

## LUMPABILITY IN LRU STACK WITH MARKOVIAN PAGE REFERENCES

Atsuhiko Tanaka      Issei Kino  
*C&C Media Research Laboratories NEC*

(Received April 30, 1997; Revised November 11, 1997)

*Abstract*    The dynamic behavior of programs is studied through LRU (Least-Recently-Used) stack distribution assuming Markovian page references. We propose a new analysis method to calculate steady state probabilities for the LRU stack. We first prove that a Markov chain which describes the behavior of an LRU stack of full-length is lumpable to a Markov chain having a smaller transition matrix than that in the original Markov chain. Based on this lumpability, we propose an algorithm to calculate the invariant vector for the transition matrix of the Markov chain for a full-length LRU stack. We show numerical examples calculated with the proposed algorithm, and evaluate computational efficiency of the proposed method.

### 1. Introduction

An awareness of the influence of dynamic behavior of programs on the performance of computer and/or network systems is essential in the design and development of successful systems [3, 13]. The caching technique was developed based on the observation that programs tend to reference part of the address space more frequently than other parts for during certain periods of time. We can apply the caching technique to increase the execution speed of computer program that accesses secondary storage. Besides its use in operating systems, applications of the caching technique can be found in many computer and/or network systems, such as distributed shared memory[10], file organizing technique[2, 5], and so on. Smith [11] has provided a complete bibliography on those techniques. In the caching technique, the rules used to decide which pages should be moved and when they should be moved are called page replacement algorithms, which have been a subject of considerable interest in computer science for a long time. The LRU (Least-Recently-Used) algorithm is known to be one of the most effective techniques and has been used in many systems. The LRU is known as alternative name MTF (Move-To-Front) in the study of file-self-organizing problems. The LRU stack distance distribution was proposed by Spirn [12] as having the best representation of essential characteristics of program behavior, and has been studied by many authors, e.g. [12, 14]. Most previous studies on the LRU stack algorithms assumed an independent page reference sequence. Knuth [9] pointed out, however, that computational experiments indicate that the MTF (LRU) algorithm works even better than predicted results with the independent-reference assumption, because successive accesses are not independent of each other. Though Markovian-page-reference assumption is more realistic than the independent-reference, not much work has been done in this direction.

Crucial problems in the study for the LRU stack with Markovian page references are how

to reduce space complexity and computation time complexity for a huge size of transition matrix which describes the transition of an full-length LRU stack. Since size of a transition matrix to describe an full-length LRU stack for a  $M$ -page program is  $M! \times M!$ , it becomes immediately intractable to compute the transition probabilities for the Markov chain, even if the page size is a moderate number. For example, the size of transition matrix for an full-length LRU stack is  $3,628,800 \times 3,628,800$  if the program refers to 10 pages.

Chu and Knott [7] studied several general cases of Markovian page reference sequences, and derived steady state probabilities for the LRU stack using the regular-expression method. The regular-expression method can reduce space complexity for the size  $M! \times M!$  matrix to matrices up to a maximum size  $M \times M$ , if the program refers  $M$  pages.

In this paper, we propose an new analysis method to calculate steady state probabilities for the LRU stack. The proposed method, which we call the lumping method, is based on the lumpability of the Markov chain which describes behavior of an full-length LRU stack. The lumping method has the same space complexity as that of the regular-expression method, and has more effective computational complexity than that of the regular-expression method if we choose frequency of multiplications as a measure for the complexity. We first study the lumpability of a Markov chain which describes the state transitions of an LRU stack of full-length, and we prove that the Markov chain is lumpable to a smaller Markov chain. Based on the lumpability, we then propose an algorithm to calculate the invariant vector of the Markov chain for a full-length LRU stack. To reduce matrix size, we developed three sub-techniques, the partial lump method, the block-diagonal decomposition and the recursive reduction technique. Applying those techniques one by one, we can reduce space complexity for the size  $M! \times M!$  matrix to matrices up to a maximum size  $M \times M$ .

The model and notations are given in the next section. The lumpability of the Markov chain is described in Section 3. The technique and algorithm to derive the invariant vector of an full-length LRU stack are described in Sections 4 and 5, respectively. Section 6 has numerical examples of stack distance distributions computed with the proposed technique. We evaluate computational complexity for the regular-expression method and the lumping method, and discuss the efficiency in Section 7.

## 2. Model description and notations

Assume that a program consists of a number of  $M$  pages labeled  $1, 2, \dots, M$ , and page  $j$  is referred to with probability  $p_{ij}$  given that page  $i$  was most recently referred to. Thus the access patterns are time-independent. Assume that  $\sum_{j=1}^M p_{ij} = 1$ , for  $i = 1, 2, \dots, M$ . We write

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{2M} \\ \vdots & \vdots & & \vdots \\ p_{M1} & p_{M2} & \cdots & p_{MM} \end{bmatrix} \quad (2.1)$$

to denote the transition matrix for the Markov chain which generates the page reference sequence.

The LRU stack with stack length  $m$  ( $1 \leq m \leq M$ ) is the ordering of page numbers of a program by recency of reference. We write  $s(a_1, a_2, \dots, a_m)$  to describe the state of the LRU stack with stack length  $m$ , where  $a_i$  indicates the page number of the  $i$ -th position

in the LRU stack for  $i = 1, 2, \dots, m$ . Page  $a_i$  is the  $i$ -th most recently accessed page, so page  $a_1$  is the most recently accessed page. Each time a page is referenced, the LRU (Least-Recently-Used) algorithm it replaces page number  $a_1$  stacked at the head of the stack with the page number of the currently-referred-to page  $a_i$  in the following way:

$$s(a_1, a_2, \dots, a_m) \Rightarrow \begin{cases} s(a_i, a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_m), & i \in \{1, 2, \dots, m\} \\ s(a_i, a_1, a_2, \dots, a_{m-1}), & i \notin \{1, 2, \dots, m\} \end{cases} \quad (2.2)$$

if the state of the stack is  $s(a_1, a_2, \dots, a_m)$  just before the page reference. We define the LRU-operation  $\Lambda_i^{(m)}$  for position  $i$  for a given state vector  $\mathbf{a} = (a_1, a_2, \dots, a_m)$  of an  $m$ -length LRU stack to be

$$\Lambda_i^{(m)}(\mathbf{a}) = \begin{cases} (a_1, a_2, \dots, a_m), & i = 1 \\ (a_i, a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_m), & i = 2, 3, \dots, m-1 \\ (a_i, a_1, a_2, \dots, a_{m-1}), & i = m, m+1, \dots, M \end{cases} \quad (2.3)$$

and the LRU-operation set for an  $m$ -length LRU stack to be  $\Lambda^{(m)}(\mathbf{a}) = \{\Lambda_i^{(m)}(\mathbf{a}) \mid i = 1, 2, \dots, M\}$ . We can describe the transition of states for a full-length  $M$  LRU stack with transition matrix  $Q_M$  in which transition probabilities are defined by

$$q_M(s(\mathbf{a}), s(\mathbf{a}')) = \begin{cases} p_{a_1 a'_1}, & \text{if } \mathbf{a}' \in \Lambda^{(M)}(\mathbf{a}), \\ 0, & \text{otherwise} \end{cases} \quad (2.4)$$

where  $\mathbf{a} = (a_1, a_2, \dots, a_M)$  and  $\mathbf{a}' = (a'_1, a'_2, \dots, a'_M)$  and  $p_{a_1 a'_1}$  is the time-independent probability that page  $a'_1$  is referenced given that page  $a_1$  was just referred to. The order of the state set  $S_M$  of the Markov chain is  $M!$ . We assume that the LRU stack has a steady state.

For simplicity, the notations used in this paper will usually not distinguish between random variables and particular values of them. For example, the notation  $P(s(a_1, a_2, \dots, a_m))$  is used for the probability that the random variables  $(a_1, a_2, \dots, a_m)$  will take generic values of  $(a_1, a_2, \dots, a_m)$ , if the context is such that confusion is unlikely to occur. We denote the steady state probability for the LRU stack with length  $m$  by

$$P(s(a_1, a_2, \dots, a_m)) = \pi(a_1, a_2, \dots, a_m).$$

We say that a page reference has distance  $i$ , if the state of the full-length LRU stack is  $s(a_1, a_2, \dots, a_M)$  just before the page reference, and if the referred to page is  $a_i$ . The stack distance  $\delta$  is a random variable defined by the distance of positions between page numbers which are referred to successively. Formally, using the full-length LRU stack distribution  $\pi$ , the stack distance distribution can be defined in the form

$$P(\delta = j) = \sum_{(a_1, a_2, \dots, a_M) \in \omega[I_M]} p_{a_1 a_j} \pi(a_1, a_2, \dots, a_j, \dots, a_M) \quad (2.5)$$

where  $I_m$  and  $\omega[I_m]$  denote the integer set of  $\{1, 2, \dots, m\}$  and the set of permutations on  $I_m$ , respectively. The stack distance distribution  $P(\delta)$  may more accurately reflect the locality of program behavior than the distribution  $P(s(\mathbf{a}))$  of the LRU stack. Using information on

the stack distance distribution  $P(\delta)$ , we can immediately calculate the page fault rate  $F(C)$  given memory storage size of  $C$  pages in the form

$$F(C) = P(\delta > C). \quad (2.6)$$

Now let us introduce some notation and terminology to be used below. We write  $\mathbf{x}\mathbf{y}$  to denote  $(x_{y_1}, x_{y_2}, \dots, x_{y_m})$  if  $\mathbf{x} = (x_1, x_2, \dots, x_m)$  and  $\mathbf{y} = (y_1, y_2, \dots, y_m)$ , e.g.,  $(2, 3, 1)_{(1,3,2)} = (2, 1, 3)$ . Let  $\sigma[\mathbf{x}]$  be the ordered set of permutations on  $\mathbf{x}$ , i.e.  $\sigma[\mathbf{x}] = \{\mathbf{x}\mathbf{y} | \mathbf{y} \in \omega[I_m]\}$ , where the ordering rule works in such way that  $\mathbf{x}\mathbf{y}$  takes precedence over  $\mathbf{x}\mathbf{z}$  for any  $\mathbf{y}$  and  $\mathbf{z}$  in  $\omega[I_m]$ . This works if the subscript for  $\mathbf{y}$  is smaller than the subscript for  $\mathbf{z}$  when regarding subscript sequence  $(y_1, y_2, \dots, y_m)$  as an  $m$ -digit number  $y_1y_2\dots y_m$  in an  $m$ -ary number system. Let  $\sigma_i[\mathbf{x}]$  be the  $i$ -th element in the ordered set  $\sigma[\mathbf{x}]$ . For example,  $\sigma[(x_1, x_2, x_3)] = \{(x_1, x_2, x_3), (x_1, x_3, x_2), (x_2, x_1, x_3), (x_2, x_3, x_1), (x_3, x_1, x_2), (x_3, x_2, x_1)\}$  and  $\sigma_1[\mathbf{x}] = (x_1, x_2, x_3)$ ,  $\sigma_2[\mathbf{x}] = (x_1, x_3, x_2)$ ,  $\sigma_3[\mathbf{x}] = (x_2, x_1, x_3)$  and so on. Let  $\mu^{(k)}[\mathbf{x}]$  be the ordered set of combinations of  $m$ -elements  $(x_1, x_2, \dots, x_m)$  taken  $k$ -elements at a time, where  $\mathbf{x} = (x_1, x_2, \dots, x_m)$  and the ordering rule works in the same way as that for  $\sigma[\mathbf{x}]$ . Let  $\mu_i^{(k)}[\mathbf{x}]$  be the  $i$ -th element in the ordered set  $\mu^{(k)}[\mathbf{x}]$ . For example, if  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , then  $\mu^{(2)}[\mathbf{x}] = \{(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4), (x_3, x_4)\}$ , and  $\mu_1^{(2)}[\mathbf{x}] = (x_1, x_2)$ ,  $\mu_2^{(2)}[\mathbf{x}] = (x_1, x_3)$ ,  $\mu_3^{(2)}[\mathbf{x}] = (x_1, x_4)$ ,  $\mu_4^{(2)}[\mathbf{x}] = (x_2, x_3)$ , and so on. For  $\mathbf{x} = (x_1, x_2, \dots, x_M)$ , we denote  $\mathbf{x}^{f(m)}$ ,  $\mathbf{x}^{l(m)}$ , and  $\mathbf{x} = (\mathbf{x}^{f(m)}, \mathbf{x}^{l(m)})$  to describe the first part  $(x_1, x_2, \dots, x_m)$  of  $\mathbf{x}$ , the latter part  $(x_{m+1}, x_{m+2}, \dots, x_M)$  of  $\mathbf{x}$ , and the partition of  $\mathbf{x}$  into these parts, respectively. For a state vector  $\mathbf{x} = (x_1, x_2, \dots, x_m)$ , we write  $I_M \setminus \{\mathbf{x}\}$  to denote  $I_M \setminus \{x_1, x_2, \dots, x_m\}$ . For vectors  $\mathbf{x}$  and  $\mathbf{y}$ , the notation  $\mathbf{x} \setminus \mathbf{y}$  describes the difference set  $\{x_1, x_2, \dots, x_m\} \setminus \{y_1, y_2, \dots, y_n\}$ . We write  $\mathbf{x} \ominus \mathbf{y}$  to denote the elimination of  $\mathbf{y}$  out of  $\mathbf{x}$  without changing the order of the original sequence, e.g.  $(a_1, a_2, a_3, a_4) \ominus (a_2, a_4) = (a_1, a_3)$ . The notation  $\mathbf{u} = (1, 2, \dots, M)$  is used to describe the original state vector. We use the notation  $\|\mathbf{x}\| = x_1 + x_2 + \dots + x_m$  to describe the summation of all elements in vector  $\mathbf{x} = (x_1, x_2, \dots, x_m)$ . Finally, we let  $E_n$  be an identity matrix with order  $n$  and we define  $e_n = \overbrace{(1, 1, \dots, 1)}^n$ .

### 3. Lumpability of Markov chains

#### 3.1. Lumpability of the original chain

Since the size of transition matrix  $Q_M$  for a full length LRU stack is  $M! \times M!$ , exact and straightforward numerical computation may be intractable when more than a few (e.g., say ten) pages are involved. To reduce the computations so as to only involve matrices with a maximum size of  $M \times M$ , we first studied the lumpability of the Markov chain for a full length LRU stack. We will use descriptions used in group theory [1], henceforth, to describe states of Markov chains. This causes rather heavy notations to prove the lumpability of the Markov chain. We, however, will have advantage in describing computational algorithms based on the lumpability in a systematic way with the group theory notations.

Let  $G_M$  be a permutation group on the integer set  $I_M$ . If an element  $g$  in  $G_M$  replaces an integer sequence  $(x_1, x_2, \dots, x_M)$  with an integer sequence  $(y_1, y_2, \dots, y_M)$ , then we write the replacement as  $\mathbf{y} = g(\mathbf{x})$  where  $\mathbf{x} = (x_1, x_2, \dots, x_M)$  and  $\mathbf{y} = (y_1, y_2, \dots, y_M)$ . For any  $g_1$  and  $g_2$  in  $G_M$ , we define the product of two permutations  $g_1$  and  $g_2$  in the form

$(g_1 g_2)(\mathbf{a}) = \mathbf{a} \mathbf{y} \mathbf{x}$ , if  $g_1(\mathbf{a}) = \mathbf{a} \mathbf{x}$  and  $g_2(\mathbf{a}) = \mathbf{a} \mathbf{y}$ , e.g.,  $(g_1 g_2)(\mathbf{a}) = \mathbf{a}_{(2,3,1)(1,3,2)} = (a_2, a_1, a_3)$  if  $g_1(a_1, a_2, a_3) = (a_1, a_3, a_2)$  and  $g_2(a_1, a_2, a_3) = (a_2, a_3, a_1)$ .

For  $\mathbf{a} = (a_1, a_2, \dots, a_M)$  and  $1 \leq m \leq M - 1$ , we define

$$\begin{aligned} H^{(m)} &= \{ h_j^{(m)} \mid h_j^{(m)} \in G_M, h_j^{(m)}(\mathbf{a}) = (\sigma_j[\mathbf{a}^{f(m)}], \mathbf{a}^{l(m)}), j = 1, 2, \dots, m! \}, \\ K^{(m)} &= \{ k_n^{(m)} \mid k_n^{(m)} \in G_M, k_n^{(m)}(\mathbf{a}) = (\mathbf{a}^{f(m)}, \sigma_n[\mathbf{a}^{l(m)}]), n = 1, 2, \dots, (M - m)! \}, \\ L^{(m)} &= \{ l_{jn}^{(m)} \mid l_{jn}^{(m)} \in G_M, l_{jn}^{(m)}(\mathbf{a}) = (\sigma_j[\mathbf{a}^{f(m)}], \sigma_n[\mathbf{a}^{l(m)}]), j = 1, 2, \dots, m!, \text{ and} \\ &\quad n = 1, 2, \dots, (M - m)! \}, \\ G^{(m)} &= \{ g_i^{(m)} \mid g_i^{(m)} \in G_M, g_i^{(m)}(\mathbf{a}) = (\mu_i^{(m)}[\mathbf{a}], \mathbf{a} \ominus \mu_i^{(m)}[\mathbf{a}]), i = 1, 2, \dots, \binom{M}{m} \}. \end{aligned}$$

Let

$$j^* = m!, \quad n^* = (M - m)! \quad \text{and} \quad i^* = \binom{M}{m}.$$

Clearly for any value  $m \in I_{M-1}$ ,  $L^{(m)}$  is a subgroup in  $G_M$  and  $K^{(m)}$  is a subgroup in  $L^{(m)}$ . Consequently, by choosing  $K^{(m)}$  and  $L^{(m)}$  as coset leaders for  $L^{(m)}$  and  $G_M$ , we can formulate the respective coset decompositions of the subgroup  $L^{(m)}$  and the group  $G_M$  expressed in the formulas

$$L^{(m)} = h_1^{(m)} K^{(m)} \cup h_2^{(m)} K^{(m)} \cup \dots \cup h_{j^*}^{(m)} K^{(m)} \quad (3.1)$$

$$\text{and} \quad G_M = L^{(m)} g_1^{(m)} \cup L^{(m)} g_2^{(m)} \cup \dots \cup L^{(m)} g_{i^*}^{(m)}. \quad (3.2)$$

Substituting the decomposition form (3.2) into (3.1), we have the overall decomposition form

$$\begin{aligned} G_M &= \bigcup_{i=1}^{i^*} \bigcup_{j=1}^{j^*} h_j^{(m)} K^{(m)} g_i^{(m)} \\ &= \bigcup_{i=1}^{i^*} \bigcup_{n=1}^{n^*} \bigcup_{j=1}^{j^*} h_j^{(m)} k_n^{(m)} g_i^{(m)}. \end{aligned} \quad (3.3)$$

Using the decomposition form (3.3), for any value  $m \in I_{M-1}$ , we can partition the set of LRU stack states  $S_M$  such that

$$S_M = A_1^{(m)} \cup A_2^{(m)} \cup \dots \cup A_{i^*}^{(m)} \quad (3.4)$$

where

$$A_i^{(m)} = A_{i1}^{(m)} \cup A_{i2}^{(m)} \cup \dots \cup A_{ij^*}^{(m)}$$

for  $i = 1, 2, \dots, i^*$  and

$$A_{ij}^{(m)} = \{ s(\mathbf{x}_n) \mid \mathbf{x}_n = (h_j^{(m)} k_n^{(m)} g_i^{(m)})(\mathbf{u}), n = 1, 2, \dots, n^* \} \quad (3.5)$$

for  $i = 1, 2, \dots, i^*$  and  $j = 1, 2, \dots, j^*$ .

We can see here that if we write  $\mathbf{x}_n = (h_j^{(m)} k_n^{(m)} g_i^{(m)})(\mathbf{u})$  for  $n = 1, 2, \dots, n^*$ , then the first  $m$ -sequence of parameters in the  $\mathbf{x}_n$  have the common values

$$\mathbf{x}_n^{f(m)} = ((h_j^{(m)} g_i^{(m)})(\mathbf{u}))^{f(m)}$$

for any value  $n \in I_{n^*}$ . Consequently, all of the elements in the set  $A_{ij}^{(m)}$  have the common  $m$ -sequence of parameters  $(x_1, x_2, \dots, x_m)$  in the first part of state description such that  $s(x_1, x_2, \dots, x_m, *, *, \dots, *) \in A_{ij}^{(m)}$  for any permutation of the other parameters.

Based on the observation, we define the first part of the parameter sequence in the set of  $A_{ij}^{(m)}$  as

$$\mathbf{a}_{ij}^{(m)} = ((h_j^{(m)} g_i^{(m)})(\mathbf{u}))^{f(m)} \quad (3.6)$$

and we respectively define the set  $S^{(m)}$  of  $m$ -length LRU stack states in the forms

$$S^{(m)} = S_1^{(m)} \cup S_2^{(m)} \cup \dots \cup S_{i^*}^{(m)} \quad (3.7)$$

where

$$S_i^{(m)} = \bigcup_{j=1}^{j^*} s(\mathbf{a}_{ij}^{(m)}) \quad \text{for } i = 1, 2, \dots, i^*. \quad (3.8)$$

Note that the order of set  $S^{(m)}$  is  $\frac{M!}{(M-m)!}$ .

We write  $Q_M^{(m)}$  to denote the transition matrix of the Markov chain for which the set of states is  $S^{(m)}$  and the transition probabilities are in the form

$$q_M^{(m)}(s(\mathbf{a}), s(\mathbf{a}')) = \begin{cases} p_{a_1 a'_1}, & \mathbf{a}' \in \Lambda^{(m)}(\mathbf{a}) \\ 0, & \text{otherwise} \end{cases} \quad (3.9)$$

for any pair of states  $\mathbf{a} = (a_1, a_2, \dots, a_m)$  and  $\mathbf{a}' = (a'_1, a'_2, \dots, a'_m)$  in  $S^{(m)}$ , where the sequence of states in the Markov chain are arranged in increasing order with index  $j$  for set  $S_i^{(m)}$  and with index  $i$  for set  $S^{(m)}$ , respectively.

**Theorem 3.1** *A Markov chain with transition matrix  $Q_M$  is lumpable with respect to the partition  $\{A_{i1}^{(m)}, A_{i2}^{(m)}, \dots, A_{ij^*}^{(m)}\}_{i=1}^{i^*}$  and the transition matrix for the lumped process is  $Q_M^{(m)}$ .*

*Proof:* To establish lumpability, the application of Theorem 6.3.2 by Kemeny and Snell [8](p.124), is sufficient and enough to show that  $\sum_{s(\mathbf{a}') \in A_{i'j'}^{(m)}} q_M(s(\mathbf{a}), s(\mathbf{a}'))$  has common values for all  $s(\mathbf{a}) \in A_{ij}^{(m)}$  and for all  $i, i' \in I_{i^*}$ ,  $j, j' \in I_{j^*}$ . Suppose that  $s(\mathbf{a}') = \Lambda_{i'}^{(M)}(\mathbf{a}) \in A_{i'j'}^{(m)}$  and  $s(\mathbf{a}'') = \Lambda_{i''}^{(M)}(\mathbf{a}) \in A_{i''j''}^{(m)}$  for a fixed state  $s(\mathbf{a})$  in  $A_{ij}^{(m)}$ . Since every  $s(\mathbf{a}') \in A_{i'j'}^{(m)}$  and every  $s(\mathbf{a}'') \in A_{i''j''}^{(m)}$  have common sequence  $\mathbf{a}'^{f(m)}$  and  $\mathbf{a}''^{f(m)}$  in the first  $m$ -tuples of their variables, respectively, if  $i' \neq i''$ , then  $\mathbf{a}'^{f(m)} \neq \mathbf{a}''^{f(m)}$ , so that  $A_{i'j'}^{(m)} \neq A_{i''j''}^{(m)}$ . Thus the transition probability  $p_{a_1 a'_1}$  appears once and only once at the corresponding position of the transition from  $s(\mathbf{a}) \in A_{ij}^{(m)}$  to  $s(\mathbf{a}') \in A_{i'j'}^{(m)}$ . We see that  $\Lambda_l^{(M)}(\mathbf{a}_1)^{f(m)} = \Lambda_l^{(M)}(\mathbf{a}_2)^{f(m)}$  if  $\mathbf{a}_1$  and  $\mathbf{a}_2$  are elements in the set  $A_{ij}^{(m)}$  for a fixed reference position  $l$ , so that if  $s(\mathbf{a}'_1) = \Lambda_l^{(M)}(\mathbf{a}_1)$  and  $s(\mathbf{a}'_2) \in A_{i'j'}^{(m)}$ , then  $s(\mathbf{a}'_2) = \Lambda_l^{(M)}(\mathbf{a}_2)$  is an element of the same state set  $A_{i'j'}^{(m)}$ . Therefore  $\sum_{s(\mathbf{a}') \in A_{i'j'}^{(m)}} q_M(s(\mathbf{a}), s(\mathbf{a}'))$  has common values for all  $s(\mathbf{a}) \in A_{ij}^{(m)}$  and for all  $i, i' \in I_{i^*}$ ,  $j, j' \in I_{j^*}$ . ■ (See Appendix for example.)

**Corollary 3.1** *The Markov chain with transition matrix  $Q_M$  is lumpable to the Markov chain with  $M \times M$  transition matrix  $P$  of the page reference Markov chain.*

*Proof:* The statement follows by choosing  $m = 1$  in Theorem 3.1. ■

### 3.2. Lumpability of the lumped chain

We will now show that the lumped Markov chain with transition matrix  $Q_M^{(m)}$  is again lumpable with respect to alternative partitions. The state set  $S^{(m)}$  of the Markov chain was defined by (3.7) in which states were arranged in increasing order with indexes  $j$  and  $i$ . We will consider an alternative arrangement for states in set  $S^{(m)}$  based on an alternative decomposition form. Since  $K^{(m)}$  is a subgroup of  $K^{(m-1)}$ , by choosing  $K^{(m-1)}$  as the coset leader, we have the coset decomposition form

$$K^{(m-1)} = K^{(m)}g_1^{(m)} \cup K^{(m)}g_2^{(m)} \cup \dots \cup K^{(m)}g_{M-m+1}^{(m)}. \quad (3.10)$$

Substituting the decomposition form (3.10) into the decomposition form

$$G_M = \bigcup_i^{i^{**}} \bigcup_j^{j^{**}} h_j^{(m-1)} K^{(m-1)} g_i^{(m-1)}$$

we have the alternative decomposition form

$$\begin{aligned} G_M &= \bigcup_{i=1}^{i^{**}} \bigcup_{k=1}^{k^{**}} \bigcup_{j=1}^{j^{**}} h_j^{(m-1)} K^{(m)} g_k^{(m)} g_i^{(m-1)} \\ &= \bigcup_{i=1}^{i^{**}} \bigcup_{k=1}^{k^{**}} \bigcup_{n=1}^{n^*} \bigcup_{j=1}^{j^{**}} h_j^{(m-1)} k_n^{(m)} g_k^{(m)} g_i^{(m-1)} \end{aligned} \quad (3.11)$$

where

$$i^{**} = \binom{M}{m-1}, \quad j^{**} = (m-1)! \quad \text{and} \quad k^{**} = M - m + 1.$$

Using the decomposition form (3.11), for any value  $m \in I_{M-1}$ , we can partition the set of LRU stack states  $S_M$  alternative form such that

$$S_M = B_1^{(m)} \cup B_2^{(m)} \cup \dots \cup B_{i^{**}}^{(m)} \quad (3.12)$$

$$B_i^{(m)} = B_{i1}^{(m)} \cup B_{i2}^{(m)} \cup \dots \cup B_{ij^{**}}^{(m)} \quad (3.13)$$

and

$$B_{ij}^{(m)} = B_{ij1}^{(m)} \cup B_{ij2}^{(m)} \cup \dots \cup B_{ijk^{**}}^{(m)} \quad (3.14)$$

for  $i = 1, 2, \dots, i^{**}$ ,  $j = 1, 2, \dots, j^{**}$  where

$$B_{ijk}^{(m)} = \{ s(\mathbf{x}_n) \mid \mathbf{x}_n = ((h_j^{(m-1)} k_n^{(m)} g_k^{(m)} g_i^{(m-1)})(\mathbf{u}), n = 1, 2, \dots, n^* \}. \quad (3.15)$$

We can see that if we write  $\mathbf{x}_n = (h_j^{(m-1)} k_n^{(m)} g_k^{(m)} g_i^{(m-1)})(\mathbf{u})$  for  $n = 1, 2, \dots, n^*$ , then the first  $m$ -sequence of parameters in the  $\mathbf{x}_n$  have the common values, i.e.,

$$\mathbf{x}_n^{f(m)} = ((h_j^{(m-1)} g_k^{(m)} g_i^{(m-1)})(\mathbf{u}))^{f(m)}$$

for any value  $n \in I_{n^*}$ . Consequently, all of the elements in the set  $B_{ijk}^{(m)}$  have the common  $m$ -tuples of parameters  $(x_1, x_2, \dots, x_m)$  in the first part of state description such that  $s(x_1, x_2, \dots, x_m, *, *, \dots, *) \in B_{ijk}^{(m)}$  for any permutation of the other parameters.

Based on the observation, we define the first  $m$ -tuples of parameters in the set of  $B_{ijk}^{(m)}$  as

$$\mathbf{b}_{ijk}^{(m)} = ((h_j^{(m-1)} g_k^{(m)} g_i^{(m-1)})(\mathbf{u}))^{f(m)} \quad (3.16)$$

and we respectively define the alternative set of  $m$ -length LRU stack states in the form

$$S^{<m>} = S_1^{<m>} \cup S_2^{<m>} \cup \dots \cup S_{i^{**}}^{<m>} \quad (3.17)$$

and, for  $i = 1, 2, \dots, i^{**}$ ,

$$S_i^{<m>} = S_{i1}^{<m>} \cup S_{i2}^{<m>} \cup \dots \cup S_{ij^{**}}^{<m>} \quad (3.18)$$

where

$$S_{ij}^{<m>} = \bigcup_{k=1}^{k^{**}} s(\mathbf{b}_{ijk}^{(m)}) \quad (3.19)$$

for  $i = 1, 2, \dots, i^{**}$  and  $j = 1, 2, \dots, j^{**}$ .

Note that

$$\begin{aligned} (\mathbf{b}_{ijk}^{(m)})^{f(m-1)} &= ((h_j^{(m-1)} g_k^{(m)} g_i^{(m-1)})(\mathbf{u}))^{f(m-1)} \\ &= ((h_j^{(m-1)} g_i^{(m-1)})(\mathbf{u}))^{f(m-1)} \\ &= \mathbf{a}_{ij}^{(m-1)} \end{aligned} \quad (3.20)$$

for  $i = 1, 2, \dots, i^{**}$  and  $j = 1, 2, \dots, j^{**}$ .

We write  $Q_M^{<m>}$  to denote a transition matrix in which the order of states is rearranged according to the set  $S^{<m>}$  of states. Note that a Markov chain with the transition matrix  $Q_M^{(m)}$  and a Markov chain with the transition matrix  $Q_M^{<m>}$  are essentially the same except for the arrangement of states in both transition matrices.

**Theorem 3.2** *A Markov chain with transition matrix  $Q_M^{<m>}$  is lumpable with respect to the partition  $\{S_1^{<m>}, S_2^{<m>}, \dots, S_{i^{**}}^{<m>}\}$  and the transition matrix for the lumped process is  $Q_M^{(m-1)}$ .*

Proof: Using almost the same method as in the proof of Theorem 3.1, one can easily show that  $\sum_{s(\mathbf{a}') \in S_{ij'}^{<m>}} q_M^{(m)}(s(\mathbf{a}), s(\mathbf{a}'))$  has common values for all  $s(\mathbf{a}) \in S_{ij}^{<m>}$  and for all  $i, i' \in I_{i^{**}}, j, j' \in I_{j^{**}}$ . Applying the relation (3.20), one can see that the state set  $S_i^{<m>}$  is reduced to the state set  $S_i^{(m-1)}$  after the lumping procedure, i.e.,  $S_i^{<m>} \Rightarrow S_i^{(m-1)}$  for  $i = 1, 2, \dots, i^{**}$ . The statement follows by again applying Theorem 6.3.2 of Kemeny and Snell [8]. ■

This result provides the foundation of the partial lump technique described in next subsection.

### 3.3. Partial lump of the lumped chain

In the Theorem 3.2, we applied the lumping procedure for the Markov chain with transition matrix  $Q_M^{<m>}$  to all of the partitions  $\{S_1^{<m>}, S_2^{<m>}, \dots, S_{i^{**}}^{<m>}\}$ .

We can also apply the procedure not to all of the partitions but to a part of them, say  $\{S_1^{<m>}, S_2^{<m>}, \dots, S_r^{<m>}\}$ . As the result of the partial lump procedure for the Markov



chain with transition matrix  $Q_M^{<m>}$ , we have the partially lumped Markov chain denoted by  $Q_M^{[1,2,\dots,r]}$  with the state set

$$S_1^{(m-1)} \cup S_2^{(m-1)} \cup \dots \cup S_r^{(m-1)} \cup S_{r+1}^{<m>} \cup S_{r+2}^{<m>} \cup \dots \cup S_{i^{**}}^{<m>} .$$

We write the partition for the state set in the form

$$S_{[1,2,\dots,r]}^{<m>} = \{ S_1^{(m-1)}, S_2^{(m-1)}, \dots, S_r^{(m-1)}, S_{r+1}^{<m>}, S_{r+2}^{<m>}, \dots, S_{i^{**}}^{<m>} \}.$$

For general cases, we have the following theorem.

**Theorem 3.3** *The Markov chain  $Q_M^{<m>}$  is lumpable with respect to the partition  $S_{[i_1, i_2, \dots, i_r]}^{<m>}$  where  $i_1, i_2, \dots, i_r \in I_{i^{**}}$ .*

Proof: The proof of the Theorem immediately follows those of Theorems 3.1 and 3.2. ■

#### 4. Invariant vector of the Markov chain

We consider a method to calculate the invariant steady-state vector of the Markov chain  $Q_M$  in this section. The invariant vector can be obtained by using combination of three techniques. These are the partial lump method based on the lumpability of the Markov chain, the block-diagonal decomposition of the principal sub-matrix of the lumped chain and the recursive reduction technique.

##### 4.1. Partial lump method

We consider a partition  $S_{[2,3,\dots,i^{**}]}^{<m>} = \{ S_1^{<m>}, S_2^{(m-1)}, S_3^{(m-1)}, \dots, S_{i^{**}}^{(m-1)} \}$  for a lumped Markov chain with transition matrix  $Q_M^{<m>}$ . Applying Theorem 3.2 derived in the previous section, we can see that the lumped Markov chain with transition matrix  $Q_M^{<m>}$  is lumpable with respect to the partition  $S_{[2,3,\dots,i^{**}]}^{<m>}$ . We write  $Q_M^{[2,3,\dots,i^{**}]}$  to denote the transition matrix for the Markov chain partially lumped with respect to the partition  $S_{[2,3,\dots,i^{**}]}^{<m>}$ .

We decompose the state set into two parts  $V_1 = S_1^{<m>}$  and  $V_{[1]} = S_2^{(m-1)} \cup S_3^{(m-1)} \cup \dots \cup S_{i^{**}}^{(m-1)}$ , and we write the transition matrix in the form

$$Q_M^{[2,3,\dots,i^{**}]} = \begin{matrix} & V_1 & V_{[1]} \\ \begin{matrix} V_1 \\ V_{[1]} \end{matrix} & \begin{pmatrix} \Phi_{11}^{(m)} & \Phi_{1[1]}^{(m-1)} \\ \Phi_{[1]1}^{(m-1)} & \Phi_{[1][1]}^{(m-1)} \end{pmatrix} \end{matrix} \quad (4.1)$$

where  $\Phi_{11}^{(m)}$ ,  $\Phi_{1[1]}^{(m-1)}$ ,  $\Phi_{[1]1}^{(m-1)}$  and  $\Phi_{[1][1]}^{(m-1)}$  are transition submatrices for transitions  $V_1 \rightarrow V_1$ ,  $V_1 \rightarrow V_{[1]}$ ,  $V_{[1]} \rightarrow V_1$  and  $V_{[1]} \rightarrow V_{[1]}$ , respectively. We write  $\tilde{\pi}(S)$  to denote an invariant vector associated with state set  $S$ , e.g.  $\tilde{\pi}(S) = (\pi(s(1,2)), \pi(s(2,1)))$  if  $S = \{s(1,2), s(2,1)\}$ . Since the invariant vector  $(\tilde{\pi}(V_1), \tilde{\pi}(V_{[1]}))$  of the transition matrix  $Q_M^{[2,3,\dots,i^{**}]}$  satisfies the equation

$$(\tilde{\pi}(V_1), \tilde{\pi}(V_{[1]})) = (\tilde{\pi}(V_1), \tilde{\pi}(V_{[1]})) Q_M^{[2,3,\dots,i^{**}]} \quad (4.2)$$

by applying relation (4.1), we obtain the equation

$$\tilde{\pi}(V_1) = \tilde{\pi}(V_1) \Phi_{11}^{(m)} + \tilde{\pi}(V_{[1]}) \Phi_{[1]1}^{(m-1)} \quad (4.3)$$

from which we can derive  $\tilde{\pi}(V_1)$  in the form

$$\tilde{\pi}(V_1) = \tilde{\pi}(V_{[1]})\Phi_{[1]1}^{(m-1)}(E - \Phi_{11}^{(m)})^{-1} \quad (4.4)$$

where  $E$  is an identity matrix with the same order as that of  $\Phi_{11}^{(m)}$ . Since  $(\tilde{\pi}(S_1^{(m-1)}), \tilde{\pi}(V_{[1]}))$  is the invariant vector of transition matrix  $Q_M^{(m-1)}$ , if we have an invariant vector for transition matrix  $Q_M^{(m-1)}$ , then we can calculate invariant vector  $\tilde{\pi}(V_1)$  ( $= \tilde{\pi}(S_1^{<m>})$ ) through equation (4.4). In a similar way, we can calculate  $\tilde{\pi}(S_i^{<m>})$  for  $i = 2, 3, \dots, i^{**}$  by choosing partition  $S_{[1,2,\dots,i-1,i+1,\dots,i^{**}]}^{<m>}$  instead of partition  $S_{[2,3,\dots,i^{**}]}^{<m>}$ . Thus, by using the invariant vector of transition matrix  $Q_M^{(m-1)}$ , we can obtain all values  $\tilde{\pi}(S_1^{<m>}), \tilde{\pi}(S_2^{<m>}), \dots, \tilde{\pi}(S_{i^{**}}^{<m>})$ , which are elements of invariant vector of the transition matrix  $Q_M^{<m>}$  and which are equivalent to the elements of the invariant vector of transition matrix  $Q_M^{(m)}$ .

From Corollary 3.1, the invariant vector of transition matrix  $Q_M^{(1)}$  is equivalent to the invariant vector of transition matrix  $P$ . Starting from this initial vector, we can calculate the invariant vectors of  $Q_M^{(2)}, Q_M^{(3)}, \dots, Q_M^{(M)}$  step by step.

#### 4.2. Block-diagonal decomposition

The size of  $\Phi_{11}^{(m)}$ , however, may become very large as the stack size  $m$  increases because the number of states in  $S_1^{<m>}$  is  $(m-1)! \times (M-m+1)$ . A technique to reduce the size of  $\Phi_{11}^{(m)}$  is required. We decompose the state set  $S_1^{(m-1)}$  of Markov chain with transition matrix  $Q_M^{(m-1)}$  into  $T_1 = S_1^{(m-1)}$  and  $T_{[1]} = S_2^{(m-1)} \cup S_3^{(m-1)} \cup \dots \cup S_{i^{**}}^{(m-1)}$ , and we write the transition matrix in the form

$$Q_M^{(m-1)} = \begin{matrix} & T_1 & T_{[1]} \\ \begin{matrix} T_1 \\ T_{[1]} \end{matrix} & \begin{pmatrix} Q_{11}^{(m-1)} & Q_{1[1]}^{(m-1)} \\ Q_{[1]1}^{(m-1)} & Q_{[1][1]}^{(m-1)} \end{pmatrix} \end{matrix} \quad (4.5)$$

Comparing this with transition matrices  $Q_M^{[2,3,\dots,i^{**}]}$  and  $Q_M^{(m-1)}$ , we can see that their submatrices have the following relations:

$$\Phi_{11}^{(m)} = Q_{11}^{(m-1)} \otimes E_{M-m+1} \quad (4.6)$$

$$\Phi_{1[1]}^{(m-1)} = Q_{1[1]}^{(m-1)} \otimes \mathbf{e}_{M-m+1}^t \quad \text{and} \quad \Phi_{[1][1]}^{(m-1)} = Q_{[1][1]}^{(m-1)} \quad (4.7)$$

where operator  $\otimes$  indicates the Kronecker product [6].

Since  $\cup_{k=1}^{j^{**}} \mathbf{b}_{1jk}^{(m)} = \cup_{l=m}^M (\mathbf{a}_{1j}^{(m-1)}, a_l)$ , we can describe  $S_1^{<m>}$  in the form

$$S_1^{<m>} = \bigcup_{j=1}^{j^{**}} \bigcup_{k=m}^M s(\mathbf{a}_{1j}^{(m-1)}, a_k) \quad (4.8)$$

Changing the order of arrangement of states in  $S_1^{<m>}$  such that

$$S_1^{<m>*} = \bigcup_{k=m}^M \bigcup_{j=1}^{j^{**}} s(\mathbf{a}_{1j}^{(m-1)}, a_k) \quad (4.9)$$

and using relation (4.6), we can see that submatrix  $\Phi_{11}^{(m)}$  can be re-formed as  $\Phi_{11}^{(m)*}$  in the form

$$\Phi_{11}^{(m)*} = \text{block-diagonal} \left\{ \overbrace{Q_{11}^{(m-1)}, Q_{11}^{(m-1)}, \dots, Q_{11}^{(m-1)}}^{M-m+1} \right\} \quad (4.10)$$

We denote  $\mathbf{a}_{1j}^{(m-1)} = (\alpha_{1j}, \mathbf{b}_{1j}^{(m-1)})$  where  $\alpha_{1j} = (\mathbf{a}_{1j}^{(m-1)})^{f(1)}$  and  $\mathbf{b}_{1j}^{(m-1)} = (\mathbf{a}_{1j}^{(m-1)})^{l(1)}$ , and denote

$$V_{[1]}^* = \bigcup_{k=m}^M \bigcup_{j=1}^{j^{**}} s(\mathbf{b}_{1j}^{(m-1)}, a_k) \quad \text{and} \quad \bar{V}_{[1]}^* = V_{[1]} \setminus V_{[1]}^*. \quad (4.11)$$

We can see that transition from the state in  $V_{[1]}$  to state  $s(\mathbf{a}_{1j}^{(m-1)}, a_k)$  in  $V_1$  may occur only from state  $s(\mathbf{b}_{1j}^{(m-1)}, a_k)$  in  $V_{[1]}^*$  with transition probability  $p_{\beta_{1j}\alpha_{1j}}$  where  $\beta_{1j}^{(m-1)} = ((\mathbf{a}_{1j}^{(m-1)})^{l(1)})^{f(1)}$ , and that no transition may occur from any other state in  $\bar{V}_{[1]}^*$ .

Consequently, we can rewrite  $\Phi_{[1]1}^{(m-1)}$  as  $\Phi_{[1]1}^{(m-1)*}$  in the form

$$\Phi_{[1]1}^{(m-1)*} = \begin{matrix} S_1^{<m>*} \\ V_{[1]}^* \\ \bar{V}_{[1]}^* \end{matrix} \begin{pmatrix} R_{[1]1}^{(m-1)} \\ \mathbf{0} \end{pmatrix} \quad (4.12)$$

where

$$R_{[1]1}^{(m-1)} = \text{block-diagonal} \{ \overbrace{R_1^{(m-1)}(\mathbf{a}_{11}^{(m-1)}), R_1^{(m-1)}(\mathbf{a}_{11}^{(m-1)}), \dots, R_1^{(m-1)}(\mathbf{a}_{11}^{(m-1)})}^{M-m+1} \}$$

$$R_1^{(m-1)}(\mathbf{a}_{11}^{(m-1)}) = \text{block-diagonal} \{ R_{11}^{(m-1)}(\mathbf{a}_{11}^{(m-1)}), R_{12}^{(m-1)}(\mathbf{a}_{11}^{(m-1)}), \dots, R_{1,m-1}^{(m-1)}(\mathbf{a}_{11}^{(m-1)}) \}$$

and

$$R_{1l}^{(m-1)}(\mathbf{a}_{11}^{(m-1)}) = \text{diagonal} \{ p_{sl} \}_{s=1, s \neq l}^{m-1} \otimes E_{(m-3)!} \quad \text{for } l = 1, 2, \dots, m-1.$$

We use the following notations henceforth:

$$\tilde{s}(\mathbf{a}, [\mathbf{x}], \mathbf{b}) = \left( s(\mathbf{a}, \sigma_i[\mathbf{x}], \mathbf{b}) \right)_{i=1}^{m!} \quad \text{and} \quad \tilde{\pi}(\mathbf{a}, [\mathbf{x}], \mathbf{b}) = \left( \pi(\mathbf{a}, \sigma_i[\mathbf{x}], \mathbf{b}) \right)_{i=1}^{m!} \quad (4.13)$$

for arbitrary vectors  $\mathbf{a}$ ,  $\mathbf{b}$  and  $\mathbf{x} = (x_1, x_2, \dots, x_m)$ , e.g.  $\tilde{s}(a_1, [a_2, a_3], a_4) = (s(a_1, a_2, a_3, a_4), s(a_1, a_3, a_2, a_4))$  and  $\tilde{\pi}(a_1, [a_2, a_3], a_4) = (\pi(a_1, a_2, a_3, a_4), \pi(a_1, a_3, a_2, a_4))$ . Note that using the notation, and applying the definition for state descriptions  $\mathbf{a}_{1j}^{(m-1)}$  and  $\mathbf{b}_{1j}^{(m-1)}$ , we can describe the state vector in the forms

$$\left( (\tilde{\pi}(\mathbf{a}_{1j}^{(m-1)}, a_k)) \right)_{j=1}^{j^{**}} = \tilde{\pi}([\mathbf{a}_{1j}^{(m-1)}], a_k) \quad \text{and} \quad \left( (\tilde{\pi}(\mathbf{b}_{1j}^{(m-1)}, a_k)) \right)_{j=1}^{j^{**}} = \tilde{\pi}([\mathbf{b}_{1j}^{(m-1)}], a_k).$$

Replacing  $\tilde{\pi}(V_1)$  with  $\tilde{\pi}(S_1^{<m>*})$ ,  $\tilde{\pi}(V_1)\Phi_{11}^{(m)}$  with  $\tilde{\pi}(S_1^{<m>*})\Phi_{11}^{(m)*}$  and  $\tilde{\pi}(V_{[1]})\Phi_{[1]1}^{(m)}$  with  $(\tilde{\pi}(V_{[1]}^*), \tilde{\pi}(\bar{V}_{[1]}^*))\Phi_{[1]1}^{(m)*}$  in equation (4.3), and using the relation

$$\tilde{\pi}(S_1^{<m>*})\Phi_{11}^{(m)*} = \left( \tilde{\pi}([\mathbf{a}_{1j}^{(m-1)}], a_k) \mathcal{Q}_{11}^{(m-1)} \right)_{k=m}^M$$

$$\begin{aligned} \text{and} \quad (\tilde{\pi}(V_{[1]}^*), \tilde{\pi}(\bar{V}_{[1]}^*))\Phi_{[1]1}^{(m)*} &= (\tilde{\pi}(V_{[1]}^*))R_{[1]1}^{(m-1)} \\ &= \left( \tilde{\pi}([\mathbf{b}_{1j}^{(m-1)}], a_k) R_1^{(m-1)}(\mathbf{a}_{11}^{(m-1)}) \right)_{k=m}^M, \end{aligned}$$

we can separate equation (4.3) into a number  $M - m + 1$  equations

$$\tilde{\pi}([\mathbf{a}_{1j}^{(m-1)}], a_k) = \tilde{\pi}([\mathbf{a}_{1j}^{(m-1)}], a_k)\mathcal{Q}_{11}^{(m-1)} + \tilde{\pi}([\mathbf{b}_{1j}^{(m-1)}], a_k)R_1^{(m-1)}(\mathbf{a}_{11}^{(m-1)}) \quad (4.14)$$

for  $k = m, m + 1, \dots, M$ . From equation (4.14), we can derive state probabilities

$$\tilde{\pi}([\mathbf{a}_{1j}^{(m-1)}], a_k) = \tilde{\pi}([\mathbf{b}_{1j}^{(m-1)}], a_k) R_1^{(m-1)}(\mathbf{a}_{11}^{(m-1)}) (E_{(m-1)!} - \mathcal{Q}_{11}^{(m-1)})^{-1} \quad (4.15)$$

for  $k = m, m + 1, \dots, M$ . N

Note that  $\tilde{\pi}([\mathbf{b}_{1j}^{(m-1)}], a_k)$  is equivalent to  $\tilde{\pi}(V_{[1]})$  which is a part of the invariant vector of transition matrix  $\mathcal{Q}_M^{(m-1)}$  and  $\tilde{\pi}([\mathbf{a}_{1j}^{(m-1)}], a_k)$  is a part of the invariant vector of transition matrix  $\mathcal{Q}_M^{(m)}$ .

One can easily extend the derivation for  $\tilde{\pi}([\mathbf{a}_{1j}^{(m-1)}], a_k)$  to other parts of invariant vectors  $\tilde{\pi}([\mathbf{a}_{ij}^{(m-1)}], a_k)$  for  $i = 2, 3, \dots, M$  by choosing  $(V_i, V_{[i]})$  and  $(T_i, T_{[i]})$  instead of  $(V_1, V_{[1]})$  in (4.1) and  $(T_1, T_{[1]})$  in (4.5), respectively. In general, we have the equation

$$\tilde{\pi}([\mathbf{a}_{ij}^{(m-1)}], a_k) = \tilde{\pi}([\mathbf{a}_{ij}^{(m-1)}], a_k) \mathcal{Q}_{ii}^{(m-1)} + \tilde{\pi}([\mathbf{b}_{ij}^{(m-1)}], a_k) R_i^{(m-1)}(\mathbf{a}_{i1}^{(m-1)}) \quad (4.16)$$

and the solution

$$\begin{aligned} \tilde{\pi}([\mathbf{a}_{ij}^{(m-1)}], a_k) &= \tilde{\pi}([\mathbf{b}_{ij}^{(m-1)}], a_k) R_i^{(m-1)}(\mathbf{a}_{i1}^{(m-1)}) (E_{(m-1)!} - \mathcal{Q}_{ii}^{(m-1)})^{-1} \\ &= \left( \tilde{\pi}([\mathbf{a}_{i1}^{(m-1)}] \ominus a_s, a_k) R_{is}^{(m-1)}(\mathbf{a}_{i1}^{(m-1)}) \right)_{s=1}^{m-1} \\ &\quad \times (E_{(m-1)!} - \mathcal{Q}_{ii}^{(m-1)})^{-1} \end{aligned} \quad (4.17)$$

for  $i = 1, 2, \dots, i^{**}$  and for  $k = m, m + 1, \dots, M$ , where  $\mathbf{a}_{ij}^{(m-1)} = ((h_j^{(m-1)} g_i^{(m-1)})(\mathbf{u}))$ ,  $\mathbf{a}_{i1}^{(m-1)} = (a_1, a_2, \dots, a_{m-1})$ ,  $\mathbf{b}_{ij}^{(m-1)} = (\mathbf{a}_{ij}^{(m-1)})^{l(1)}$ ,  $\mathcal{Q}_{ii}^{(m-1)}$  is the transition submatrix in  $\mathcal{Q}_M^{(m-1)}$  associated with transition from the state in  $S_i^{(m-1)}$  to the state in  $S_i^{(m-1)}$  itself,

$$R_i^{(m-1)}(\mathbf{a}_{i1}^{(m-1)}) = \text{block-diagonal} \{ R_{i1}^{(m-1)}(\mathbf{a}_{i1}^{(m-1)}), R_{i2}^{(m-1)}(\mathbf{a}_{i1}^{(m-1)}), \dots, R_{i,m-1}^{(m-1)}(\mathbf{a}_{i1}^{(m-1)}) \}$$

and

$$R_{il}^{(m-1)}(\mathbf{a}_{i1}^{(m-1)}) = \text{diagonal} \{ p_{a_s a_i} \}_{s=1, s \neq l}^{m-1} \otimes E_{(m-3)!} \quad \text{for } l = 1, 2, \dots, m-1.$$

### 4.3. Recursive reduction

We define  $(m-1)! \times (m-1)(m-2) \cdots (m-r)$  dimensional matrix  $W_r^{(m-1)}$  as

$$W_r^{(m-1)} = E_{(m-1)(m-2) \cdots (m-r)} \otimes \mathbf{e}_{(m-r-1)!}^t.$$

For simplicity of notation, we denote  $\mathbf{a}_{i1}^{(m-1)} = \mathbf{a}_{m-1} = (a_1, a_2, \dots, a_{m-1})$ ,  $\mathbf{a}_r = \mathbf{a}_{m-1}^{f(r)} = (a_1, a_2, \dots, a_r)$  for  $1 \leq r \leq m-1$ , and summation of state probabilities in the form

$$\sum_{\mathbf{y} \in \sigma[\mathbf{x}]} \pi(\mathbf{a}, \mathbf{y}, \mathbf{b}) = \pi(\mathbf{a}, (\mathbf{x}), \mathbf{b})$$

e.g.  $\pi(a_1, (a_2, a_3), a_4) = \pi(a_1, a_2, a_3, a_4) + \pi(a_1, a_3, a_2, a_4)$ . Let

$$\tau^{(r)}(\mathbf{x}) = \pi(\mathbf{x}, (\mathbf{a}_r \setminus \mathbf{x}), \mathbf{a}_{m-1}^{l(r)}, a_k), \quad \nu^{(r)}(\mathbf{x}) = \pi(\mathbf{x}^{l(1)}, (\mathbf{a}_r \setminus \mathbf{x}), \mathbf{a}_{m-1}^{l(r)}, a_k)$$

and

$$\begin{aligned} \varphi^{(r)}(\mathbf{x}) &= \tau^{(r)}(\mathbf{x}^{l(1)}) + \nu^{(r)}(\mathbf{x}) \\ &= \pi(\mathbf{x}^{l(1)}, (\mathbf{a}_r \setminus \mathbf{x}^{l(1)}), \mathbf{a}_{m-1}^{l(r)}, a_k) + \pi(\mathbf{x}^{l(1)}, (\mathbf{a}_r \setminus \mathbf{x}), \mathbf{a}_{m-1}^{l(r)}, a_k) \end{aligned}$$

where  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ ,  $x_1, x_2, \dots, x_n \in \{a_1, a_2, \dots, a_r\}$  and  $x_i \neq x_j$  if  $i \neq j$ .

For example,

$$\begin{aligned}\tau^{(m-1)}(a_1) &= \pi(a_1, (a_2, \dots, a_{m-1}), a_k) \\ \tau^{(m-1)}(a_2) &= \pi(a_2, (a_1, a_3, \dots, a_{m-1}), a_k) \\ \tau^{(m-1)}(a_1, a_2) &= \pi(a_1, a_2, (a_3, \dots, a_{m-1}), a_k) \\ \tau^{(3)}(a_1) &= \pi(a_1, (a_2, a_3), a_4, \dots, a_{m-1}, a_k) \\ \nu^{(m-1)}(a_1, a_2) &= \pi(a_2, (a_3, \dots, a_{m-1}), a_k) \\ \nu^{(4)}(a_1, a_2) &= \pi(a_2, (a_3, a_4), a_5, \dots, a_{m-1}, a_k) \\ \varphi^{(m-1)}(a_1, a_2, a_3) &= \pi(a_2, a_3, (a_1, a_4, \dots, a_{m-1}), a_k) + \pi(a_2, a_3, (a_4, \dots, a_{m-1}), a_k) \\ \varphi^{(4)}(a_1, a_2) &= \pi(a_2, (a_1, a_3, a_4), a_5, \dots, a_{m-1}, a_k) + \pi(a_2, (a_3, a_4), a_5, \dots, a_{m-1}, a_k)\end{aligned}$$

and so on. Notations  $\tilde{\tau}^{(r)}(\mathbf{a}, [\mathbf{x}], \mathbf{b})$  and  $\tilde{\nu}^{(r)}(\mathbf{a}, [\mathbf{x}], \mathbf{b})$  are defined in the same way as defined in (4.13).

Postmultiplying with both sides of equation (4.14) by  $W_1^{(m-1)}$ , we can derive the following formulation

$$\left( \tau^{(m-1)}(a_1), \tau^{(m-1)}(a_2), \dots, \tau^{(m-1)}(a_{m-1}) \right) = \mathbf{v}^{(m-1)}(\mathbf{a}_{m-1}) \left( E_{(m-1)} - U^{(m-1)}(\mathbf{a}_{m-1}) \right)^{-1} \quad (4.18)$$

$$\text{where } U^{(m-1)}(\mathbf{a}_{m-1}) = \begin{pmatrix} p_{a_1 a_1} & p_{a_1 a_2} & \cdots & p_{a_1 a_{m-1}} \\ p_{a_2 a_1} & p_{a_2 a_2} & \cdots & p_{a_2 a_{m-1}} \\ \vdots & \vdots & \ddots & \vdots \\ p_{a_{m-1} a_1} & p_{a_{m-1} a_2} & \cdots & p_{a_{m-1} a_{m-1}} \end{pmatrix},$$

$$\text{and } \mathbf{v}^{(m-1)}(\mathbf{a}_{m-1}) = \left( \|\tilde{\pi}([\mathbf{a}_{m-1} \ominus a_s], a_k) R_{is}^{(m-1)}(\mathbf{a}_{m-1})\| \right)_{s=1}^{m-1}.$$

Postmultiplying  $W_2^{(m-1)}$  with both sides of equation (4.14), we can derive the relation

$$\tau^{(m-1)}(a_1, a_2) = \frac{p_{a_2 a_1}}{1 - p_{a_1 a_1}} \{ \tau^{(m-1)}(a_2) + \nu^{(m-1)}(a_1, a_2) \}. \quad (4.19)$$

Thus we can calculate  $\tau^{(m-1)}(a_1, a_2)$  by using the  $\tau^{(m-1)}(a_1)$  obtained in (4.18) and  $\nu^{(m-1)}(a_1, a_2)$  which can be calculated using information given by the invariant vector of transition matrix  $Q_M^{(m-1)}$ . Postmultiplying  $W_3^{(m-1)}$  with both sides of equation (4.14), we have

$$\tilde{\tau}^{(m-1)}([a_1, a_2], a_3) = \left( \tilde{\varphi}^{(m-1)}(a_s, [(a_1, a_2) \ominus a_s], a_3) R_{is}^{(2)}(a_1, a_2) \right)_{s=1}^2 \left( E_2 - U^{(2)}(a_1, a_2) \right)^{-1}. \quad (4.20)$$

Since

$$\begin{aligned}\varphi^{(m-1)}(a_1, (a_1, a_2) \ominus a_1, a_3) &= \varphi^{(m-1)}(a_1, a_2, a_3), \\ \varphi^{(m-1)}(a_1, a_2, a_3) &= \tau^{(m-1)}(a_1, a_2) + \nu^{(m-1)}(a_1, a_2, a_3), \\ \text{and } \nu^{(m-1)}(a_1, a_2, a_3) &= \pi(a_2, a_3, (a_4, \dots, a_{m-1}), a_k),\end{aligned}$$

we can calculate  $\tau^{(m-1)}(a_1, a_2, a_3)$  by using the  $\tau^{(m-1)}(a_1, a_2)$  obtained in (4.19) and  $\pi(a_2, a_3, (a_4, \dots, a_{m-1}), a_k)$  which can be calculated using information given by the invariant vector of transition matrix  $Q_M^{(m-1)}$ .

Similarly, postmultiplying  $W_4^{(m-1)}$  with both sides of equation (4.14), we have

$$\tilde{\tau}^{(m-1)}([a_1, a_2, a_3], a_4) = \left( \tilde{\varphi}^{(m-1)}(a_s, [(a_1, a_2, a_3) \ominus a_s], a_4) R_{is}^{(3)}(a_1, a_2, a_3) \right)_{s=1}^3 \left( E_3! - Q_{ii}^{(3)} \right)^{-1}. \quad (4.21)$$

We have to deal with  $3! \times 3!$  dimensional matrix  $Q_{ii}^{(3)}$ , if we calculate  $\tau^{(m-1)}(a_1, a_2, a_3, a_4)$  directly through equation (4.21). Postmultiplying  $W_1^{(3)} (= E_3 \otimes \mathbf{e}_2^t)$  and  $W_2^{(3)} (= E_6 \otimes \mathbf{e}_1^t)$  with both sides of equation (4.21), we can obtain alternative forms

$$(\tau_1^{(3)}(a_1), \tau_1^{(3)}(a_2), \tau_1^{(3)}(a_3)) = \left( \|\tilde{\varphi}(a_s, [\mathbf{a}_3 \ominus a_s], a_k) R_{is}^{(3)}(\mathbf{a}_3)\| \right)_{s=1}^3 \left( E_3 - U^{(3)}(\mathbf{a}_3) \right)^{-1} \quad (4.22)$$

$$\text{and} \quad \tau^{(m-1)}(a_1, a_2, a_3) = \frac{p_{a_2 a_1}}{1 - p_{a_1 a_1}} \{ \tau_1^{(3)}(a_2) + \varphi^{(m-1)}(a_1, a_2, a_3, a_4) \} \quad (4.23)$$

where

$$\begin{aligned} \tau_1^{(3)}(a_1) &= \pi(a_1, (a_2, a_3), a_4, (a_5, \dots, a_{m-1}), a_k), \\ \tau_1^{(3)}(a_2) &= \pi(a_2, (a_1, a_3), a_4, (a_5, \dots, a_{m-1}), a_k), \\ \text{and} \quad \tau_1^{(3)}(a_3) &= \pi(a_3, (a_1, a_2), a_4, (a_5, \dots, a_{m-1}), a_k). \end{aligned}$$

Thus we can calculate  $\tau^{(m-1)}(a_1, a_2, a_3)$  through alternative equations (4.22) and (4.23) dealing with  $3 \times 3$  matrix  $U^{(3)}(\mathbf{a}_3)$  instead of matrix  $Q_{ii}^{(3)}$  in equation (4.21).

For general cases, by postmultiplying  $W_n^{(m-1)}$  with both sides of equation (4.14), we can derive the following form for  $n = 3, 4, \dots, m-2$ .

$$\begin{aligned} \tilde{\tau}^{(m-1)}([\mathbf{a}_{n-1}], a_n) &= \left( \tilde{\varphi}^{(m-1)}(a_s, [\mathbf{a}_{n-1} \ominus a_s], a_n) R_{is}^{(n-1)}(\mathbf{a}_{n-1}) \right)_{s=1}^{n-1} \\ &\quad \times \left( E_{(n-1)!} - Q_{ii}^{(n-1)} \right)^{-1}. \end{aligned} \quad (4.24)$$

We can see that both forms (4.17) and (4.24) have a very similar structure. Using this similarity, postmultiplying  $W_s^{(n)}$  with both sides of equation (4.24) for  $s = 1, 2, \dots, n-2$ , we can derive another set of equations each of which is again similar to original equation (4.24) which has a smaller matrix than that of original matrix  $Q_{ii}^{(n-1)}$ .

Applying the procedure repeatedly, we can finally obtain the form  $\tau_{m-2}^{(m-1)}(a_1, \dots, a_{m-2})$  which is equal to element  $\pi(a_1, \dots, a_{m-1}, a_k)$  of invariant vector of transition matrix  $Q_M^{(m)}$ . Thus we can calculate each element of the invariant vector of transition matrix  $Q_M^{(m)}$  given that of  $Q_M^{(m-1)}$ .

## 5. Computational algorithm

The calculation algorithm for the invariant vector of transition matrix  $Q_M^{(m)}$  given the invariant vector of transition matrix  $Q_M^{(m-1)}$  is described in the following. We define the index set

$$J_s^{(r)} = \{(i_1, i_2, \dots, i_s) | i_1 \in I_r, i_2 \in I_r \setminus \{a_1\}, \dots, i_s \in I_r \setminus \{i_1, \dots, i_{s-1}\}\} \quad \text{for } r \leq s.$$

The main procedure to calculate the invariant vector of the transition matrix for a  $m$ -length stack LRU is **{ begin : Call Calculation[m] : end }**. The main procedure **Calculation[m]** calls **Procedure[m-1]** which is recursive, i.e., **Procedure[m-1]** calls

**Procedure[r-1]** for  $r = 2, 3, \dots, m-3$ . (See Appendix B for computation sequence of the algorithm.) The main procedure **Calculation[m]** and the recursive procedure **Procedure[r-1]** are defined as follows.

**Calculation[m]: Input** :  $\tilde{\pi}([1, 2, \dots, m-1])$     **Output** :  $\tilde{\pi}([1, 2, \dots, m])$

**begin** : For  $a_m = 1$  to  $M$   $\{(1)$     For  $(i_1, i_2, \dots, i_{m-1}) \in J_{m-1}^{(M)} \{(2)$   
 $\mathbf{a}_m \leftarrow (a_{i_1}, a_{i_2}, \dots, a_{i_{m-1}}, a_m)$   
 For  $i = 1, 2, \dots, m-1 \{(3)$     For  $\sigma[\mathbf{a}_{m-1}] \{(4)$

$$\tilde{\varphi}^{m-1}([\mathbf{a}_{m-1} \ominus a_i], a_m) \leftarrow \tilde{\pi}([\mathbf{a}_{m-1} \ominus a_i], a_m) \quad \{(4) \} \{(3)$$

Call **Procedure[m-1]**

$$\text{For } \sigma[\mathbf{a}_{m-1}] \quad \{(5) \} \tilde{\pi}([\mathbf{a}_{m-1}], a_m) \leftarrow \tilde{\tau}_m([\mathbf{a}_{m-1}], a_m) \{(5) \} \{(2) \} \{(1)$$

: **end**.

**Procedure[1]: Input** :  $\mathbf{a}_2 = (a_1, a_2)$  and  $\varphi^{(1)}(a_2)$ .    **Output** :  $\tau_2(a_1, a_2)$ .

$$\text{begin :} \quad \tau_2(a_1, a_2) \leftarrow \frac{p_{a_2 a_1}}{1 - p_{a_1 a_1}} \varphi^{(1)}(a_2) \quad \text{: end.}$$

**Procedure[2]:**

**Input** :  $\mathbf{a}_3 = (a_1, a_2, a_3)$  and  $\tilde{\varphi}^{(2)}([\mathbf{a}_2 \ominus a_i], a_3)$  for  $i = 1, 2$ . ( $\mathbf{a}_2 = (a_1, a_2)$ .)

**Output** :  $\tilde{\tau}_3([a_1, a_2], a_3)$

**begin** :

$$\tilde{\tau}_3([a_1, a_2], a_3) \leftarrow \left( \|\tilde{\varphi}^{(2)}([\mathbf{a}_2 \ominus a_i], a_3) R_i^{(2)}(\mathbf{a}_2)\| \right)_{i=1}^2 \left( E_2 - U^{(2)}(\mathbf{a}_2) \right)^{-1}$$

: **end**.

**Procedure[r-1]:**

**Input** :  $\mathbf{a}_r = (a_1, a_2, \dots, a_r)$  ;  $\tilde{\varphi}^{(r-1)}([\mathbf{a}_{r-1} \ominus a_i], a_r)$  for  $\sigma[\mathbf{a}_{r-1}]$  and  $i = 1, 2, \dots, r-1$ .

**Output** :  $\tilde{\tau}_r([a_1, a_2, \dots, a_{r-1}], a_r)$ .

**begin** :

$$\tilde{\tau}_1(a_i)_{i=1}^{r-1} \leftarrow \left( \|\tilde{\varphi}^{(r-1)}([\mathbf{a}_{r-1} \ominus a_i], a_r) R_i^{(r-1)}(\mathbf{a}_{r-1})\| \right)_{i=1}^{r-1} \left( E_{r-1} - U^{(r-1)}(\mathbf{a}_{r-1}) \right)^{-1}$$

For  $s = 2$  to  $r-3 \{(1)$     For  $(i_1, i_2, \dots, i_{r-1}) \in J_s^{(r-1)} \{(2)$

$$\mathbf{a}_s \leftarrow (a_{i_1}, a_{i_2}, \dots, a_{i_{s-1}}, a_r)$$

$$\begin{aligned} \varphi^{s-1}(a_2, a_3, \dots, a_s) &\leftarrow \tau_{s-1}(a_2, a_3, \dots, a_{i_s}) \\ &+ \varphi^{(r-1)}(a_2, a_3, \dots, a_s, (\mathbf{a}_{r-1} \ominus \{a_1, a_2, \dots, a_s\}), a_r) \end{aligned}$$

Call **Procedure[s-1] \{(2) \} \{(1)**

: **end**.

### 6. Examples

Two examples for stack distance distribution are shown in this section. We will hereafter assume that the number of pages equals seven ( $M = 7$ ) for those two examples, and assume the transition matrix  $P$  have the property  $\sum_{i=1}^7 p_{ij} = 1, \forall j \in I_7$ , so that the steady state distribution in the original page reference, i.e. the invariant vector of  $P$ , can be given by

$$\pi(1) = \pi(2) = \dots = \pi(7) = \frac{1}{7}.$$

Since each page is referred with the same probability in the original page reference, if an independent-page-reference is assumed, then the stack distance distributions for both of the examples are the same as the distribution

$$P(\delta = 1) = P(\delta = 2) = \dots = P(\delta = 7) = \frac{1}{7}.$$

The two examples, however, will show that the stack distance distributions are much different from those of the independent-page-reference, if we assume the Markovian-page-reference. Those results suggest that an independent-page-reference model is insufficient for evaluating the LRU caching strategy. The first example has “loop” structure and the second example has “self reference” structure in each Markov chain for the page-reference. We developed a set of Mathematica programs (about 500 lines) which calculated the stack distance distributions for the examples based on the algorithm described in the Section 5.

**Example 6.1 (Loop)** Let  $P$  be the transition matrix described below. This Markov chain has “loops” in its state transition.

$$P = \begin{bmatrix} \varepsilon & 1 - \varepsilon & 0 & 0 & 0 & 0 & 0 \\ 0 & \varepsilon & 1 - \varepsilon & 0 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon & 1 - \varepsilon & 0 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon & 1 - \varepsilon & 0 & 0 \\ 0 & 0 & 0 & 0 & \varepsilon & 1 - \varepsilon & 0 \\ \frac{-\varepsilon}{2} + p & 0 & 0 & 0 & 0 & \varepsilon & 1 - \frac{\varepsilon}{2} - p \\ 1 - \frac{\varepsilon}{2} - p & 0 & 0 & 0 & 0 & 0 & \frac{\varepsilon}{2} + p \end{bmatrix}. \tag{6.1}$$

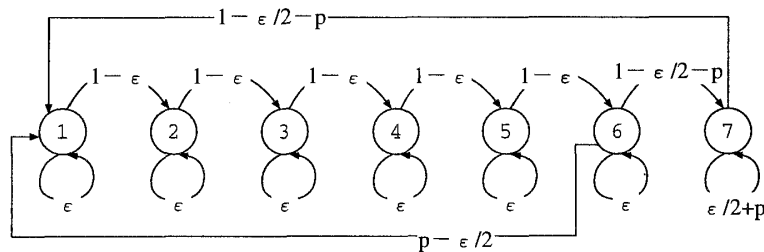


Figure 1: Markov chain containing loop structures

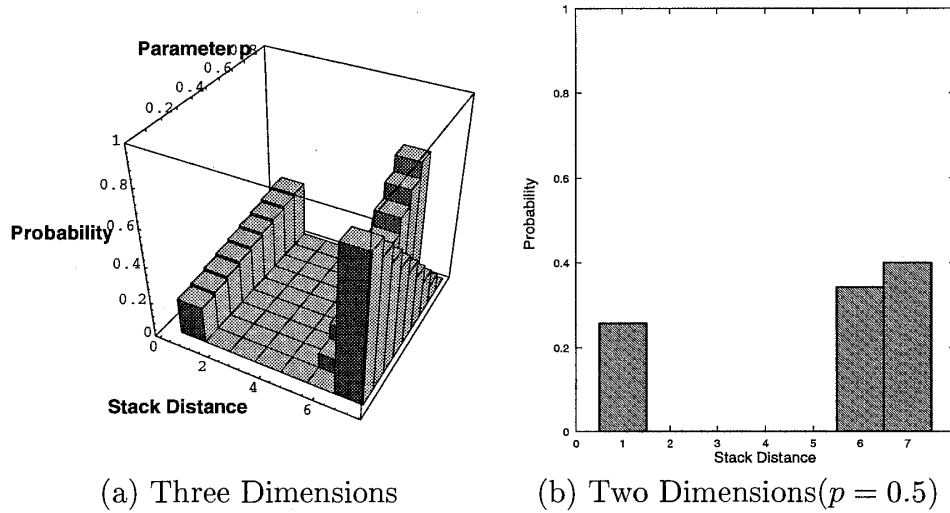
In this example, the Mathematica programs can produce not only numerical result but also exact solution for the the stack distance distribution  $P(\delta)$  as in the form

$$P(\delta = 1) = \frac{13}{14}\varepsilon + \frac{1}{7}p, \quad P(\delta = 2) = P(\delta = 3) = \dots = P(\delta = 5) = 0,$$



Table 1: Numerical Result for  $\varepsilon = 0.2$  in Example 6.1

$p$	Stack Distance ( $\delta$ )			
	1	2,3,4,5	6	7
0.1	0.20	0	0.00	0.80
0.3	0.23	0	0.17	0.60
0.5	0.26	0	0.34	0.40
0.7	0.29	0	0.51	0.20
0.9	0.31	0	0.69	0.00

Figure 2: Stack Distance Distribution for  $\varepsilon = 0.2$  in Example 6.1.

$$P(\delta = 6) = \frac{3}{7}(2p - \varepsilon), \quad P(\delta = 7) = \frac{1}{2}(2 - \varepsilon - 2p).$$

It took about 30 minutes CPU time for obtaining the result on a MIPS R4000 based UNIX workstation. Numerical results, given  $\varepsilon = 0.2$ , are shown in Table 1, and Fig. 2.

**Example 6.2 (Self Reference)** We can construct a transition matrix  $P$  in which the probabilities of “self reference” become greater than the others, by using the following procedure.

1. Let  $q_i = p(1 - p)^i$  (geometric distribution with parameter  $p$ ) for  $i = 0, 1, 2, 3$ .
2. Define that

$$p_i = \frac{q_i}{q_0 + 2(q_1 + q_2 + q_3)}.$$

Then the equation  $p_3 + p_2 + p_1 + p_0 + p_1 + p_2 + p_3 = 1.0$  holds.

3. Construct the transition matrix as follows:

$$P = \begin{bmatrix} p_0 + p_1 & p_1 + p_2 & p_2 + p_3 & p_3 & 0 & 0 & 0 \\ p_1 + p_2 & p_0 + p_3 & p_1 & p_2 & p_3 & 0 & 0 \\ p_2 + p_3 & p_1 & p_0 & p_1 & p_2 & p_3 & 0 \\ p_3 & p_2 & p_1 & p_0 & p_1 & p_2 & p_3 \\ 0 & p_3 & p_2 & p_1 & p_0 & p_1 & p_2 + p_3 \\ 0 & 0 & p_3 & p_2 & p_1 & p_0 + p_3 & p_1 + p_2 \\ 0 & 0 & 0 & p_3 & p_2 + p_3 & p_1 + p_2 & p_0 + p_1 \end{bmatrix} \quad (6.2)$$

Table 2: Numerical Result of Stack Distance Distribution for Example 6.2.

$p$	Stack Distance( $\delta$ )						
	1	2	3	4	5	6	7
0.1	0.25	0.20	0.14	0.11	0.094	0.095	0.11
0.3	0.32	0.19	0.14	0.10	0.081	0.078	0.091
0.5	0.43	0.18	0.12	0.085	0.063	0.056	0.068
0.7	0.60	0.15	0.088	0.057	0.039	0.031	0.037
0.9	0.84	0.079	0.033	0.018	0.012	0.0081	0.0078

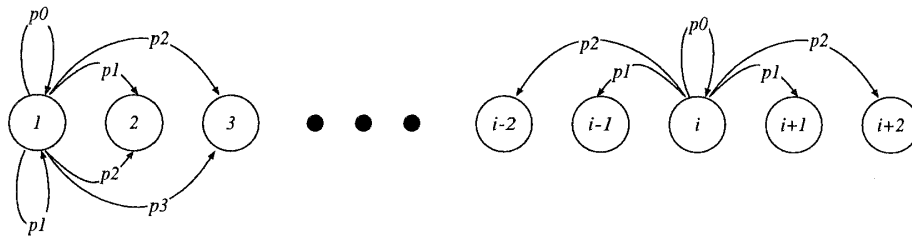
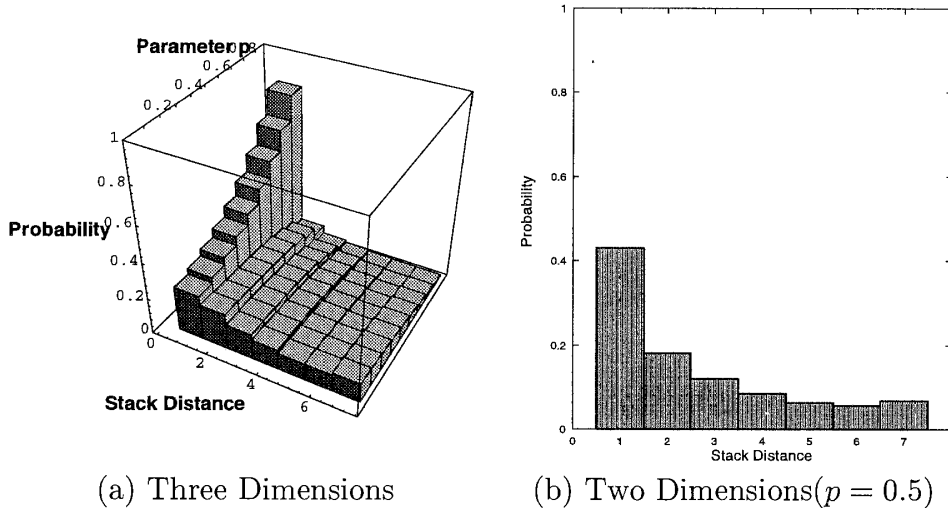


Figure 3: Markov chain with "self reference" characteristics



(a) Three Dimensions

(b) Two Dimensions( $p = 0.5$ )

Figure 4: Stack Distance Distribution for Example 6.2.

The stack distance distribution calculated with the Mathematica programs are shown in Table 2 and Fig. 4. It took about 20 minutes CPU time for calculating the stack distance distribution for each  $p$ .

### 7. Evaluation of the efficiency

We evaluate approximately the computational efficiency of the proposed method comparing with the method derived by Chu and Knott [7]. We refer the proposed method to the lumping method and the method of Chu and Knott to regular-expression method, respectively.

Since both of them have the same space complexity, i.e., they require at most  $M \times M$  size of matrix operations, we consider just the computational complexity.

In both methods, the matrix inversion form of  $(E_n - A_n)^{-1}$ ,  $n = 1, 2, \dots, M$ , appear as fundamental operations in their calculation, where  $A_n$  is a  $n$ -dimensional matrix. Hence, we estimate the frequency of appearance of the matrix inversion form in both of the methods and compare the computational complexity, and we use frequency of multiplications as a measure of the computational complexity. Based on the observation that a number  $n^3$  of multiplications is required to solve an  $n$ -dimensional linear equation, we assume that the computational complexity to calculate  $n$ -dimensional matrix inversion form of  $(E_n - A_n)^{-1}$  requires a number of multiplications in the form  $c_n = c \cdot n^3$ , where  $c$  is a proportional constant. We write  $C_L(M)$  and  $C_R(M)$  to describe total computational complexity for the lumping method and regular-expression method, respectively.

**Lumping method:** Let  $D_{m-1}$  be the computational complexity to calculate the invariant vector  $\pi(a_1, a_2, \dots, a_m)$  given the invariant vector  $\pi(a_1, a_2, \dots, a_{m-1})$  for all of feasible states by using equation (4.4). By applying the partial lump method and the block-diagonal decomposition, we can decompose the equation (4.4) to a number  $M - m + 1$  of equations that have the common coefficient matrix  $(E_{(m-1)!} - Q_{11}^{(m-1)})^{-1}$  as shown in the form (4.15). We write  $K_{m-1}$  to describe the computational complexity to solve the equation (4.15). Since a parameter  $a_k$  in equation (4.15) may take a number  $M - m + 1$  values and a number of possible sequences of the other parameters  $(a_1, a_2, \dots, a_{m-1})$  is the combinations of  $M$  taken  $m - 1$  at a time, we have the relation

$$D_{m-1} = (M - m + 1) \binom{M}{m-1} K_{m-1} = m \binom{M}{m} K_{m-1}, \quad m \geq 2. \quad (7.1)$$

Note that  $K_1 = c_1$  and  $K_2 = c_2$ . We consider the computational complexity to solve the equation (4.18) and the equation (4.24), which are derived by multiplying  $W_1^{(m-1)}$  and  $W_n^{(m-1)}$  with both sides of the equation (4.14), respectively. The computational complexity to solve the equation (4.24) is  $c_{m-1}$  clearly. A parameter  $a_n$  in the equation (4.24) may take  $m - 1$  values and a number of possible sequences of the other parameters  $(a_1, a_2, \dots, a_{n-1})$  is the combinations of  $m - 2$  taken  $n - 1$  at a time. Since the computational complexity to solve the equation (4.24) for a set of fixed parameters  $(a_1, a_2, \dots, a_n)$  is equivalent to  $K_{n-1}$ , the computational complexity to solve the equation (4.24) for all of feasible set of parameters is

$$(m-1) \binom{m-1}{n-1} K_{n-1} = n \binom{m-1}{n} K_{n-1} \quad (7.2)$$

for  $n = 2, 3, \dots, m - 2$ . Thus, for  $m \geq 4$ , we have the relation

$$K_{m-1} = c_{m-1} + \sum_{n=2}^{m-2} n \binom{m-1}{n} K_{n-1}. \quad (7.3)$$

The computational complexity to calculate the initial values  $(\pi(1), \pi(2), \dots, \pi(M))$  is  $c_M$ . Therefore we have the relation

$$C_L(M) = c_M + \sum_{m=2}^M D_{m-1}$$

$$= c_M + \sum_{m=2}^M m \binom{M}{m} c_{m-1} + \sum_{m=4}^M \sum_{n=2}^{m-2} m n \binom{M}{m} \binom{m-1}{n} K_{n-1}. \quad (7.4)$$

**Regular-expression method:** The algorithm requires  $n$ -dimensional matrix inversion operation for  $n = 1, 2, \dots, M$  for a set of fixed parameters  $(a_1, a_2, \dots, a_M)$  in addition to the calculation of initial values, and the order of feasible set of parameters is the permutations of  $M$  taken  $M$  at a time. In this case, we can estimate the frequency of multiplications explicitly in the form

$$\begin{aligned} C_R(M) &= c_M + M! \sum_{n=1}^M c_n \\ &= c \left\{ M^3 + \frac{1}{4} M! M^2 (M+1)^2 \right\}. \end{aligned} \quad (7.5)$$

**Comparison of the efficiency:** Table 3 shows the computational complexity  $C_L(M)$ ,  $C_R(M)$  and the ratio  $C_R(M)/C_L(M)$  for  $M = 2, 4, \dots, 16$ , respectively.

Although real efficiency of the algorithms depends on not only frequency of multiplications but also data structure and the other operations, as shown in Table 3, the lumping method shows better efficiency than that of the regular-expression method. The amount of  $C_L(M)$ , however, increases rapidly as the number  $M$  of pages increase, so that computation for a large number of pages may become intractable. For example, if we assume that one operation of multiplication requires 10 ns ( $10 \times 10^{-9}$  second) and  $c = 1$ , then it takes more than one year to compute the distribution for the case  $M = 16$ . Thus, from a practical view point, one can say that evaluation of the efficiency for both of the methods does not make sense any more in those cases that  $M$  is greater than 16.

$M$	$C_L(M)$	$C_R(M)$	$C_R(M)/C_L(M)$
2	$1.0 \times 10$	$2.6 \times 10$	2.6
4	$3.0 \times 10^2$	$2.4 \times 10^3$	8.1
6	$1.4 \times 10^4$	$3.1 \times 10^5$	22.4
8	$1.2 \times 10^6$	$5.2 \times 10^7$	42.9
10	$1.6 \times 10^8$	$1.0 \times 10^{10}$	65.1
12	$3.4 \times 10^{10}$	$2.9 \times 10^{12}$	85.1
14	$9.5 \times 10^{12}$	$9.6 \times 10^{14}$	100.4
16	$3.5 \times 10^{15}$	$3.8 \times 10^{17}$	109.5

Table 3: Frequency of multiplications ( $c = 1$ ).

## 8. Summary

We presented the lumpability of the Markov chain for an full-length LRU stack. We also developed a new computation method that is based on the characteristics of the Markov chain. With the help of this method, we showed some numerical examples of LRU stack distance distributions. These results reconfirm that assuming independent page references is much simpler in analyzing practical LRU caching algorithms than assuming Markovian page references. We are not likely to find the same characteristics of lumpability in other page replacement algorithms, such as FIFO and LFU (Least-Frequently-Used). We thus need alternative methods of analysis in these cases.

### Acknowledgment

The authors express their sincere thanks to three anonymous referees whose comments were valuable in revising the original manuscript.

## Appendix

### A. Example of lumpability

We consider the case where  $M = 4$  and  $m = 2$ . We have  $4! = 24$  states for the Markov chain with transition matrix  $Q_4$ . The state set  $S_4$  of the Markov chain is described by

$$S_4 = \{s(1, 2, 3, 4), s(1, 2, 4, 3), s(1, 3, 2, 4), s(1, 3, 4, 2), \dots, s(4, 3, 2, 1)\}.$$

We can write the decomposition form (3.3) of permutation group  $G_4$  in the form

$$G_4 = \bigcup_{i=1}^6 \bigcup_{n=1}^2 \bigcup_{j=1}^2 h_j^{(2)} k_n^{(2)} g_i^{(2)}. \quad (\text{A.1})$$

Note that  $i^* = 6$ ,  $n^* = 2$ ,  $j^* = 2$ ,  $\mathbf{u} = (1, 2, 3, 4)$ ,

$$H^{(2)} = \{h_1^{(2)}, h_2^{(2)}\}, h_1^{(2)}(\mathbf{u}) = (1, 2, 3, 4), h_2^{(2)}(\mathbf{u}) = (2, 1, 3, 4)$$

$$K^{(2)} = \{k_1^{(2)}, k_2^{(2)}\}, k_1^{(2)}(\mathbf{u}) = (1, 2, 3, 4), k_2^{(2)}(\mathbf{u}) = (1, 2, 4, 3)$$

$$G^{(2)} = \{g_1^{(2)}, g_2^{(2)}, g_3^{(2)}, g_4^{(2)}, g_5^{(2)}, g_6^{(2)}\},$$

$$g_1^{(2)}(\mathbf{u}) = (1, 2, 3, 4), g_2^{(2)}(\mathbf{u}) = (1, 3, 2, 4), g_3^{(2)}(\mathbf{u}) = (1, 4, 2, 3), g_4^{(2)}(\mathbf{u}) = (2, 3, 1, 4),$$

$$g_5^{(2)}(\mathbf{u}) = (2, 4, 1, 3) \quad \text{and} \quad g_6^{(2)}(\mathbf{u}) = (3, 4, 1, 2).$$

According to the decomposition form (A.1), we can partition  $S_4$  such that

$$\begin{aligned} S_4 &= \{A_1^{(2)}, A_2^{(2)}, A_3^{(2)}, A_4^{(2)}, A_5^{(2)}, A_6^{(2)}\} \\ &= \{ \{A_{11}^{(2)}, A_{12}^{(2)}\}, \{A_{21}^{(2)}, A_{22}^{(2)}\}, \dots, \{A_{61}^{(2)}, A_{62}^{(2)}\} \} \end{aligned} \quad (\text{A.2})$$

where  $A_{ij}^{(2)} = \{s(\mathbf{a}_n) \mid \mathbf{a}_n = (h_j^{(2)} k_n^{(2)} g_i^{(2)})(\mathbf{u}), n = 1, 2\}$  for  $i = 1, 2, \dots, 6$ ,  $j = 1, 2$

that is

$$A_{11}^{(2)} = \{s((h_1^{(2)} k_1^{(2)} g_1^{(2)})(\mathbf{u})), s((h_1^{(2)} k_2^{(2)} g_1^{(2)})(\mathbf{u}))\} = \{s(1, 2, 3, 4), s(1, 2, 4, 3)\}$$

$$A_{12}^{(2)} = \{s((h_2^{(2)} k_1^{(2)} g_1^{(2)})(\mathbf{u})), s((h_2^{(2)} k_2^{(2)} g_1^{(2)})(\mathbf{u}))\} = \{s(2, 1, 3, 4), s(2, 1, 4, 3)\}$$

$$A_{21}^{(2)} = \{s((h_1^{(2)} k_1^{(2)} g_2^{(2)})(\mathbf{u})), s((h_1^{(2)} k_2^{(2)} g_2^{(2)})(\mathbf{u}))\} = \{s(1, 3, 2, 4), s(1, 3, 4, 2)\}$$

$$A_{22}^{(2)} = \{s((h_2^{(2)} k_1^{(2)} g_2^{(2)})(\mathbf{u})), s((h_2^{(2)} k_2^{(2)} g_2^{(2)})(\mathbf{u}))\} = \{s(3, 1, 2, 4), s(3, 1, 4, 2)\}$$

$$A_{31}^{(2)} = \{s((h_1^{(2)} k_1^{(2)} g_3^{(2)})(\mathbf{u})), s((h_1^{(2)} k_2^{(2)} g_3^{(2)})(\mathbf{u}))\} = \{s(1, 4, 2, 3), s(1, 4, 3, 2)\}$$

$$A_{32}^{(2)} = \{s((h_2^{(2)} k_1^{(2)} g_3^{(2)})(\mathbf{u})), s((h_2^{(2)} k_2^{(2)} g_3^{(2)})(\mathbf{u}))\} = \{s(4, 1, 2, 3), s(4, 1, 3, 2)\}$$

...

$$A_{61}^{(2)} = \{s((h_1^{(2)} k_1^{(2)} g_6^{(2)})(\mathbf{u})), s((h_1^{(2)} k_2^{(2)} g_6^{(2)})(\mathbf{u}))\} = \{s(3, 4, 1, 2), s(3, 4, 2, 1)\}$$

$$A_{62}^{(2)} = \{s((h_2^{(2)} k_1^{(2)} g_6^{(2)})(\mathbf{u})), s((h_2^{(2)} k_2^{(2)} g_6^{(2)})(\mathbf{u}))\} = \{s(4, 3, 1, 2), s(4, 3, 2, 2)\}$$

Based on the partition (A.2), we can describe the Markov chain  $Q_4$  in the form





$$Q_4^{(1)} = \begin{matrix} s(1) \\ s(2) \\ s(3) \\ s(4) \end{matrix} \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{pmatrix}.$$

**Partial lump in the lumped chain:** The transition matrix  $Q_4^{<2>}$  is lumpable with respect to the partition

$$S_{[2,3,4]}^{<2>} = \{S_1^{<2>}, S_2^{(2)}, S_3^{(2)}, S_4^{(2)}\} = \{\{s(1, 2), s(1, 3), s(1, 4)\}, s(2), s(3), s(4)\}$$

(Theorem 3.3) and the transition matrix of the lumped process is written in the form

$$Q_4^{[2,3,4]} = \begin{matrix} s(1, 2) \\ s(1, 3) \\ s(1, 4) \\ s(2) \\ s(3) \\ s(4) \end{matrix} \left( \begin{array}{c|ccc} p_{11} & p_{12} & p_{13} & p_{14} \\ & p_{11} & p_{12} & p_{13} & p_{14} \\ & & p_{11} & p_{12} & p_{13} & p_{14} \\ \hline p_{21} & p_{22} & p_{23} & p_{24} \\ & p_{31} & p_{32} & p_{33} & p_{34} \\ & & p_{41} & p_{42} & p_{43} & p_{44} \end{array} \right). \quad (\text{A.4})$$

**Partial lump method:** Applying notations in the matrix form (4.1) to the matrix (A.4), we have matrices

$$\Phi_{11}^{(2)} = \begin{pmatrix} p_{11} & & \\ & p_{11} & \\ & & p_{11} \end{pmatrix} (= Q_{11}^{(1)} \otimes E_3) \quad \text{and} \quad \Phi_{[1]1}^{(1)} = \begin{pmatrix} p_{21} & & \\ & p_{31} & \\ & & p_{41} \end{pmatrix}.$$

With the notations in the equation (4.4), we can derive equation

$$\left( \pi(1, 2), \pi(1, 3), \pi(1, 4) \right) = \left( \pi(2), \pi(3), \pi(4) \right) \Phi_{[1]1}^{(1)} \left( E_3 - \Phi_{11}^{(2)} \right)^{-1}. \quad (\text{A.5})$$

Solving the equation (A.5), we can obtain  $(\pi(1, 2), \pi(1, 3), \pi(1, 4))$ , a part of the invariant vector for matrix  $Q_4^{(2)}$ , given  $(\pi(2), \pi(3), \pi(4))$ , and a part of the invariant vector for matrix  $Q_4^{(1)}$ . Applying the procedure to the other partitions, we can derive the full-length invariant vector for matrix  $Q_4^{(2)}$ .

**Block-diagonal decomposition:** We consider another example. Suppose that  $M = 6$  and  $m = 4$ . The matrix  $\Phi_{11}^{(3)}$  can be written in the form

$$\Phi_{11}^{(4)} = Q_{11}^{(3)} \otimes E_3 \quad \text{where} \quad Q_{11}^{(3)} = \begin{matrix} s(1, 2, 3) \\ s(1, 3, 2) \\ s(2, 1, 3) \\ s(2, 3, 1) \\ s(3, 1, 2) \\ s(3, 2, 1) \end{matrix} \left( \begin{array}{c|cc} p_{11} & p_{12} & p_{13} \\ & p_{11} & p_{12} & p_{13} \\ \hline p_{21} & p_{22} & p_{23} \\ p_{21} & p_{22} & p_{23} \\ \hline p_{31} & p_{32} & p_{33} \\ & p_{31} & p_{32} & p_{33} \end{array} \right).$$



Changing the order of arrangement for states according to (4.9) and (4.11),  $\Phi_{11}^{(4)}$  and  $\Phi_{[1]1}^{(3)}$  can be reformed such that

$$\begin{aligned} \Phi_{11}^{(4)*} &= \text{block-diagonal}\{\mathcal{Q}_{11}^{(3)}, \mathcal{Q}_{11}^{(3)}, \mathcal{Q}_{11}^{(3)}\} \\ \text{and } \Phi_{[1]1}^{(3)*} &= \text{block-diagonal}\{R_1^{(3)}(1, 2, 3), R_1^{(3)}(1, 2, 3), R_1^{(3)}(1, 2, 3)\} \end{aligned}$$

respectively, where  $R_1^{(3)}(1, 2, 3) = \text{block-diagonal}\{R_{11}^{(3)}(1, 2, 3), R_{12}^{(3)}(1, 2, 3)\}, R_{13}^{(3)}(1, 2, 3)\}$

$$R_{11}^{(3)}(1, 2, 3) = \begin{pmatrix} p_{21} \\ p_{31} \end{pmatrix}, R_{12}^{(3)}(1, 2, 3) = \begin{pmatrix} p_{12} \\ p_{32} \end{pmatrix} \quad \text{and} \quad R_{13}^{(3)}(1, 2, 3) = \begin{pmatrix} p_{13} \\ p_{23} \end{pmatrix}.$$

Therefore the equation (4.17) takes the following form in this case.

$$\tilde{\pi}([1, 2, 3], a_k) = \left( \tilde{\pi}([(1, 2, 3) \ominus s], a_k) R_{1s}^{(3)}(1, 2, 3) \right)_{s=1}^3 \times \left( E_{3!} - \mathcal{Q}_{11}^{(3)} \right)^{-1} \quad (\text{A.6})$$

where  $a_k$  may be 4, 5 or 6.

Note that  $\tilde{\pi}([1, 2, 3], a_k) = (\pi(1, 2, 3, a_k), \pi(1, 3, 2, a_k), \dots, \pi(3, 2, 1, a_k))$ ,  $\pi([(1, 2, 3) \ominus 1], a_k) = \tilde{\pi}([2, 3], a_k) = (\pi(2, 3, a_k), \pi(3, 2, a_k))$  and so on.

**Recursive reduction:** Multiplying  $W_1^{(3)} (= E_3 \otimes e_2^t)$  by both sides of the equation (A.6), we derive the following relation.

$$\left( \pi(1, (2, 3), a_k), \pi(2, (1, 3), a_k), \pi(3, (1, 2), a_k) \right) = (v_1^{(3)}, v_2^{(3)}, v_3^{(3)}) \left( E_3 - U^{(3)}(1, 2, 3) \right)^{-1} \quad (\text{A.7})$$

where

$$U^{(3)}(1, 2, 3) = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix}$$

$v_1^{(3)} = \|\tilde{\pi}([(1, 2, 3) \ominus 1], a_k) R_{11}^{(3)}(1, 2, 3)\| = p_{21}\pi(2, 3, a_k) + p_{31}\pi(3, 2, a_k)$ ,  $\pi(1, (2, 3), a_k) = \pi(1, 2, 3, a_k) + \pi(1, 3, 2, a_k)$  and so on. Multiplying  $W_2^{(3)} (= E_6 \otimes e_1^t)$  by both sides of equation (A.6), we derive the following relation.

$$\pi(1, 2, 3, a_k) = \frac{p_{21}}{1 - p_{11}} \{ \pi(2, (1, 3), a_k) + \pi(2, 3, a_k) \}. \quad (\text{A.8})$$

Note that  $\tau^{(3)}(1) = \pi(1, (2, 3), a_k)$ ,  $\tau^{(3)}(1, 2) = \pi(1, 2, , a_k)$ , and so on. We can calculate values  $\{\pi(*, (*, *), *)\}$  through equation (A.7) given the invariant vector  $\tilde{\pi}([*, *, *])$  of matrix  $\mathcal{Q}_6^{(3)}$ . Substituting these values of  $\{\pi(*, (*, *), *)\}$  into the right of equation (A.8), we obtain the invariant vector  $\tilde{\pi}([*, *, *, *])$  of transient matrix  $\mathcal{Q}_4^{(6)}$ . For larger transition matrices, we can apply the procedure again and again recursively.

**B. Computation sequence of the algorithm**

**For**  $m = 1$ :  $(\pi(1), \pi(2), \dots, \pi(M))$  : the invariant vector of the Markov chain  $P$ .

**For**  $m = 2$ :

$$\begin{array}{l} \text{Calc}[2] \\ \hookrightarrow \text{Proc}[1] \end{array}$$

**For**  $m = 3$ :

$$\begin{array}{l} \text{Calc}[3] \\ \hookrightarrow \text{Proc}[2] \end{array}$$

**For**  $m = 4$ :

$$\begin{array}{l} \text{Calc}[4] \\ \hookrightarrow \text{Proc}[3] \\ \hookrightarrow \text{Proc}[1] \end{array}$$

**For**  $m = 5$ :

$$\begin{array}{l} \text{Calc}[5] \rightarrow \text{Proc}[4] \\ \hookrightarrow \text{Proc}[1] \rightarrow \text{Proc}[2] \end{array}$$

**For**  $m = 6$ :

$$\begin{array}{l} \text{Calc}[6] \rightarrow \text{Proc}[5] \\ \hookrightarrow \text{Proc}[1] \rightarrow \text{Proc}[2] \rightarrow \text{Proc}[3] \\ \hookrightarrow \text{Proc}[1] \end{array}$$

**For**  $m = 7$ :

$$\begin{array}{l} \text{Calc}[7] \rightarrow \text{Proc}[6] \\ \hookrightarrow \text{Proc}[1] \rightarrow \text{Proc}[2] \rightarrow \text{Proc}[3] \rightarrow \text{Proc}[4] \\ \hookrightarrow \text{Proc}[1] \hookrightarrow \text{Proc}[1] \\ \hookrightarrow \text{Proc}[2] \end{array}$$

**For**  $m = 8$ :

$$\begin{array}{l} \text{Calc}[8] \rightarrow \text{Proc}[7] \\ \hookrightarrow \text{Proc}[1] \rightarrow \text{Proc}[2] \rightarrow \text{Proc}[3] \rightarrow \text{Proc}[4] \rightarrow \text{Proc}[5] \\ \hookrightarrow \text{Proc}[1] \hookrightarrow \text{Proc}[1] \hookrightarrow \text{Proc}[1] \\ \hookrightarrow \text{Proc}[2] \hookrightarrow \text{Proc}[2] \\ \hookrightarrow \text{Proc}[3] \\ \hookrightarrow \text{Proc}[1] \end{array}$$

## References

- [1] G. Birkov and S. Mac Lane, *A Survey of Modern Algebra* (The Macmillan Company, New York, Revised Edition, 1953).
- [2] R.P. Dobrow and J.A. Fill, The Move-to-front rule for self-organizing lists with Markov dependent requests, 57-80 (*Discrete Probability and Algorithms*, Springer-Verlag, 1995).
- [3] D. Ferrari, Program Behavior, *IEEE Computer*, **9** (11) (1976) 7-8.
- [4] M. Hofri and P. Tzelnic, The working set size distribution for the Markov chain model of program behavior, *SIAM J.Comput.* **11** (1982) 453-466.
- [5] K. Lam, M-Y Leung and M-K Siu, Self-organizing files with dependent accesses, *J.Appl.Prob.* **21** (1984) 334-359.
- [6] A. Graham, *Kronecker Products and Matrix Calculus with Applications* (Ellis Horwood Limited, 1981).
- [7] J-H Chu and G.D. Knott, A new method for computing page-fault rates, *SIAM J.Comput.* **22** (1993) 1319-1330.
- [8] J.G. Kemeny and J.L. Snell, *Finite Markov Chains* (D. Van Nostrand Company, 1960).
- [9] D. E. Knuth, *The Art of Computer Programing, Searching and Sorting*, **3** (Addison-Wesley, 1973).
- [10] B. Nitzberg and V. Lo, Distributed shared memory; A survey of issues and algorithms, *Computer*, (1991) 52-60.
- [11] Smith,A.J., Bibliography on paging and related topics, *Operating System Rev.*, **12** (1978) 39-56.
- [12] J.R. Spirn, Distance string models for program behavior, *IEEE Computer*, **9** (11) (1976) 14-20.
- [13] J.R. Spirn, *Program Behavior: Models and Measurements* (Elsevier-North Holland, N.Y., 1979).
- [14] R. Turner and B. Strecker, Use of the LRU stack depth distribution for simulation of paging behavior, *Comm.ACM*, **20** (11) (1977) 795-798.

Atsuhiko Tanaka (田中淳裕) and Issei Kino (紀一誠)  
C&C Media Research Laboratories NEC  
Miyazaki Miyamae-ku Kawasaki 216 Japan  
omochi@ccm.cl.nec.co.jp and kino@ccm.cl.nec.co.jp