

Uniform page migration problem in Euclidean space

メタデータ	言語: eng 出版者: 公開日: 2018-01-05 キーワード (Ja): キーワード (En): 作成者: メールアドレス: 所属:
URL	https://doi.org/10.24517/00049671

This work is licensed under a Creative Commons Attribution 3.0 International License.



Article

Uniform Page Migration Problem in Euclidean Space

Amanj Khorramian * and Akira Matsubayashi

Division of Electrical Engineering and Computer Science, Kanazawa University, Kanazawa 920-1192, Japan; mbayashi@t.kanazawa-u.ac.jp

* Correspondence: khorradian@gmail.com; Tel.: +98-918-974-3123

Academic Editor: Dimitris Fotakis

Received: 16 June 2016; Accepted: 18 August 2016; Published: 23 August 2016

Abstract: The page migration problem in Euclidean space is revisited. In this problem, online requests occur at any location to access a single page located at a server. Every request must be served, and the server has the choice to migrate from its current location to a new location in space. Each service costs the Euclidean distance between the server and request. A migration costs the distance between the former and the new server location, multiplied by the page size. We study the problem in the uniform model, in which the page has size $D = 1$. All request locations are not known in advance; however, they are sequentially presented in an online fashion. We design a 2.75-competitive online algorithm that improves the current best upper bound for the problem with the unit page size. We also provide a lower bound of 2.732 for our algorithm. It was already known that 2.5 is a lower bound for this problem.

Keywords: page migration problem; competitive analysis; server problems; online algorithms; adversary model; euclidean space; uniform model

1. Introduction

The page migration problem is a classical formulation of efficient memory management in a shared memory multiprocessor system comprising a network of processors having their own local memories. In this problem, we are given a sequence of requests issued by processors to access a single shared data object, called a page. Each request must be served via communication between the processor issuing the request and the processor holding the page at the time of the request. The service for the request costs the distance of the communication. After each service, the page is allowed to migrate from the current processor to a new processor before the next request is issued. The migration of the page costs the distance of the migration multiplied by the page size. The objective of the page migration problem is to minimize the total costs of services and migrations (see Section 2 for a formal definition of this problem). This problem can also be viewed as a formulation of general situations for managing shared information in a distributed network, such as the efficient allocation of each shared data object to be accessed by a distributed program running on the network [1].

Black and Sleator [2] first studied online algorithms for the page migration problem via competitive analysis. Some basic notions of competitive analysis are given in Section 2. Chrobak, Larmore, Reingold, and Westbrook [3] studied this problem on continuous metric spaces as well as networks. They proposed a $(2 + 1/2D)$ -competitive randomized algorithm on trees against oblivious adversaries, together with the same lower bound as this competitiveness even at two points, where D is a positive integer representing the page size. They also proved that any randomized page migration algorithm on \mathbb{R}^n can be derandomized while preserving its competitiveness. As any page migration algorithm on a space can be generalized on Cartesian products of the space [2], a $(2 + 1/2D)$ -competitive deterministic algorithm on \mathbb{R}^n under L^1 norm can be obtained through generalizing the algorithm on trees in \mathbb{R}^n [3]. For general metric spaces, Westbrook [4] proposed

a $c(D)$ -competitive randomized algorithm against oblivious adversaries, where $c(D)$ is a function defined in [4] that tends toward 2.618 as D grows large. Derandomizing this algorithm on \mathbb{R}^n , we have a $c(D)$ -competitive deterministic algorithm on \mathbb{R}^n under any norm [3].

Moreover, $2 + 1/2D$ is a lower bound for any randomized or deterministic algorithm in the interval $[0, 1]$ [3]. This lower bound is also admitted on \mathbb{R}^n under L^p norm. This is because, for the interval $I = [0, 1]$ on a dimension k in \mathbb{R}^n , any online algorithm A locating its server in $\mathbb{R}^n \setminus I$ for requests only in I has a cost of at least that of a certain algorithm locating its server only in I , i.e., projection on the k th coordinate of A 's server location if the projection is in I , and the closer endpoint of I otherwise. It is a longstanding open question whether we can close the gap of $c(D)$ and $2 + 1/2D$ under L^p norm even with any $p \geq 2$.

In this paper, we partially answer this question by proposing a 2.75-competitive deterministic algorithm on \mathbb{R}^n under L^2 norm for the case $D = 1$. We note that the competitiveness improves $c(1)$, which is equal to 2.8 as mentioned in [4]. We also provide a lower bound of 2.732 for our algorithm. Our algorithm is extremely simple: it maintains the page at the center of two special points, both initially located at the initial location of the server. Upon each request, the algorithm chooses one of the points farther from the requested location and moves the point to the requested location; the other point does not move. This algorithm can be viewed as a natural extension of the $(2 + 1/2D)$ -competitive algorithm under L^1 norm of [3]. We note that if there exists a $(2 + 1/2D)$ -competitive algorithm under L^2 norm, then it would have to behave on any line in the same way as the $(2 + 1/2D)$ -competitive algorithm of [3]. The setting of the unit page size, i.e., $D = 1$ is called the uniform model in the context of data management problems including the page migration problem [1,5–7].

Other previous results for the page migration problem are as follows. Bartal, Charikar, and Indyk [8] proposed a 4.086-competitive deterministic algorithm for general networks. For trees and uniform networks, Black and Sleator [2] presented 3-competitive deterministic algorithms, together with the lower bound of 3 even at two points. The first deterministic lower bound larger than 3 for general networks was 3.148 [3], which was improved to 3.168 by Matsubayashi [5]. Although these lower bounds, larger than 3, were proved only for $D = 1$, the first lower bound of $3 + \Omega(1)$ was recently proved [9], where Ω notation is with respect to D . For three points, 3-competitive deterministic algorithms with $D \in \{1, 2\}$ [3,10] and a $(3 + 1/D)$ -competitive deterministic algorithm with $D \geq 3$ [10] were proposed. The latter algorithm asymptotically matches the lower bound of $3 + \Omega(1/D)$ on three points for every $D \geq 3$ [10]. As for randomized algorithms against adaptive online adversaries, a 3-competitive algorithm for general metric spaces was proposed in [4] and the upper bound of 3 is also a lower bound on two points [11]. As for a randomized algorithm against oblivious adversaries, Lund, Reingold, Westbrook, and Yan [12] proved that the optimal competitiveness of $2 + 1/2D$ can be achieved even on uniform networks.

The rest of this paper is organized as follows. Some preliminaries are given in Section 2. We define our algorithm on \mathbb{R}^n under L^2 norm and prove its 2.75 competitiveness in Section 3. In Section 4, we prove a lower bound of 2.732 for our algorithm. We conclude the paper with remarks and future works in Section 5.

2. Preliminaries

Let M be a metric space with a distance function δ . For a given initial page location $s_0 \in M$, a sequence of requests $r_1, \dots, r_n \in M$, and a page size $D \in \mathbb{Z}^+$, the page migration problem is to compute page locations $s_1, \dots, s_n \in M$ such that the cost function $\sum_{i=1}^n (\delta(s_{i-1}, r_i) + D\delta(s_{i-1}, s_i))$ is minimized. We call s_i a server.

An online algorithm must compute s_i without any information regarding r_{i+1}, \dots, r_n , whereas an offline algorithm may compute s_i using the entire sequence of requests r_1, \dots, r_n . An adversary against an online algorithm A generates a sequence of requests given to A and computes an output sequence for the generated requests. If A is deterministic, then the adversary generates requests using the definition of A , or equivalently, the information of the actual behavior of A , and computes its own

output according to an optimal offline algorithm OPT. The deterministic algorithm A is c -competitive if $cost_A(s_0, \sigma) \leq c \cdot cost_{OPT}(s_0, \sigma) + \alpha$ for the initial server s_0 and any sequence σ of requests, where $cost_A$ and $cost_{OPT}$ are costs of A and OPT, respectively, and α is a constant. For randomized online algorithms, there are two types of adversaries. An adversary is said to be oblivious if it generates requests in advance only using the definition of A , i.e., without any information about the random behavior of A and computes its own output according to OPT. In contrast, an adaptive online adversary generates requests using information of the random behavior of A and computes its own output in an online fashion. The competitiveness of a randomized online algorithm against oblivious or adaptive online adversaries is defined in a similar way to that of a deterministic online algorithm, except that expected values are used for randomized costs.

One commonly used technique for proving the competitiveness of an online algorithm A is to utilize a potential function Φ , which typically maps the situation at a point of time, such as the page locations of A and OPT, to a real value. More specifically, we suitably divide the sequence of the online processes of A and OPT into certain events. Our goal is to define the value of Φ in such a way that the initial value of Φ is at most some constant β , Φ is always at least some constant $-\gamma$ and that $\Delta cost_A + \Delta \Phi \leq c \cdot \Delta cost_{OPT}$ for any event, where Δ denotes the change of values by the event. Summing the inequality over all events for an initial server s_0 and a request sequence σ , we have $cost_A(s_0, \sigma) \leq c \cdot cost_{OPT}(s_0, \sigma) + \beta + \gamma$, which means that A is c -competitive.

3. Algorithm and Its Upper Bound Analysis

We focus on the page size $D = 1$ and the metric space \mathbb{R}^n under L^2 norm δ for $n \geq 1$. We state our deterministic algorithm, called PQ, in the following paragraph. Intuitively, the algorithm determines the location of the page using two locations from previous requests. We prove the competitiveness of 2.75, claimed in Theorem 1, using a potential function Φ . As mentioned in Section 2, defining suitable events and showing that $\Delta cost_{PQ} + \Delta \Phi \leq 2.75 \Delta cost_{OPT}$ for any defined event is sufficient. For the proof of the theorem, we need to show an inequality different from a triangle inequality. We separately provide a technical part for that inequality as Lemma 1. The lemma is about points computed by the algorithm.

Algorithm PQ. *This algorithm maintains the server at the center of two points p and q , both of which are initially located at the initial server location. Upon each request at location r , if $\delta(p, r) \geq \delta(q, r)$, then p moves to r ; otherwise, q moves to r . The algorithm migrates its server to $s = \frac{p+q}{2}$ after p or q moves.*

Lemma 1. *For any $\rho > 2$ and $p, q, r, s \in \mathbb{R}^2$ such that $p \neq q$, $\delta(p, r) \geq \delta(q, r) > 0$, and s is the center of p and q , $g = \delta(s, r) - \left(\frac{\rho}{2} - \frac{1}{2}\right) \cdot \delta(p, r) - \left(\frac{\rho}{2} - 1\right) \cdot (\delta(q, r) - \delta(p, q))$ is maximized if $\delta(p, r) = \delta(q, r)$, or $\delta(p, q) = \delta(p, r) + \delta(q, r)$, or $\delta(p, r) = \delta(p, q) + \delta(q, r)$.*

Proof. We may assume without loss of generality that $p = (-1, 0)$, $s = (0, 0)$, $q = (1, 0)$, and $r = (\ell \cos \theta, \ell \sin \theta)$ with $0 \leq \theta \leq \pi/2$. Moreover, we fix $\delta(s, r) = \ell$ and regard g as a function of θ . The aim is to prove that g is maximized at $\theta = 0$ or $\theta = \pi/2$.

It follows that:

$$\begin{aligned} \delta(p, r) &= \sqrt{(\ell \cos \theta + 1)^2 + \ell^2 \sin^2 \theta} = \sqrt{\ell^2 + 2\ell \cos \theta + 1} \\ \delta(q, r) &= \sqrt{(\ell \cos \theta - 1)^2 + \ell^2 \sin^2 \theta} = \sqrt{\ell^2 - 2\ell \cos \theta + 1} \\ \frac{d\delta(p, r)}{d\theta} &= \frac{-\ell \sin \theta}{\sqrt{\ell^2 + 2\ell \cos \theta + 1}} = -\frac{\ell \sin \theta}{\delta(p, r)} \end{aligned}$$

and

$$\frac{d\delta(q,r)}{d\theta} = \frac{\ell \sin \theta}{\sqrt{\ell^2 - 2\ell \cos \theta + 1}} = \frac{\ell \sin \theta}{\delta(q,r)}$$

Therefore, we have

$$\frac{dg}{d\theta} = \ell \sin \theta \left\{ \frac{\rho - 1}{2} \cdot \frac{1}{\delta(p,r)} - \left(\frac{\rho}{2} - 1\right) \cdot \frac{1}{\delta(q,r)} \right\} = \frac{(\rho - 1) \cdot \ell \cdot \sin \theta}{2\delta(q,r)} \left\{ \frac{\delta(q,r)}{\delta(p,r)} - \frac{\rho - 2}{\rho - 1} \right\}$$

$\frac{\delta(q,r)}{\delta(p,r)} = \sqrt{\frac{\ell^2 - 2\ell \cos \theta + 1}{\ell^2 + 2\ell \cos \theta + 1}}$ monotonically increases from $\frac{|\ell-1|}{\ell+1}$ to 1 as θ changes from 0 to $\pi/2$. If $\frac{|\ell-1|}{\ell+1} \geq \frac{\rho-2}{\rho-1}$, then $\frac{dg}{d\theta} \geq 0$ for any θ . Therefore, g is maximized at $\theta = \pi/2$. Otherwise, since $0 < \frac{\rho-2}{\rho-1} < 1$, there exists $0 < t < \pi/2$ such that $\sqrt{\frac{\ell^2 - 2\ell \cos t + 1}{\ell^2 + 2\ell \cos t + 1}} = \frac{\rho-2}{\rho-1}$. Since $\frac{dg}{d\theta} \geq 0$ for $\theta \geq t$ and $\frac{dg}{d\theta} \leq 0$ for $\theta < t$, g is maximized at $\theta = 0$ or $\theta = \pi/2$. \square

Theorem 1. PQ is ρ -competitive for $\rho = \frac{11}{4}$ and the page size $D = 1$.

Proof. We use the following potential function for OPT’s server location t , PQ’s server s , and point locations p and q :

$$\Phi = \frac{\rho}{2} \cdot (\delta(p,t) + \delta(q,t)) - \frac{\rho-2}{2} \cdot \delta(p,q)$$

We separate the online events into two parts. The first is to consider only the migration costs incurred by OPT’s server, and the second is to consider the service costs incurred by OPT together with the migration and service costs incurred by PQ. It is sufficient to show that the inequality $\Delta cost_{PQ} + \Delta \Phi \leq \rho \cdot \Delta cost_{OPT}$ follows in both parts, upon each request located at r .

Part 1. The migration of OPT’s server from t to t' induces a change of $\delta(t,t')$ to the total cost of the optimal algorithm OPT but no change to the cost incurred by PQ. The total change of Φ is then $\frac{\rho}{2} \cdot (\delta(p,t') - \delta(p,t) + \delta(q,t') - \delta(q,t))$. Therefore, it is sufficient to show the following inequality:

$$\frac{\rho}{2} \delta(p,t') - \frac{\rho}{2} \delta(p,t) + \frac{\rho}{2} \delta(q,t') - \frac{\rho}{2} \delta(q,t) \leq \frac{\rho}{2} \delta(t,t') + \frac{\rho}{2} \delta(t,t')$$

This follows by the symmetry of the distance function ($\delta(a,b) = \delta(b,a)$) and the triangle inequalities $\delta(p,t') \leq \delta(t,t') + \delta(p,t)$ and $\delta(q,t') \leq \delta(t,t') + \delta(q,t)$.

Part 2. We may assume, without loss of generality, that $\delta(p,r) \geq \delta(q,r)$. By this assumption, PQ moves p to r . Since PQ maintains its server at the center of p and q , the migration cost incurred by PQ is then $\frac{1}{2} \delta(p,r)$. For this part, we have the following equalities:

$$\Delta cost_{PQ} = \delta(s,r) + \frac{1}{2} \delta(p,r)$$

$$\Delta \Phi = \frac{\rho}{2} \cdot (\delta(r,t) - \delta(p,t)) + \left(\frac{\rho}{2} - 1\right) \cdot (\delta(p,q) - \delta(q,r))$$

$$\Delta cost_{OPT} = \delta(t,r)$$

Therefore, we shall show the following inequality:

$$\delta(s,r) + \frac{1}{2} \delta(p,r) + \frac{\rho}{2} \cdot (\delta(r,t) - \delta(p,t)) + \left(\frac{\rho}{2} - 1\right) \cdot (\delta(p,q) - \delta(q,r)) - \rho \delta(t,r) \leq 0$$

Since $\delta(r,t) + \delta(p,t) \geq \delta(p,r)$, it is sufficient to show:

$$\delta(s,r) - \left(\frac{\rho}{2} - \frac{1}{2}\right) \cdot \delta(p,r) - \left(\frac{\rho}{2} - 1\right) \cdot (\delta(q,r) - \delta(p,q)) \leq 0. \tag{1}$$

This follows for the cases $p = q$ and $q = r$, because $s = p = q$ and $\delta(s, r) = \frac{1}{2} \delta(p, r)$, respectively. We assume $p \neq q$ and $q \neq r$. It is sufficient to show that the maximum value of the left-hand side of Equation (1) is less than or equal to zero. If we regard p, q, r , and s as vectors in \mathbb{R}^n , then at most three vectors of them, say p, q , and r , are independent. Therefore, the points p, q, r , and s are on a plane in \mathbb{R}^n . Applying Lemma 1 on this plane, the left-hand side of Equation (1) is maximized in one of three situations. Situation 1: $\delta(p, r) = \delta(q, r)$, Situation 2: $\delta(p, q) = \delta(p, r) + \delta(r, q)$, or Situation 3: $\delta(p, r) = \delta(p, q) + \delta(q, r)$. We proceed to show inequality (1) for each of these three situations. *Situation 1:* Substitution of $\delta(p, q)$ by $2 \cdot \delta(s, q)$ in Equation (1) reveals the following inequality.

$$\delta(s, r) + (\rho - 2) \cdot \delta(s, q) \leq \left(\rho - \frac{3}{2}\right) \cdot \delta(q, r)$$

By applying $\rho = \frac{11}{4}$ and dividing both sides by $\frac{5}{4}$, it is sufficient to show

$$\frac{4}{5} \delta(s, r) + \frac{3}{5} \delta(s, q) \leq \delta(q, r) \tag{2}$$

Since $(\delta(q, r))^2 = (\delta(s, r))^2 + (\delta(s, q))^2$, Equation (2) can be written as:

$$\frac{4}{5} \delta(s, r) + \frac{3}{5} \sqrt{(\delta(q, r))^2 - (\delta(s, r))^2} \leq \delta(q, r).$$

By taking the derivative with respect to $\delta(s, r)$ on the plane containing q, r , and s , the left-hand side of Equation (2) is maximized at $\delta(s, r) = \frac{4}{5} \delta(q, r)$ and $\delta(s, q) = \frac{3}{5} \delta(q, r)$. Therefore, Equation (2) follows.

Situation 2: The inequality in Equation (1) can be rewritten as:

$$\delta(s, r) - \left(\frac{\rho}{2} - 1\right) (\delta(p, r) + \delta(q, r) - \delta(p, q)) \leq \frac{1}{2} \delta(p, r) \tag{3}$$

In this situation, we recall that p, s, r , and q are all located on the same line segment. Since $\delta(p, q) \geq \delta(p, r)$, and s is located at the center of p and q , it follows that $\frac{1}{2} \delta(p, r) = \delta(p, s) = \delta(s, q) = \delta(s, r) + \delta(r, q) \geq \delta(s, r)$. Since $\delta(p, r) + \delta(q, r) - \delta(p, q) = 0$, the inequality in Equation (3) follows.

Situation 3: Since $\rho = \frac{11}{4}$, we rewrite Equation (1) as:

$$\delta(s, r) - \frac{7}{8} \delta(p, r) - \frac{3}{8} \delta(q, r) + \frac{3}{8} \delta(p, q) \leq 0$$

The points p, s, r , and q are also located on the same line segment in this situation. By $\delta(p, r) = \delta(p, q) + \delta(q, r)$, we have $\delta(s, r) - \frac{1}{2} \delta(p, q) - \frac{5}{4} \delta(q, r) \leq 0$. By $\delta(p, q) = 2 \delta(s, q)$ we have $\delta(s, r) - \delta(s, q) - \frac{5}{4} \delta(q, r) \leq 0$. In addition, by $\delta(s, r) = \delta(s, q) + \delta(q, r)$ we have $\delta(q, r) - \frac{5}{4} \delta(q, r) \leq 0$.

Therefore, the proof completes. \square

4. A Lower Bound for PQ

In this section, we show that the exact competitiveness of the algorithm is greater than 2.732. We introduce an adversary through Theorem 2 and prove the existence of such a lower bound. The adversary makes a special sequence of requests on \mathbb{R}^2 against the proposed algorithm. The requests are given at vertices of a triangle that tends to be equilateral. After the second request, the server repeatedly migrates between the center points of two sides of the triangle.

Theorem 2. For a sufficiently large integer n , there exist a request sequence $\sigma = r_1, \dots, r_{2n+1}$, three request locations a, b , and s on a plane, and the initial server location s , such that $cost_{PQ}(s, \sigma) \geq \rho \cdot cost_{OPT}(s, \sigma)$ for $\rho = 1 + \sqrt{3} \approx 2.732$.

Proof. We describe our approach before more technical discussion in the subsequent paragraphs. In Figure 1, we illustrate the behavior of PQ against requests by our adversary. We assume that $\delta(s, b)$ is slightly larger than $\delta(s, a) = \delta(b, a)$. Points u and v are the centers of a and s , and a and b , respectively. For requests $a, b, s, b, s, b, s, \dots$, PQ migrates its server along the dashed line, because p does not move from a . A request occurs at b when the server is at u , and a request occurs at s when the server is at v . Note that a, b , and s tend to be the vertices of an equilateral triangle as $\delta(s, b)$ approaches $\delta(s, a) = \delta(b, a)$.

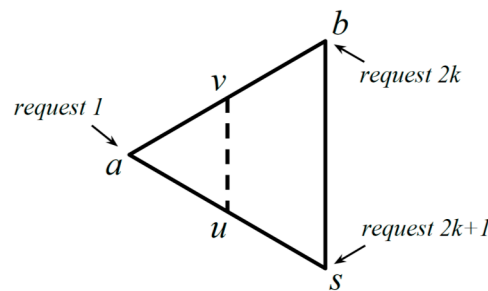


Figure 1. The request sequence $a, b, s, b, s, b, s, \dots$ to access the page at the server; $k \geq 1$.

For $r_1 = a, r_{2k} = b$, and $r_{2k+1} = s$, where $1 \leq k \leq n$ and a, b , and s are the vertices of an equilateral triangle with a unit side length, the cost of the optimal algorithm $cost_{OPT}(s, \sigma)$ is at most $n + 1$ by keeping the server at the initial location s .

The adversary infinitesimally perturbs the distances by slightly increasing the distance between s and b . Upon the first request at a , PQ serves the request by the cost of $\delta(s, a)$ and migrates the server to u , which is the center of s and a , by the cost of $\delta(s, u)$. This is because point p at s moves to a . The $2k$ th request at b is served with the cost of $\delta(u, b)$, causing point q at s to move to b , and the server migrates to v , which is at the center of a and b , with the cost of $\delta(u, v)$. The $(2k + 1)$ st request at s is served with the cost of $\delta(v, s)$. The point q at b moves to s ; hence, the server migrates to u with the cost of $\delta(u, v)$.

We compute the total cost of PQ by using the distances of the unperturbed triangle, since the actual distances (and hence the actual costs of PQ and OPT) may differ by an infinitesimally small amount from these computations. Therefore, we have

$$cost_{PQ}(s, \sigma) = \delta(s, a) + \delta(s, u) + n \cdot (\delta(u, b) + \delta(u, v) + \delta(v, s) + \delta(v, u)) \approx 1.5 + (\sqrt{3} + 1)n$$

Since

$$\lim_{n \rightarrow \infty} \frac{1.5 + (\sqrt{3} + 1)n}{n + 1} = \sqrt{3} + 1$$

then $\frac{cost_{PQ}(s, \sigma)}{cost_{OPT}(s, \sigma)}$ is at least 2.732. This completes the proof. \square

5. Concluding Remarks

In this paper, we proposed a deterministic algorithm for the page migration problem in Euclidean space with a page size $D = 1$. The 2.75-competitiveness of the algorithm is an improvement on the former 2.8 ratio. An adversary was found to express a lower bound of 2.732 for the algorithm. If possible, one could seek to find an algorithm to cover a larger D with better competitiveness than 2.618. Another area of improvement is to narrow the upper and lower bounds of the algorithm. Moreover, the generalization of the algorithm under norms other than the Euclidean and Manhattan ones remains an open problem in this research area.

Acknowledgments: The authors are thankful to the anonymous reviewers who provided valuable and helpful comments. This work was supported by JSPS KAKENHI Grant Number 26330008.

Author Contributions: The design of the algorithm and the proofs were all done together by the authors, step by step.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bienkowski, M. Migrating and replicating data in networks. *Comput. Sci.-Res. Dev.* **2012**, *27*, 169–179. [[CrossRef](#)]
2. Black, D.L.; Sleator, D.D. *Competitive Algorithms for Replication and Migration Problems*; Technical Report CMU-CS-89-201; Carnegie Mellon University: Pittsburgh, PA, USA, 1989.
3. Chrobak, M.; Larmore, L.L.; Reingold, N.; Westbrook, J. Page migration algorithms using work functions. *J. Algorithms.* **1997**, *24*, 124–157. [[CrossRef](#)]
4. Westbrook, J. Randomized algorithms for multiprocessor page migration. *SIAM J. Comput.* **1994**, *23*, 951–965. [[CrossRef](#)]
5. Matsubayashi, A. Uniform page migration on general networks. *Int. J. Pure Appl. Math.* **2008**, *42*, 161–168.
6. Meyer auf der Heide, F.; Vöcking, B.; Westermann, M. Provably Good and Practical Strategies for Non-Uniform Data Management in Networks. In Proceedings of the 7th Annual European Symposium on Algorithms (LNCS 1643), Prague, Czech Republic, 16–18 July 1999; Springer: New York, NY, USA; pp. 89–100.
7. Maggs, B.M.; Meyer auf der Heide, F.; Vöcking, B.; Westermann, M. Exploiting Locality for Data Management in Systems of Limited Bandwidth. In Proceedings of the 38th Annual Symposium on Foundations of Computer Science, Miami Beach, FL, USA, 19–22 October 1997; pp. 284–293.
8. Bartal, Y.; Charikar, M.; Indyk, P. On page migration and other relaxed task systems. *Theor. Comput. Sci.* **2001**, *268*, 43–66. [[CrossRef](#)]
9. Matsubayashi, A. A $3+\Omega(1)$ Lower Bound for Page Migration. In Proceedings of the 2015 Third International Symposium on Computing and Networking, Sapporo, Japan, 8–11 December 2015; pp. 314–320.
10. Matsubayashi, A. Asymptotically optimal online page migration on three points. *Algorithmica* **2015**, *71*, 1035–1064. [[CrossRef](#)]
11. Bartal, Y.; Fiat, A.; Rabani, Y. Competitive algorithms for distributed data management. *J. Comput. Syst. Sci.* **1995**, *51*, 341–358. [[CrossRef](#)]
12. Lund, C.; Reingold, N.; Westbrook, J.; Yan, D. Competitive on-line algorithms for distributed data management. *SIAM J. Comput.* **1999**, *28*, 1086–1111. [[CrossRef](#)]



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).