社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－AL－109 (6)
2006／11／21

リングネットワークにおけるページ移動について

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，ネットワーク上でページと呼ばれるデータへのアクセス要求を発行するノード系列に対して，ページを動的に移動することにより要求に対するサービスコストと移動コストの総和を最小化する問題である．この問題に対しては，木，一様ネットワーク，およびそれらの Cartesian 積を除いて，4 未満の競合比を持つ決定的オンラインアルゴリズムは知られていない．本稿では，ページサイズが 1 に限定されている条件の下で，リングネットワークに対する決定的な $2+\sqrt{2}(\simeq 3.4142)$-競合オンラインアルゴリズムを示す．このアルゴリズムはリングの木とトーラスにも拡張できる．さらにページ移動問題の競合比の下界として，一般のネットワークに対して 3.1639，リングネットワークに対して 3.1213 を与える．

Page Migration on Ring Networks

Akira Matsubayashi

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

Abstract The page migration problem is to compute dynamic allocation of a page on a network for a given sequence of nodes issuing requests for the page. The goal is to minimize the total communication costs of services for requests and of migrations of the page. We did not know any deterministic online algorithm with competitive ratio less than 4 for networks other than trees, uniform networks, and Cartesian products of those networks so far. In this paper we give a $2+\sqrt{2}(\simeq 3.4142)$-competitive deterministic algorithm on rings (with edge weights) for the setting that the page size is 1. We can also derive algorithms for trees of rings and tori with the same competitive ratio and with the same setting. Moreover, we show an improved lower bound of 3.1639 for general networks and a lower bound of 3.1213 for rings. Our lower bound for rings is the first result which gives an explicit lower bound greater than 3 for rings, together with an explicit proof.

1 Introduction

The problem of computing efficient dynamic allocation of data objects stored in nodes of a network in response to requests issued by nodes for accessing the data objects commonly arises in network applications such as memory management in a shared memory multiprocessor system and Peer-to-Peer applications on the Internet. This problem is generally called the *data management problem* and has been extensively studied so far. Since it is not feasible to know the future accesses in advance, online algorithms for the problem are practical and interesting.

In this paper we focus on one of the traditional settings of the data management problem, called the *page migration problem*, in which only one copy of a data object, or a page, is allowed. The objective function to be minimized is the total sum of the service cost for each request, which is the distance between server and client nodes, and of the management cost for each data migration, which is the migration distance multiplied by the data size. There have been studied more general settings such as k-page migration [3], file allocation problem, e.g., [1][4][9], and data management on dynamic networks, e.g, [2][5][6].

For general networks, a 3-competitive randomized algorithm against an adaptive online adversary was given by Westbrook [10]. The algorithm is tight since Bartal, Fiat, and Rabani [4] showed that no randomized algorithm has competitive ratio less than 3 against an adaptive online adversary for one link networks. A randomized algorithm with competitive ratio against an oblivious adversary which tends to $\frac{3+\sqrt{5}}{2} \simeq 2.6180$ as the page size D grows large was given also in [10]. Optimal randomized algorithms for trees and product of trees, including grids and hypercubes, and for uniform networks with competitive ratio $2+\frac{1}{2D}$ against an oblivious adversary were given by Chrobak, Larmore, Reingold, and Westbrook [8] and by Lund, Reingold, Westbrook, and Yan [9], respectively. The tightness of the competitive ratio of $2+\frac{1}{2D}$ against an oblivious adversary was shown also in [8] by proving that no randomized algorithm has competitive ratio less than $2+\frac{1}{2D}$ against an oblivious adversary for one link networks.

As for deterministic page migration, Bartal, Charikar, and Indyk [3] gave a 4.086-competitive deterministic algorithm for general networks. It is mentioned in [10] that a naive deterministic algorithm which moves the page to the requesting node after each request is $2D+2$-competitive, which is better than the result of [3] when $D=1$. Black and Sleator [7] gave a 3-competitive deterministic algorithm for trees, uniform networks, and products of those networks, including grids and hypercubes. Besides, a 3-competitive deterministic algorithm on arbitrary 3-node networks for the setting of $D=1$ was given in [8]. The tightness of the competitive ratio of 3 for deterministic algorithms was first shown in [7] by proving that no deterministic algorithm has competitive ratio less than 3 for one link networks. For lower bounds of deterministic algorithms for networks other than one link networks, a lower bound of $\frac{85}{27} \simeq 3.1481$ for general networks was given in [8]. It was also mentioned in [8] that the lower bound for rings is greater than 3, but neither explicit value nor written proof was given.

In this paper we consider the deterministic data migration on rings. We give a $2+\sqrt{2}(\simeq 3.4142)$-competitive deterministic algorithm on rings for $D=1$. The setting of $D=1$ is often called *uniform model.* We can also derive algorithms for trees of rings and tori with the same competitive ratio for the uniform model. Moreover, we show an improved lower bound of 3.1639 for general networks and a lower bound of 3.1213 for rings. Our lower bound for rings is the first result which gives an explicit lower bound greater than 3 for rings, together with an explicit proof.

2 Preliminaries

Graphs $G=(V,E)$ considered here have edge weights $w: E \rightarrow \mathbb{R}^+$. The distance between two nodes u and v, denoted by $\mathrm{dist}(u,v)$, is the minimum sum of the weights of the edges of a path connecting u and v. We define that an n-node *ring* is a graph with the node set $\{0,\ldots,n-1\}$ and edge set $\{(v,(v+1) \bmod n) \mid 0 \le v < n\}$. We also model the ring as a closed curve with length $L=\sum_{e\in E} w(e)$, or a half-closed interval $[0,L)$ whose end-points 0 and L coincide. We define that for $0 \le p < q < L$, $[q,p]$ is $[q,L) \cup [0,p]$ and has the length of $L-(q-p)$. Each node $0 \le v < n$ corresponds to a point $\pi(v)=\sum_{j=0}^{v-1} w((j,j+1)) \in [0,L)$, and each edge $(v,(v+1) \bmod n)$ $(0 \le v < n)$ corresponds to $[\pi(v), \pi((v+1) \bmod n)]$. For $p \in [0,L)$, $\overline{p}$ is $p+\frac{L}{2}$ if $p<\frac{L}{2}$, $p-\frac{L}{2}$ otherwise. We denote the length of an interval I by $l(I)$.

The page migration problem is, given a graph G, a node s_0 of G which initially holds a page of size D, and a sequence $c_1, \ldots, c_k$ of nodes of G which issue requests for accessing the page, to compute a sequence $s_1, \ldots, s_k$ of nodes of G to hold the page so that the cost function $\sum_{i=1}^{k} \mathrm{dist}(s_{i-1},c_i)+D\mathrm{dist}(s_{i-1},s_i)$ is minimized. We call nodes $s_0,\ldots,s_k$ and $c_1,\ldots,c_k$ *servers* and *clients*, respectively. An online data migration algorithm determines s_i without knowing $c_{i+1}, \ldots, c_k$ for $1 \le i < k$. We denote by $\mathrm{cost}_A(\sigma)$ the cost of a data migration algorithm A for an instance $\sigma=(G,s_0,c_1,\ldots,c_k)$. An online data migration algorithm ALG is ρ-competitive if there exists a value α independent of k such that $\mathrm{cost}_{\mathrm{ALG}}(\sigma) \le \rho\mathrm{cost}_{\mathrm{OPT}}(\sigma)+\alpha$ for an optimal offline algorithm OPT and for any σ.

3 Algorithm for Rings

In this section we show the following theorem by constructing a desired algorithm:

Theorem 1 *There exists a $2+\sqrt{2}$-competitive deterministic data migration algorithm on rings for uniform model, i.e., $D=1$.*

3.1 Definition

We describe our algorithm UNIFORM_PAGE_MIGRATION_ON_RINGS (UPMR). For each edge of a given ring, UPMR has a counter whose value is 0, 1, or 2. All the counters are initially set to 0. Let $X_0=[\pi(s_0),\pi(s_0)]$. After determining s_i $(i \ge 1)$ UPMR preserves the condition that all the counters have 0 or 1 and that all the edges with counters of 1 induce a single interval X_i with an end-point $\pi(s_i)$ and with length at most $\frac{L}{2}$. Let $\rho=2+\sqrt{2}$. UPMR determines s_i $(i \ge 1)$ after serving the request from c_i as follows:

1. Assume without loss of generality that $\pi(s_{i-1})=0$ and $X_{i-1}=[0,x] \subseteq [0,\frac{L}{2}]$.

2. If $\pi(c_i) \le \frac{L}{2}$, then increment the counters of edges in $[0,\pi(c_i)]$ by 1.

3. If $\pi(c_i) > \frac{L}{2}$, then let y be the length of $[0,\pi(c_i)]$, i.e., $\pi(c_i)$.

   (a) If $x \leq \rho(y - \frac{L}{2})$, then decrement the counters of the edges of X_{i-1} by 1, i.e., set them to 0, and increment the counters of the edges in $[\pi(c_i), 0]$ by 1.

   (b) If $x > \rho(y - \frac{L}{2})$, then increment the counters of the edges in $[0, \pi(c_i)]$ by 1.

4. Move the page along all the edges with counters of 2, and set the counters of the edges to 0.

5. Let X_i be the interval induced by $\pi(s_i)$ and all the edges with counters of 1.

3.2 Correctness

UPMR is well-defined by the following lemma:

Lemma 1 UPMR *has the following properties for* $i \geq 1$*:*

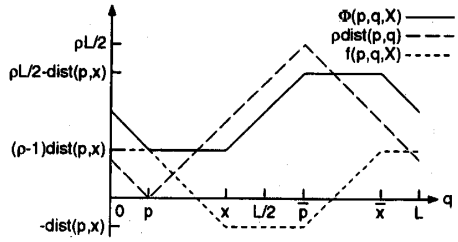
- *After Step 3,* $\pi(s_{i-1})$ *and all the edges with counters of* 2 *induce a single interval with an end-point* $\pi(s_{i-1})$.
- *After Step 4,* $\pi(s_i)$ *and all the edges with counters of* 1 *induce a single interval with an end-point* $\pi(s_i)$ *and with length at most* $\frac{L}{2}$.

Proof We prove the lemma by induction on i. As a base case, we can observe that $X_0 = [\pi(s_0), \pi(s_0)]$ satisfies the second property of the lemma. Assume that the lemma holds for $i-1$ ($i \geq 1$). Thus, before Step 1, $\pi(s_{i-1})$ and all the edges with counters of 1 induce a single interval X_{i-1} with an end-point $\pi(s_{i-1})$ and with length at most $\frac{L}{2}$. Assume without loss of generality that $\pi(s_{i-1}) = 0$ and $X_{i-1} \subseteq [0, \frac{L}{2}]$.

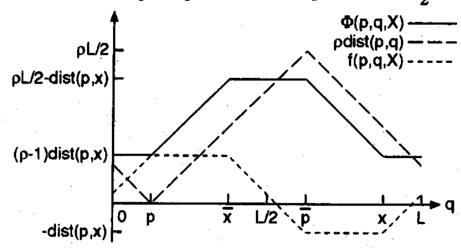
If s_i is determined via Step 2, then all the edges with counters of 2 induce $X_{i-1} \cap [0, \pi(c_i)]$, and all the edges with counters of 1 induce either $X_{i-1} - [0, \pi(c_i)]$ or $[0, \pi(c_i)] - X_{i-1}$. Then the page is migrated in Step 4 along $X_{i-1} \cap [0, \pi(c_i)]$ and all the counters of the edges in $X_{i-1} \cap [0, \pi(c_i)]$ are set to 0. Since both $X_{i-1} - [0, \pi(c_i)]$ and $[0, \pi(c_i)] - X_{i-1}$ have length at most $\frac{L}{2}$ by induction hypothesis and the assumption that $\pi(c_i) \leq \frac{L}{2}$, the lemma holds.

If s_i is determined via Step 3a, then all the edges with counters of 1 induce $[\pi(c_i), 0]$ and no edge has counter of 2. Since $[\pi(c_i), 0]$ has length $L - \pi(c_i) < \frac{L}{2}$, the lemma holds.

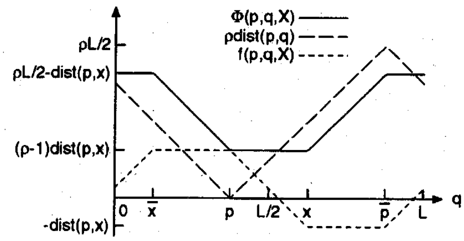
If s_i is determined via Step 3b, then all the edges with counters of 2 induce X_{i-1} and all the edges with counters of 1 induce $[0, \pi(c_i)] - X_{i-1}$. Then the page is migrated in Step 4 along X_{i-1} and all the counters of the edges in X_{i-1} are set to 0. Since $[0, \pi(c_i)] - X_{i-1}$ has length $y - x < \frac{x}{\rho} + \frac{L}{2} - x \leq \frac{L}{2}$, the lemma holds. □



(a) $X = [p, x]$ with $0 \leq p \leq x \leq \frac{L}{2}$



(b) $X = [x, p]$ with $0 \leq p \leq \frac{L}{2} < x < L$ and $x - p > \frac{L}{2}$



(c) $X = [p, x]$ with $0 \leq p \leq \frac{L}{2} < x < L$ and $x - p \leq \frac{L}{2}$

Figure 1: Plots of Φ in terms of q.

3.3 Competitiveness

Lemma 2 UPMR *is* $2 + \sqrt{2}$*-competitive.*

Proof We prove the lemma by observing the inequality

$$\mathrm{cost}_{\mathrm{UPMR}}(\sigma) + \Phi \leq \rho \mathrm{cost}_{\mathrm{OPT}}(\sigma), \qquad (1)$$

where $\rho = 2 + \sqrt{2}$ and Φ is a potential function. For $p, q \in [0, L)$, let $I^-_{p,q}$ be $[p, q] \cup [\overline{p}, \overline{q}]$ if $q \in [p, \overline{p}]$, $[q, p] \cup [\overline{q}, \overline{p}]$ otherwise, and let $I^+_{p,q} = [0, L) - I^-_{p,q}$. For $p, q \in [0, L)$ and an interval X on $[0, L)$, let $f(p, q, X) = -l(I^-_{p,q} \cap X) + (\rho - 1) l(I^+_{p,q} \cap X)$. We define $\Phi(p, q, X) = \rho \mathrm{dist}(p, q) + f(p, q, X)$, where p and q are the servers located by UPMR and OPT, respectively, and X is the interval induced by the edges with counters of 1. Figure 1 shows plots of Φ in terms of q. It should be noted that Φ consists of straight lines with slopes $-\rho$, 0, or ρ and that the values for $q = 0$ and $q \to L$ coincide.

Since Φ is initially 0, we can obtain (1) by observing that for each event of

- service and migration by UPMR and service by OPT, and
- migration of OPT

for each request,

$$\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi \leq \rho\Delta\mathrm{cost}_{\mathrm{OPT}}, \qquad (2)$$

where $\Delta\mathrm{cost}_A$ is the cost paid by an algorithm A for the event, and $\Delta\Phi$ is the increased amount of Φ by the event. For the event of migration of OPT of length λ, (2) is satisfied because $\Delta\mathrm{cost}_{\mathrm{UPMR}} = 0$, $\Delta\Phi \leq \rho\lambda$, and $\Delta\mathrm{cost}_{\mathrm{OPT}} = \lambda$.

In the rest of the proof, we consider the event consisting of service and migration by UPMR and service by OPT. We fix $1 \leq i < k$ and suppose that $y = \pi(c_i)$ and $p = \pi(s_{i-1})$. We may assume without loss of generality that $p = 0$ and $X_{i-1} = [p, x] \subseteq [0, \frac{L}{2}]$.

If UPMR determines s_i via Step 2 and $p \leq y \leq x$, then $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}} = 2\mathrm{dist}(p, y) + \Phi(y, q, [y, x]) - \Phi(p, q, [p, x]) - \rho\mathrm{dist}(y, q)$, which has slope in terms of q as described below:

q	p – y	y – x	x – $\overline{p}$	$\overline{p}$ – $\overline{y}$	$\overline{y}$ – $\overline{x}$	$\overline{x}$ – L
$\Phi(y, q, [y, x])$	$-\rho$	0	ρ	ρ	0	$-\rho$
$-\Phi(p, q, [p, x])$	0	0	$-\rho$	0	0	ρ
$-\rho\mathrm{dist}(y, q)$	ρ	$-\rho$	$-\rho$	$-\rho$	ρ	ρ
total	0	$-\rho$	$-\rho$	0	ρ	ρ

Thus, when $q = y$, $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}}$ has the maximum value $2\mathrm{dist}(p, y) + (\rho - 1)\mathrm{dist}(y, x) - (\rho - 1)\mathrm{dist}(p, x) = -(\rho - 3)y \leq 0$.

If UPMR determines s_i via Step 2 and $p \leq x < y$, then $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}} = \mathrm{dist}(p, y) + \mathrm{dist}(p, x) + \Phi(x, q, [x, y]) - \Phi(p, q, [p, x]) - \rho\mathrm{dist}(y, q)$, which has slope in terms of q as described below:

q	p – x	x – y	y – $\overline{p}$	$\overline{p}$ – $\overline{x}$	$\overline{x}$ – $\overline{y}$	$\overline{y}$ – L
$\Phi(x, q, [x, y])$	$-\rho$	0	ρ	ρ	0	$-\rho$
$-\Phi(p, q, [p, x])$	0	$-\rho$	$-\rho$	0	ρ	ρ
$-\rho\mathrm{dist}(y, q)$	ρ	ρ	$-\rho$	$-\rho$	$-\rho$	ρ
total	0	0	$-\rho$	0	0	ρ

Thus, when $q = x$, $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}}$ has the maximum value $\mathrm{dist}(p, y) + \mathrm{dist}(p, x) + (\rho - 1)\mathrm{dist}(x, y) - (\rho - 1)\mathrm{dist}(p, x) - \rho\mathrm{dist}(y, x) = -(\rho - 3)x \leq 0$.

If UPMR determines s_i via Step 3a, then $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}} = \mathrm{dist}(p, y) + \Phi(p, q, [y, p]) - \Phi(p, q, [p, x]) - \rho\mathrm{dist}(y, q)$. If $y - x \geq \frac{L}{2}$, then it has slope in terms of q as described below:

q	p – x	x – $\overline{y}$	$\overline{y}$ – $\overline{p}$	$\overline{p}$ – $\overline{x}$	$\overline{x}$ – y	y – L
$\Phi(p, q, [y, p])$	ρ	ρ	0	$-\rho$	$-\rho$	0
$-\Phi(p, q, [p, x])$	0	$-\rho$	$-\rho$	0	ρ	ρ
$-\rho\mathrm{dist}(y, q)$	$-\rho$	$-\rho$	ρ	ρ	ρ	$-\rho$
total	0	$-\rho$	0	0	ρ	0

Thus, when $q = p$, $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}}$ has the maximum value $\mathrm{dist}(p, y) + (\rho - 1)\mathrm{dist}(p, y) - (\rho - 1)\mathrm{dist}(p, x) - \rho\mathrm{dist}(y, p) = -(\rho - 1)x \leq 0$. If $y - x < \frac{L}{2}$, then $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}}$ has slope in terms of q as described below:

q	p – $\overline{y}$	$\overline{y}$ – x	x – $\overline{p}$	$\overline{p}$ – y	y – $\overline{x}$	$\overline{x}$ – L
$\Phi(p, q, [y, p])$	ρ	0	0	$-\rho$	0	0
$-\Phi(p, q, [p, x])$	0	0	$-\rho$	0	0	ρ
$-\rho\mathrm{dist}(y, q)$	$-\rho$	ρ	ρ	ρ	$-\rho$	$-\rho$
total	0	ρ	0	0	$-\rho$	0

Thus, when $q = x$, $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}}$ has the maximum value $\mathrm{dist}(p, y) + \rho\frac{L}{2} - \mathrm{dist}(p, y) - (\rho - 1)\mathrm{dist}(p, x) - \rho\mathrm{dist}(y, x) = \rho\frac{L}{2} - (\rho - 1)x - \rho(y - x) = x - \rho(y - \frac{L}{2}) \leq 0$.

If UPMR determines s_i via Step 3b, then $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}} = \mathrm{dist}(p, y) + \mathrm{dist}(p, x) + \Phi(x, q, [x, y]) - \Phi(p, q, [p, x]) - \rho\mathrm{dist}(y, q)$, which has slope in terms of q as described below:

q	p – $\overline{y}$	$\overline{y}$ – x	x – $\overline{p}$	$\overline{p}$ – y	y – $\overline{x}$	$\overline{x}$ – L
$\Phi(x, q, [x, y])$	0	$-\rho$	0	0	ρ	0
$-\Phi(p, q, [p, x])$	0	0	$-\rho$	0	0	ρ
$-\rho\mathrm{dist}(y, q)$	$-\rho$	ρ	ρ	ρ	$-\rho$	$-\rho$
total	$-\rho$	0	0	ρ	0	0

Thus, when $q = p$, $\Delta\mathrm{cost}_{\mathrm{UPMR}} + \Delta\Phi - \rho\Delta\mathrm{cost}_{\mathrm{OPT}}$ has the maximum value $\mathrm{dist}(p, y) + \mathrm{dist}(p, x) + \rho\frac{L}{2} - \mathrm{dist}(x, y) - (\rho - 1)\mathrm{dist}(p, x) - \rho\mathrm{dist}(y, p) = -(\rho - 1)(L - y) - (\rho - 2)x + \rho\frac{L}{2} - (y - x) = (\rho - 2)(y - \frac{L}{2}) - (\rho - 3)x = (\rho - 3)(\frac{\sqrt{2}}{\sqrt{2}-1}(y - \frac{L}{2}) - x) = (\rho - 3)(\rho(y - \frac{L}{2}) - x) < 0$. □

Therefore, the proof of Theorem 1 is completed.

4 Algorithms for Trees of Rings and Tori

Any ρ-competitive data migration algorithm on a class $\mathcal{C}$ of graphs can be extended to a ρ-competitive algorithm for Cartesian products of graphs in $\mathcal{C}$ [8]. Thus, we can immediately obtain the following theorem from Theorem 1:

Theorem 2 *There exists a $2 + \sqrt{2}$-competitive deterministic data migration algorithm on tori for uniform model, i.e., $D = 1$.*

A *tree of rings* is a graph obtained from an underlying tree T by replacing each node v of T with a cycle C_v so that nodes u and v of T are adjacent if and only if C_u and C_v share exactly one node. We can easily extend UPMR to an algorithm for trees of rings.

Theorem 3 *There exists a $2+\sqrt{2}$-competitive deterministic data migration algorithm on trees of rings for uniform model, i.e., $D=1$.*

Proof Our algorithm UPMTR on trees of rings is defined as follows: Let $G=(V_G,E_G)$ be a tree of rings with an underlying tree $T=(V_T,E_T)$. For $p \in V_G$ and $v \in V_T$, let $p^v \in V_G$ be the node of C_v nearest to p. For a given instance $\sigma = (G, s_0, c_1, \ldots, c_k)$, UPMTR performs UPMR on each cycle C_v for the instance $\sigma_v = (C_v, s_0^v, c_1^v, \ldots, c_k^v)$. The correctness of UPMTR can be shown by observing the following properties for $i \geq 1$:

- After Step 3 of UPMR is performed on every cycle, s_{i-1} and all the edges with counters of 2 induce a single path with an end-point s_{i-1}.
- After Step 4 of UPMR is performed on every cycle, s_i and all the edges with counters of 1 induce a single path with an end-point s_i.

These properties can be observed by induction on i. As a base case, the path of length 0 with the end-node s_0 satisfies the second property. The inductive step can be shown by Lemma 1, by the fact that UPMR increases the counters of the edges of a path between s_{i-1}^v and c_i^v for $v \in V_T$, and by the fact that any two cycles in G are connected by a unique sequence of cycles.

By definition, it clearly follows that $\mathrm{cost}_{\mathrm{UPMTR}}(\sigma) = \sum_{v\in V_T} \mathrm{cost}_{\mathrm{UPMR}}(\sigma_v)$. Moreover, since any two cycles of G share at most one node, the services and migrations performed by an algorithm OPT on G can be divided into algorithms A_v on each cycle C_v with the instance σ_v in such a way that A_v manages $s_1^v, \ldots, s_k^v$ for servers $s_1, \ldots, s_k$ managed by OPT, and it follows that $\mathrm{cost}_{\mathrm{OPT}}(\sigma) = \sum_{v\in V_T} \mathrm{cost}_{A_v}(\sigma_v)$. Therefore, we have by Lemma 2 that $\mathrm{cost}_{\mathrm{UPMTR}}(\sigma) = \sum_{v\in V_T} \mathrm{cost}_{\mathrm{UPMR}}(\sigma_v) \leq \sum_{v\in V_T}\{(2+\sqrt{2})\mathrm{cost}_{A_v}(\sigma_v) + \alpha\} = (2+\sqrt{2})\mathrm{cost}_{\mathrm{OPT}}(\sigma) + \alpha|V_T|$. □

5 Lower Bound for General Networks

In this section we show the following theorem:

Theorem 4 *There exists no deterministic ρ-competitive data migration algorithm for general networks if $\rho < 3.1639$.*

A lower bound of the competitive ratio of $\frac{85}{27} \simeq 3.1481$ for general networks was given in [8] by showing the following lemmas:

Lemma A *For any deterministic online data migration algorithm* ALG, *there exists an instance σ such that $\mathrm{cost}_{\mathrm{ALG}}(\sigma) \geq \frac{85}{27}\mathrm{cost}_{\mathrm{OPT}}(\sigma) > 0$ and that both* ALG *and* OPT *put the page on the last client in σ.*

Lemma B *For any deterministic online data migration algorithm* ALG, *if there exists an instance σ such that $\mathrm{cost}_{\mathrm{ALG}}(\sigma) \geq \rho\mathrm{cost}_{\mathrm{OPT}}(\sigma) > 0$ and that both* ALG *and* OPT *put the page on the last client in σ, then there exists an instance σ' such that $\mathrm{cost}_{\mathrm{ALG}}(\sigma') \geq \rho\mathrm{cost}_{\mathrm{OPT}}(\sigma') + \alpha$ for any α independent of the number of the requests in σ'.*

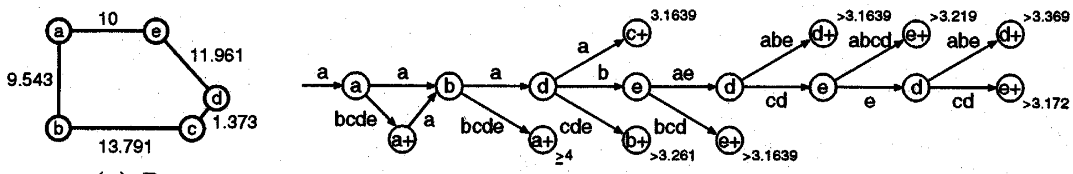
Lemma A was proved in [8] by giving a 4-node ring and an adversary's strategy which satisfy the conditions of the lemma. We modify the ring and the strategy of [8] and obtain the following lemma:

Lemma 3 *For any deterministic online data migration algorithm* ALG, *there exists an instance σ such that $\mathrm{cost}_{\mathrm{ALG}}(\sigma) \geq 3.1639\mathrm{cost}_{\mathrm{OPT}}(\sigma) > 0$ and that both* ALG *and* OPT *put the page on the last client in σ.*

Proof We define a 5-node ring R_1 and a strategy for an adversary ADV to generate clients on R_1 as shown in Fig. 2. We set $D=1$ and the initial server to the node a. The strategy is illustrated by a tree-like DAG, in which each edge represents a server determined by an online algorithm ALG, and each node represents a client chosen by ADV. An edge with more than one server denotes that ALG put the page on one of the servers. A client followed by a plus sign denotes that ADV repeats the requests from the client until ALG moves the page to the client. In response to the choices of the servers of ALG, an online game between ALG and ADV proceeds along a path from the unique source node to a sink node on the DAG. Table 1 shows the servers of OPT and the ratio of the costs of ALG and OPT for each path except the paths preceded by the nodes $aa+$, which clearly increase only the cost of ALG. By Table 1, the cost ratio is at least 3.1639 whichever path ALG chooses. □

By Lemmas B and 3, we have Theorem 4.

The precise edge weights of R_1 are obtained from the conditions that the four cost ratios for ALG's servers (ADV's clients, respectively) *aaaac* (*abdc*+), *aaabbe* (*abdee*+), *aaabde* (*abdee*+), *aaabeed* (*abdedd*+) are the same and that $\mathrm{dist}(a,c)$ is exactly half of the total weights, maximizing the cost ratio for ALG's servers *aaaac* (ADV's clients *abdc*+).



(a) R_1 (b) ADV's strategy

Figure 2: Ring R_1 and ADV's strategy on R_1

Table 1: An optimal algorithm OPT and the ratios of the costs of OPT and ALG

servers of ALG	clients of ADV	servers of OPT	$\text{cost}_{\text{ALG}}/\text{cost}_{\text{OPT}} \geq$
aaaac	*abdc*+	*abccc*	$78.172/24.707 > 3.1639$
aaab[*ae*][*abe*]*d*	*abdedd*+	*abddddd*	$116.016/36.668 > 3.1639$
aaab[*ae*][*cd*][*abcd*]*e*	*abdedee*+	*aaeeeeee*	$139.938/43.465 > 3.219$
aaab[*ae*][*cd*]*e*[*abe*]*d*	*abdededd*+	*abdddddd*	$163.86/48.629 > 3.369$
aaab[*ae*][*cd*]*e*[*cd*]*e*	*abdedede*+	*aaeeeeeee*	$175.821/55.426 > 3.172$
aaab[*bcd*]*e*	*abdee*+	*aaeeee*	$99.676/31.504 > 3.1639$
aaa[*cde*]*b*	*abdb*+	*abbbb*	$80.59/24.707 > 3.261$
aa[*bcde*]*a*	*aba*+	*aaaa*	$38.172/9.543 \geq 4$

6 Lower Bound for Rings

The proof of Theorem 4 requires sufficiently large tree-of-rings-like networks due to Lemma B. In this section we give a lower bound for ring networks.

Theorem 5 *There exists no deterministic ρ-competitive data migration algorithm for rings if $\rho <$ 3.1213.*

Proof We show that for any deterministic online data migration algorithm ALG, there exists an instance σ with a ring such that $\text{cost}_{\text{ALG}}(\sigma) \geq 3.1213\text{cost}_{\text{OPT}}(\sigma) + \alpha$ for any α independent of the number of the clients of σ. To show this, we define a 5-node ring R_2 as shown in Fig. 3 and a strategy for an adversary ADV to generate arbitrarily long sequence of clients on R_2 such that $\frac{\text{cost}_{\text{ALG}}(\sigma)}{\text{cost}_{\text{OPT}}(\sigma)} \geq$ 3.1213 with an arbitrarily large $\text{cost}_{\text{OPT}}(\sigma)$. The strategy consists of partial strategies S_a, S_b, S_c, S_d, and S_e (Fig. 3). By an argument similar to the proof of Lemma 3, together with the cost ratios shown in Table 2 for the partial strategies, for each node v of R_2 and any online algorithm A, there exists a sequence χ_A^v of clients such that $\text{cost}_A((R_2, v, \chi_A^v)) \geq 3\text{cost}_{\text{OPT}}((R_2, v, \chi_A^v)) > 0$ and that both A and OPT put the server on the last client of χ_A^v. As done in the proof of Lemma 3, we omit to consider the sequences beginning with $vv+$ in S_v.

We set $D = 1$ and the initial server of σ to the node a. ADV generates clients in phases: The ith phase ($i \geq 1$) is defined as $\chi_{\text{ALG}_i}^{v_i}$, where ALG_i is the algorithm performed by ALG in the ith phase, and v_i is the node on which ALG and OPT have the page just before the phase begins. Let $\sigma_i = (R_2, v_i, \chi_{\text{ALG}_i}^{v_i})$. The theorem is proved by observing that $\frac{\sum_i \text{cost}_{\text{ALG}_i}(\sigma_i)}{\sum_i \text{cost}_{\text{OPT}}(\sigma_i)} \geq 3.1213$. By Table 2, all the sequences of clients in the partial strategies yield the cost ratios greater than 3.1213 except for *baa*+, *cbaa*+, *daa*+, and *eaa*+. Assume that there exists $j > 1$ with $v_j \in \{b, c, d, e\}$ and $\chi_{\text{ALG}_j}^{v_j} \in \{baa+, cbaa+, daa+, eaa+\}$. If there exists no such j, then the theorem is immediate. Table 3 shows that $\frac{\text{cost}_{\text{ALG}_{j-1}}(\sigma_{j-1}) + \text{cost}_{\text{ALG}_j}(\sigma_j)}{\text{cost}_{\text{OPT}}(\sigma_{j-1}) + \text{cost}_{\text{OPT}}(\sigma_j)} \geq 3.1213$ for every possible combination of $\chi_{\text{ALG}_j}^{v_j}$ and $\chi_{\text{ALG}_{j-1}}^{v_{j-1}}$ ending with v_j+. Therefore, we have the theorem. □

The precise edge weights of R_2 are obtained from the conditions that the four combined cost ratios for ALG's servers (ADV's clients, respectively) *aaaac-cccba* (*abdc*+-*cbaa*+), *aaabbe-eeea* (*abdee*+-*eaa*+), *aaabde-eeea* (*abdee*+-*eaa*+), *aaaeb-bbba* (*abdb*+-*baa*+) are the same and that $\text{dist}(a, c)$ is exactly half of the total weights.

7 Concluding Remarks

Our analysis of competitiveness of UPMR is tight: For repeated pairs of alternate requests issued from the two nodes at a distance of $\frac{2+\sqrt{2}-\epsilon}{3+\sqrt{2}} \frac{L}{2}$ ($\epsilon > 0$) from the initial server, UPMR does not move the page and pays the cost of $\frac{2+\sqrt{2}-\epsilon}{3+\sqrt{2}} L$ for each pair of the requests. On the other hand, an optimal

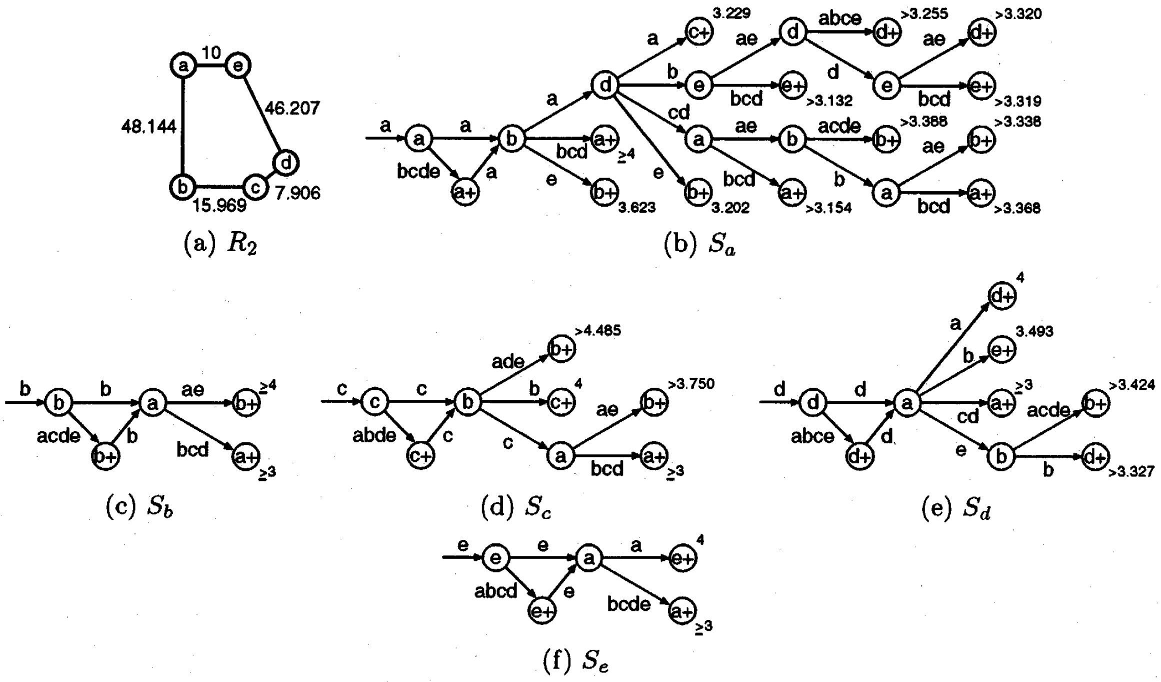


Figure 3: Ring R_2 and ADV's partial strategies on R_2

Table 2: An optimal algorithm OPT and the ratios of the costs of OPT and ALG

servers of ALG	clients of ADV	servers of OPT	$\mathrm{cost}_{\mathrm{ALG}}/\mathrm{cost}_{\mathrm{OPT}} \geq$
aaaac	*abdc+*	*abccc*	232.577/72.019 > 3.229
aaab[ae][abce]d	*abdedd+*	*abddddd*	384.915/118.226 > 3.255
aaab[ae]d[ae]d	*abdeded+*	*abdddddd*	546.025/164.433 > 3.320
aaab[ae]d[bcd]e	*abdedee+*	*aaeeeeee*	499.818/150.558 > 3.319
aaab[bcd]e	*abdee+*	*aaeeee*	326.927/104.351 > 3.132
aaa[cd][ae][acde]b	*abdabb+*	*abbbbbb*	407.167/120.163 > 3.388
aaa[cd][ae]b[ae]b	*abdabab+*	*abbbbbbb*	561.836/168.307 > 3.338
aaa[cd][ae]b[bcd]a	*abdabaa+*	*aaaaaaaa*	513.692/152.495 > 3.368
aaa[cd][bcd]a	*abdaa+*	*aaaaaa*	329.179/104.351 > 3.154
aaaeb	*abdb+*	*abbbb*	230.639/72.019 > 3.202
aa[bcd]a	*aba+*	*aaaa*	192.576/48.144 = 4
aaeb	*abb+*	*abbb*	174.432/48.144 > 3.623
bb[ae]b	*bab+*	*bbbb*	192.576/48.144 = 4
bb[bcd]a	*baa+*	*baaa*	144.432/48.144 = 3
cc[ade]b	*cbb+*	*cbbb*	71.625/15.969 > 4.485
ccbc	*cbc+*	*cccc*	63.876/15.969 = 4
ccc[ae]b	*cbab+*	*cbbbb*	240.483/64.113 > 3.750
ccc[bcd]a	*cbaa+*	*cbaaa*	192.339/64.113 = 3
ddad	*dad+*	*dddd*	224.828/56.207 = 4
ddbe	*dae+*	*deee*	196.37/56.207 > 3.493
dd[cd]a	*daa+*	*daaa*	168.621/56.207 = 3
dde[acde]b	*dabb+*	*dbbbb*	246.609/72.019 > 3.424
ddebd	*dabd+*	*ddddd*	266.452/80.082 > 3.327
eeae	*eae+*	*eeee*	40/10 = 4
ee[bcde]a	*eaa+*	*eaaa*	30/10 = 3

Table 3: Combined cost ratios for the $(j-1)$st and jth phases

$\chi^{v_j}_{\mathrm{ALG}_j}$	$\chi^{v_{j-1}}_{\mathrm{ALG}_{j-1}}$ ending with v_j+	$\frac{\mathrm{cost}_{\mathrm{ALG}_{j-1}}(\sigma_{j-1})+\mathrm{cost}_{\mathrm{ALG}_j}(\sigma_j)}{\mathrm{cost}_{\mathrm{OPT}}(\sigma_{j-1})+\mathrm{cost}_{\mathrm{OPT}}(\sigma_j)} \geq$
baa+	*abdabb*+	$(407.167+144.432)/(120.163+48.144) > 3.277$
	abdabab+	$(561.836+144.432)/(168.307+48.144) > 3.262$
	abdb+ or *dabb*+	$(230.639+144.432)/(72.019+48.144) > 3.1213$
	abb+ or *bab*+	$(174.432+144.432)/(48.144+48.144) > 3.311$
	cbb+	$(71.625+144.432)/(15.969+48.144) > 3.369$
	cbab+	$(240.483+144.432)/(64.113+48.144) > 3.428$
cbaa+	*abdc*+	$(232.577+192.339)/(72.019+64.113) > 3.1213$
	cbc+	$(63.876+192.339)/(15.969+64.113) > 3.199$
daa+	*abdedd*+	$(384.915+168.621)/(118.226+56.207) > 3.173$
	abdeded+	$(546.025+168.621)/(164.433+56.207) > 3.238$
	dad+	$(224.828+168.621)/(56.207+56.207) = 3.5$
	dabd+	$(266.452+168.621)/(80.082+56.207) > 3.192$
eaa+	*abdedee*+	$(499.818+30)/(150.558+10) > 3.299$
	abdee+	$(326.927+30)/(104.351+10) > 3.1213$
	dae+	$(196.37+30)/(56.207+10) > 3.419$
	eae+	$(40+30)/(10+10) = 3.5$

algorithm moves the page exactly once to one of the nodes for the first request and pays the cost of $L - \frac{2+\sqrt{2}-\epsilon}{3+\sqrt{2}} L = \frac{1+\epsilon}{3+\sqrt{2}} L$ for each succeeding pair of the requests. As the number of requests increases, the cost ratio tends to $\frac{2+\sqrt{2}-\epsilon}{1+\epsilon} \simeq 2+\sqrt{2}$ for small ϵ.

An online algorithm ALG is said to be *strictly ρ-competitive* if $\mathrm{cost}_{\mathrm{ALG}}(\sigma) \leq \rho \mathrm{cost}_{\mathrm{OPT}}(\sigma)$ for any σ. UPMR (and UPMTR) is strictly $2+\sqrt{2}$-competitive since Φ defined here is a non-negative function and is initially 0, and since (2) holds for each request. Lemma 3 implies that our lower bound of 3.1639 for general networks is also a lower bound of strict competitive ratio for deterministic data migration on rings.

We do not know any lower bound greater than 3 for deterministic page migration on unweighted graphs, i.e., graphs with edges of equal weights.

References

[1] B. Awerbuch, Y. Bartal, and A. Fiat. Competitive distributed file allocation. In *Proc. 25th Annual ACM Symposium on Theory of Computing*, pages 164–173, 1993.

[2] B. Awerbuch, Y. Bartal, and A. Fiat. Distributed paging for general networks. *J. Algorithms*, 28(1):67–104, 1998.

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

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

[5] M. Bienkowski, M. Dynia, and M. Korzeniowski. Improved algorithms for dynamic page migration. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science*, volume 3404 of *Lecture Notes in Computer Science*, pages 365–376. Springer-Verlag, 2005.

[6] M. Bienkowski and F. Meyer auf der Heide. Page migration in dynamic networks. In *Proc. 30th International Symposium on Mathematical Foundations of Computer Science*, pages 1–14, 2005.

[7] 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.

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

[9] 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.

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