b^2 = 2(1-(\rho-1)k) + (-(\rho-1)+(\rho-2)k)y + \rho kx$ with $0 \le k \le \frac{1}{2}$ and $y \ge 1 - \frac{x}{\rho}$. We will prove $F_{\max}^2 := \max_{x,y,k} \min\{F_c, F_b^2\} \le 0$. If $y = y_2 := \frac{2(1-(\rho-1)k)+(2\rho-1)kx}{k+2}$, then $F_c = F_b^2$, and that $F_{\max}^2 = \max_{x,k} F_c|_{y=y_2}$. Because $\frac{dF_c|_{y=y_2}}{dx} = \frac{(-(\rho-3)+(\rho-1)k)(2\rho-1)k}{k+2} - (\rho-1)k = \frac{(2(\rho-1)^2k-(2\rho^2-5\rho+1))k}{k+2} \le \frac{((\rho-1)^2-(2\rho^2-5\rho+1))k}{k+2} = -\frac{\rho(\rho-3)k}{k+2} \le 0$, $F_c|_{y=y_2}$ is maximized when $x = x_2$ such that $\frac{2(1-(\rho-1)k)+(2\rho-1)kx_2}{k+2} = 1 - \frac{x_2}{\rho}$. Thus, we have $F_{\max}^2 = \max_k F_c|_{y=y_2,x=x_2}$, where $x_2 := \frac{(2\rho-1)\rho k}{(2\rho^2-\rho+1)k+2}$ and $y_2 := 1 - \frac{(2\rho-1)k}{(2\rho^2-\rho+1)k+2}$. Because $\frac{dF_c|_{y=y_2,x=x_2}}{dk} = \frac{-2((\rho-1)^2(2\rho^2-\rho+1)k^2+4(\rho-1)^2k-(2\rho^2-5\rho+1))}{((2\rho^2-\rho+1)k+2)^2}$, $F_c|_{y=y_2,x=x_2}$ is maximized when $k = \frac{-2+\sqrt{4+(2\rho^2-\rho+1)(2\rho^2-5\rho+1)/(\rho-1)^2}}{2\rho^2-\rho+1}$. Therefore, $F_{\max}^2 = \frac{-4\rho^5+20\rho^4-29\rho^3+34\rho^2-29\rho+12-4(\rho-1)\sqrt{(2\rho-1)(2\rho^3-5\rho^2+4\rho-5)}}{(2\rho^2-\rho+1)^2}$, which is negative if $\rho \ge 3.114$. □