Journal of the Operations Research
Society of Japan
Vol. 33, No. 4, December, 1990

APPLICATION OF THE REPLACEMENT PROCESS APPROACH FOR COMPUTING THE ERGODIC PROBABILITY VECTOR OF LARGE SCALE ROW-CONTINUOUS MARKOV CHAINS

Ushio Sumita
University of Rochester
and
International University of Japan

Maria Rieders
Northwestern University

(Received June 21, 1988; Final May 21, 1990)

Abstract In a recent paper by Sumita and Rieders (1990), a new algorithm has been developed for computing the ergodic probability vector for large Markov chains. Decomposing the state space into M lumps, the algorithm generates a sequence of replacement processes on individual lumps in such a way that in the limit the ergodic probability vector for a replacement process on one lump will be proportional to the ergodic probability vector of the original Markov chain restricted to that lump. In general, the algorithm involves the computation of inverse matrices of size $M-1$. Because of the skip free structure of row-continuous Markov chains, however, replacement distributions for these Markov chains can be constructed explicitly without involving any inverse matrices. The purpose of this paper is to develop an algorithm for computing the ergodic probability vector for row-continuous Markov chains based on the replacement process approach. Relevance to Takahashi's modified aggregation-disaggregation algorithm is also discussed. When successive substitution is employed for solving systems of linear equations, extensive numerical experiments suggest that both the replacement process algorithm and the ordinary aggregation-disaggregation algorithm dominate the modified aggregation-disaggregation algorithm. Furthermore, the replacement process algorithm outperforms the ordinary aggregation-disaggregation algorithm as both state space size and density of the underlying matrices increase.

1. Introduction

In the performance analysis of computer and production systems, one often encounters large scale bivariate Markov chains which have a skip free structure in their row index and can be described as row-continuous. Because of the prevalence of such chains in applications, many algorithms for computing the ergodic probability vector have been developed. Some algorithms find the ergodic distribution by solving a system of linear equations with special consideration of the underlying skip-free structure. The matrix geometric method of Neuts [13], subsequent improvements of Neuts' method by Ramaswami and Lucantoni [14], and the extended birth-death type recursive schemes by Latouche, Jacobs, and Gaver [11], Hajek [2], Keilson, Sumita and Zachmann [8], and Keilson and Zachmann [9] are such examples. Iterative schemes which find the ergodic distribution algorithmically starting with an initial guess are represented by the aggregation-disaggregation method of Takahashi [23] and Takahashi and Takami [25]. Subsequently, the aggregation-disaggregation approach has been studied extensively by Haviv [3,4], Haviv and Ritov [5], Haviv and Van der Heyden [6], Schweitzer [16] and Schweitzer, Puterman and Kindle [18]. The reader is referred to an excellent survey paper recently written by Schweitzer [17]. Of

special interest is the modified aggregation-disaggregation algorithm for row-continuous Markov chains described in Takahashi [23], exploiting the underlying skip-free structure. In the context of Ph/Ph/c queueing systems, Seelen [19] developed an algorithm built on the aggregation-disaggregation approach but incorporating various acceleration techniques such as overrelaxation, pointwise Gauss-Seidel iteration and weight factors based on the aggregation aspect.

The replacement process approach for achieving rank reduction has been first introduced in Sumita and Rieders [20] for computing the ergodic probability vector of time reversible and other specific Markov chains. In a subsequent paper by the same authors [21], the idea has been extended to incorporate general Markov chains. The algorithm of [21] using the replacement process approach is based on a decomposition of the state space $\mathcal{S}$ into lumps, i.e.

$$\mathcal{L} = \{L(1), \ldots, L(M)\};\ \mathcal{S} = \cup_{i=1}^{M} L(i). \tag{1.1}$$

Using positive probability vectors (typically a uniform distribution) on each lump in $\mathcal{L}\backslash L(1)$ as input, all states of individual lumps except $L(1)$ are aggregated. For this modified process, the replacement process on $L(1)$ is generated involving matrices of the sizes $|L(1)|$ and an inverse matrix of size $(M-1)$, where $|L(m)|$ denotes the number of states in $L(m)$. Using the resulting ergodic probability vector of the replacement process together with the input probability vector on $L(i)$, $i = 3, \ldots, M$, the replacement process for $L(2)$ can be constructed. The algorithm proceeds in a similar manner until it converges. It is shown that in the limit the ergodic probability vectors of the replacement processes are proportional to the ergodic probability vector of the original Markov chain restricted to individual lumps. The weights for individual lumps are then easily found by manipulating a square matrix of size M. Extensive numerical experiments reported in [21] suggest that the algorithm is quite promising.

In general, the replacement process algorithm requires the computation of inverse matrices of size $M-1$, although this task can be done recursively via rank two modification. For row-continuous Markov chains, however, the underlying skip free property enables one to construct the replacement distributions explicitly without computing any inverse matrices at all. Consequently, the application of the replacement process approach to row-continuous Markov chains results in a new algorithm of surprising simplicity and efficiency.

Although a theoretical proof has not been given, the new algorithm converged for all of more than 500 random numerical experiments with state space size varying from 100 to 1000 and sparsity level ranging from 0% (completely dense) to 80%. In all of these

experiments, monotone convergence in $\|\cdot\|_1$ norm was observed. Furthermore, the new algorithm is tested against the ordinary aggregation-disaggregation algorithm of Takahashi and its modified version tailored to row-continuous Markov chains. When successive substitution is employed as a means to solve systems of linear equations involved in these algorithms, the extensive numerical experiments reveal that both the replacement process algorithm and the ordinary aggregation-disaggregation algorithm dominate the modified aggregation-disaggregation algorithm. Furthermore, the replacement process algorithm outperforms the ordinary aggregation-disaggregation algorithm as both state space size and density of the underlying matrices increase.

In Section 2, a brief summary of the replacement process algorithm for general Markov chains is given from [21]. A new algorithm for row-continuous Markov chains is developed in Section 3 by applying the general replacement process algorithm with consideration of the skip free structure. In Section 4, the modified algorithm of Takahashi [23] for row-continuous Markov chains based on the aggregation-disaggregation approach is described and its relevance to the new algorithm is discussed. Section 5 summarizes the numerical results, and some concluding remarks are given in Section 6.

2. Replacement Process Approach for General Markov Chains

Let $J(k)$ be a discrete time Markov chain on $\mathcal{S} = \{1, \ldots, N\}$ goverend by one step transition probability matrix $\underline{\underline{a}} = [a_{ij}]_{i,j\in\mathcal{S}}$. Throughout the paper, we assume that $J(k)$ is ergodic having the ergodic probability vector $\underline{\pi} = [\pi_i]_{i\in\mathcal{S}}$. Let $\mathcal{L} = \{L(1), \ldots, L(M)\}$ be a decomposition of $\mathcal{S}$ as given in (1.1). For notational convenience, we define $\tilde{L}(m) = \mathcal{S}\backslash L(m)$ and $\tilde{m} = \mathcal{M}\backslash\{m\}$ where $\mathcal{M} = \{1, \ldots, M\}$. Submatrices of $\underline{\underline{a}}$ are written as $\underline{\underline{a}}_{L(m)L(n)} = [a_{ij}]_{i\in L(m), j\in L(n)}$, etc.

We now consider a Markov chain $J_m(k)$ on $L(m)$ governed by

$$\underline{\underline{a}}^*_{L(m)L(m)} = \underline{\underline{a}}_{L(m)L(m)} + \underline{\underline{a}}_{L(m)\tilde{L}(m)}\left[\underline{\underline{I}} - \underline{\underline{a}}_{\tilde{L}(m)\tilde{L}(m)}\right]^{-1}\underline{\underline{a}}_{\tilde{L}(m)L(m)}. \tag{2.1}$$

It should be noted that $J_m(k)$ is a replacement process on $L(m)$ obtained from the original process $J(k)$ where an exit from $L(m)$ is replaced instantaneously at some state in $L(m)$ according to replacement distributions specified in the second term of (2.1) while transitions within lump $L(m)$ are governed by the first term. It can be readily seen (see e.g. Courtois and Semal [1], Kemeny, Snell and Knapp [10], Meyer [12] and Sumita and Rieders [20,21]) that $J_m(k)$ governed by $\underline{\underline{a}}^*_{L(m)L(m)}$ has the ergodic probability vector which is proportional to $\underline{\pi}_{L(m)}$, the ergodic probability vector $\underline{\pi}$ of $J(k)$ restricted to $L(m)$.

If we find the ergodic probability vector of $\underline{\underline{a}}^*_{L(m)L(m)}$ for all $m \in \mathcal{M}$, the ergodic probability vector $\underline{\pi}$ of the original Markov chain can be obtained easily. However, for large

Markov chains, the cardinality of $\tilde{L}(m)$ is large and the computation of $[\underline{\underline{I}} - \underline{\underline{a}}_{\tilde{L}(m)\tilde{L}(m)}]^{-1}$ in (2.1) for all $m \in \mathcal{M}$ is simply overwhelming. Consequently, the use of (2.1) for obtaining $\underline{\pi}$ is computationally limited.

In order to avoid the computation of inverse matrices of large size, we consider a modified Markov chain $\tilde{J}^{(m)}(k)$ obtained from the original process $J(k)$ by aggregating all states in $L(n)$ into one state n^* for all lumps $L(n)$ except $L(m)$, using given probability vectors $\underline{y}_{L(n)} > \underline{0}$ on $L(n)$, $n \in \tilde{m}$. We write $\mathcal{M}^* = \{1^*, \ldots, M^*\}$ and $\tilde{m}^* = \mathcal{M}^* \backslash \{m^*\}$. The state space of $\tilde{J}^{(m)}(k)$ is $\tilde{m}^* \cup L(m)$ where $\tilde{m}^* = \mathcal{M}^* \backslash \{m^*\}$ represents aggregated states. The transition probability matrix $\underline{\underline{a}}_m(\underline{y}_{\tilde{L}(m)}) = [a_{m:ij}(\underline{y}_{\tilde{L}(m)})]_{i,j \in \tilde{m}^* \cup L(m)}$ of $\tilde{J}^{(m)}(k)$ is given by

$$(2.2) \qquad a_{m:ij}(\underline{y}_{\tilde{L}(m)}) = \begin{cases} \underline{y}^T_{L(i)} \underline{\underline{a}}_{L(i)L(j)} \underline{e}_{L(j)} & \text{for } i,j \in \tilde{m}^* \\ \underline{y}^T_{L(i)} \underline{\underline{a}}_{L(i)\{j\}} & \text{for } i \in \tilde{m}^*,\ j \in L(m) \\ \underline{\underline{a}}_{\{i\}L(j)} \underline{e}_{L(j)} & \text{for } i \in L(m),\ j \in \tilde{m}^* \\ a_{ij} & \text{for } i,j \in L(m) \end{cases}$$

where $\underline{e}_{L(j)}$ is a column vector having all components equal to one.

The replacement process $\tilde{J}^{(m)}_m(k)$ for this partially aggregated process $\tilde{J}^{(m)}(k)$ can be constructed based on (2.1), so that the ergodic probability vector of $\tilde{J}^{(m)}_m(k)$ on $L(m)$ is proportional to the ergodic probability vector of $\tilde{J}^{(m)}(k)$ restricted to $L(m)$. The transition probability matrix of $\tilde{J}^{(m)}_m(k)$ can then be obtained by computing an inverse matrix of size $(M-1)$ only.

For describing the modified replacement process $\tilde{J}^{(m)}_m(k)$, we first introduce the following two matrices.

$$(2.3) \qquad \underline{\underline{E}} = \begin{bmatrix} \underline{e}_{L(1)} & \underline{0} & \underline{0} & \cdots & \underline{0} \\ \underline{0} & \underline{e}_{L(2)} & \underline{0} & \cdots & \underline{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \underline{0} & & \cdots & & \underline{e}_{L(M)} \end{bmatrix} ; \quad N \times M,$$

and

$$(2.4) \qquad \underline{\underline{Y}} = \begin{bmatrix} \underline{y}^T_{L(1)} & \underline{0}^T & \underline{0}^T & \cdots & \underline{0}^T \\ \underline{0}^T & \underline{y}^T_{L(2)} & \underline{0}^T & \cdots & \underline{0}^T \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \underline{0}^T & & \cdots & & \underline{y}^T_{L(M)} \end{bmatrix} ; \quad M \times N,$$

where $\underline{y}_{L(n)} > \underline{0}$ is a probability vector on $L(n)$, $n \in \mathcal{M}$. We note that postmultiplying $\underline{\underline{a}}$ by $\underline{\underline{E}}$ corresponds to taking row sums within lumps $L(n)$, $n \in \mathcal{M}$. Similarly, premultiplying

$\underline{\underline{a}}$ by $\underline{\underline{Y}}$ results in taking a mixture of row vectors of $\underline{\underline{a}}$ in lump $L(n)$ according to $\underline{y}_{L(n)}$ for $n \in \mathcal{M}$. Let $\underline{y} = [\underline{y}_{L(n)}]_{n \in \mathcal{M}}$ and define for an $N \times N$ matrix $\underline{\underline{Z}}$

$$RS(\underline{\underline{Z}}) = \underline{\underline{Z}}\,\underline{\underline{E}} \tag{2.5}$$

and

$$MX(\underline{y}^T, \underline{\underline{Z}}) = \underline{\underline{Y}}\,\underline{\underline{Z}}, \tag{2.6}$$

where RS and MX stand for "Row Sum" and "Mixture" respectively. $RS(\underline{\underline{Z}}_{\tilde{L}(m)\tilde{L}(m)}) = \underline{\underline{Z}}_{\tilde{L}(m)\tilde{L}(m)}\underline{\underline{E}}_{\tilde{L}(m)\tilde{m}}$, $MX(\underline{y}^T_{\tilde{L}(m)}, \underline{\underline{Z}}_{\tilde{L}(m)L(m)}) = \underline{\underline{Y}}_{\tilde{m}\tilde{L}(m)}\underline{\underline{Z}}_{\tilde{L}(m)L(m)}$, etc. are defined in a similar manner.

Given probability vectors $\underline{y}_{\tilde{L}(m)} = [\underline{y}_{L(n)}]_{n \in \tilde{m}}$, $\underline{y}_{L(n)} > \underline{0}$, $\underline{y}^T_{L(n)}\underline{e}_{L(n)} = 1$, $n \in \tilde{m}$, the transition probability matrix governing $\tilde{J}^{(m)}_m(k)$ can now be given as

$$\underline{\underline{a}}^{**}_m(\underline{y}_{\tilde{L}(m)}) = \underline{\underline{a}}_{L(m)L(m)} + RS(\underline{\underline{a}}_{L(m)\tilde{L}(m)})[\underline{\underline{I}} - MX(\underline{y}^T_{\tilde{L}(m)}, RS(\underline{\underline{a}}_{\tilde{L}(m)\tilde{L}(m)}))]^{-1} MX(\underline{y}^T_{\tilde{L}(m)}, \underline{\underline{a}}_{\tilde{L}(m)L(m)}). \tag{2.7}$$

It has been shown in [21] that $\underline{\underline{a}}^{**}_m(\underline{y}_{\tilde{L}(m)})$ is an ergodic stochastic matrix for an arbitrary set of positive probability vectors $\underline{y}_{L(n)}$, $n \in \tilde{m}$. Furthermore, if $\underline{y}_{L(n)}$ is taken to be proportional to $\underline{\pi}_{L(n)}$ (the ergodic probability vector of the original Markov chain restricted to $L(n)$) for all $n \in \tilde{m}$, then the ergodic probability vector of $\underline{\underline{a}}^{**}_m(\underline{y}_{\tilde{L}(m)})$ is also proportional to $\underline{\pi}_{L(m)}$.

Once a set of probability vectors $\underline{y}_{L(m)}$ proportional to $\underline{\pi}_{L(m)}$ is found, it is easy to convert $\{\underline{y}_{L(m)}, m \in \mathcal{M}\}$ to $\{\underline{\pi}_{L(m)}, m \in \mathcal{M}\}$ via the aggregation procedure of Takahashi [23]. Indeed, if we define $\underline{\gamma}^T = [\gamma_1, \ldots, \gamma_M]$ by $\gamma_m \underline{y}_{L(m)} = \underline{\pi}_{L(m)}$ for all $m \in \mathcal{M}$, then $\underline{\gamma}^T$ is obtained as the ergodic vector of the stochastic matrix $\underline{\underline{\beta}} = MX(\underline{y}^T, RS(\underline{\underline{a}}))$.

Based on these results, iterative algorithms have been developed in [21] for computing the ergodic probability vector of large Markov chains. We next describe Version I which will help the reader to understand the development of the new algorithm for row-continuous Markov chains to be described in subsequent sections.

Algorithm 2.1: Replacement Process Algorithm (Version I)

Input: a transition probability matrix: $\underline{\underline{a}}$ $(N \times N)$;
lumping: $\mathcal{L} = \{L(1), \ldots, L(M)\}$;
vector: $\underline{y}^T_0 = [\underline{y}^T_{0:L(1)}, \ldots, \underline{y}^T_{0:L(M)}] > \underline{0}^T$, $\underline{y}^T_{0:L(n)}\underline{e}_{L(n)} = 1$, $n \in \mathcal{M}$;
level of accuracy: ϵ.

Output: a probability vector $\underline{q}^T = [\underline{q}^T_{L(1)}, \ldots, \underline{q}^T_{L(M)}]$ if it converges.

[1] Compute $\underline{\underline{\alpha}} = RS(\underline{\underline{a}}) = \underline{\underline{a}}\,\underline{\underline{E}}$. $(N \times M)$

[2] Set $\underline{y}^T_1 = \underline{y}^T = \underline{y}^T_0$.

[3] Compute $\underline{\underline{\beta}} = MX(\underline{y}^T, \underline{\underline{\alpha}}) = \underline{\underline{Y}}\,\underline{\underline{\alpha}}$. $(M \times M)$

[4] LOOP1: Set $m = 1$.

[5] LOOP2: Compute $\underline{\underline{X}}_{\bar{m}\,\bar{m}} = [\underline{\underline{I}} - \underline{\underline{\beta}}_{\bar{m}\,\bar{m}}]^{-1}$. $((M-1) \times (M-1))$

[6] Compute $\underline{\underline{\xi}}_{\bar{m}\,L(m)} = MX(\underline{y}^T_{\bar{L}(m)}, \underline{\underline{a}}_{\bar{L}(m)L(m)}) = \underline{\underline{Y}}_{\bar{m}\,\bar{L}(m)}\,\underline{\underline{a}}_{\bar{L}(m)L(m)}$. $((M-1)\times|L(m)|)$

[7] Set $\underline{\underline{a}}^{**}_m = \underline{\underline{a}}_{L(m)L(m)} + \underline{\underline{\alpha}}_{L(m)\bar{m}}\,\underline{\underline{X}}_{\bar{m}\,\bar{m}}\,\underline{\underline{\xi}}_{\bar{m}\,L(m)}$. $(|L(m)| \times |L(m)|)$

[8] Find $\underline{x}^T_{L(m)} > \underline{0}^T$ such that $\underline{x}^T_{L(m)} = \underline{x}^T_{L(m)}\underline{\underline{a}}^{**}_m$ and $\underline{x}^T_{L(m)}\underline{e}_{L(m)} = 1$.

[9] Set $\underline{\underline{Y}}_{mL(m)} = \underline{y}^T_{L(m)} = \underline{x}^T_{L(m)}$.

[10] Replace the m^{th} row of $\underline{\underline{\beta}}$ by $\underline{\beta}^T_{m\mathcal{M}} = \underline{x}^T_{L(m)}\underline{\underline{\alpha}}_{L(m)\mathcal{M}}$.

[11] If $m < M$, set $m = m + 1$ and go to LOOP2.

[12] If $\|\underline{y} - \underline{y}_1\| \geq \epsilon$, set $\underline{y}_1 = \underline{y}$ and go to LOOP1.

[13] Find $\underline{\gamma}^T > \underline{0}^T$ on $\mathcal{M}$ satisfying $\underline{\gamma}^T\underline{\underline{\beta}} = \underline{\gamma}^T$ and $\underline{\gamma}^T\underline{e}_{\mathcal{M}} = 1$.

[14] Construct $\underline{q}^T = [\underline{q}^T_{L(1)}, \ldots, \underline{q}^T_{L(M)}]$ by setting $\underline{q}^T_{L(m)} = \gamma_m\,\underline{y}^T_{L(m)}$, $m \in \mathcal{M}$.

In this algorithm, the row sum matrix $\underline{\underline{\alpha}} = RS(\underline{\underline{a}})$ is first prepared. Then initiated are a vector $\underline{y}^T$ and the matrix $\underline{\underline{\beta}} = MX(\underline{y}^T, \underline{\underline{\alpha}})$, which will be altered as the algorithm progresses. The outer loop starts in [4] LOOP1, while the inner loop starts in [5] LOOP2. The one-step transition probability matrix $\underline{\underline{a}}^{**}_m$ is constructed in [5] - [7] based on (2.7). The ergodic probability vector $\underline{x}_{L(m)}$ of $\underline{\underline{a}}^{**}_m$ is computed in [8], followed by updating $\underline{\underline{Y}}$ and $\underline{\underline{\beta}}$ in [9] and [10] using $\underline{x}_{L(m)}$. This process is continued until all lumps are covered, constituting one outer loop. The convergence is then examined in [12]. If a given accuracy is satisfied, lumped probabilities $\{\gamma_m,\ m \in \mathcal{M}\}$ are constructed in [13] and the final output is generated in [14].

It should be noted that $\underline{\underline{\beta}}_{\bar{m}+1,\bar{m}+1}$ in Algorithm 2.1 can be constructed from $\underline{\underline{\beta}}_{\bar{m}\,\bar{m}}$ by replacing one column and one row. Consequently, $\underline{\underline{X}}_{\bar{m}+1,\bar{m}+1}$ can be computed from $\underline{\underline{X}}_{\bar{m}\,\bar{m}}$ with rank 2 modification, as can be seen in [21].

3. A New Algorithm for Row-Continuous Markov Chains: Replacement Process Approach

In this section, we develop a new algorithm for computing the ergodic probability vector of row-continuous Markov chains via the replacement process approach.

Let $J(k)$ be an ergodic Markov chain on $\mathcal{S}$ governed by $\underline{\underline{a}}$ as described in Section 2. Let $\mathcal{L} = \{L(1), \ldots, L(M)\}$ be a decomposition of $\mathcal{S}$ as given in (1.1). The Markov chain

$J(k)$ is called row-continuous with respect to $\mathcal{C}$ if

$$\underline{\underline{a}}_{L(m)L(n)} = \underline{\underline{0}} \qquad \text{for } |m-n| > 1,\ m,n \in \mathcal{M}. \tag{3.1}$$

For convenience, we write

$$\underline{\underline{a}}_{L(m)L(n)} = \begin{cases} \underline{\underline{a}}_m^- & \text{if } n = m-1,\ m \in \mathcal{M}\backslash\{1\} \\ \underline{\underline{a}}_m^{(0)} & \text{if } n = m,\ m \in \mathcal{M} \\ \underline{\underline{a}}_m^+ & \text{if } n = m+1,\ m \in \mathcal{M}\backslash\{M\}. \end{cases} \tag{3.2}$$

Accordingly, the stochastic matrix $\underline{\underline{a}}$ is of the form

$$\underline{\underline{a}} = \begin{bmatrix} \underline{\underline{a}}_1^{(0)} & \underline{\underline{a}}_1^+ & & & & & & & & \\ \underline{\underline{a}}_2^- & \underline{\underline{a}}_2^{(0)} & \underline{\underline{a}}_2^+ & & & & & & & \\ & \underline{\underline{a}}_3^- & \underline{\underline{a}}_3^{(0)} & \ddots & & & & \underline{\underline{0}} & & \\ & & \ddots & \ddots & \ddots & & & & & \\ & & & \ddots & \underline{\underline{a}}_{m-1}^{(0)} & \underline{\underline{a}}_{m-1}^+ & & & & \\ & & & & \underline{\underline{a}}_m^- & \underline{\underline{a}}_m^{(0)} & \underline{\underline{a}}_m^+ & & & \\ & & & & & \underline{\underline{a}}_{m+1}^- & \underline{\underline{a}}_{m+1}^{(0)} & \ddots & & \\ & \underline{\underline{0}} & & & & & \ddots & \ddots & \ddots & \\ & & & & & & & \ddots & \underline{\underline{a}}_{M-1}^{(0)} & \underline{\underline{a}}_{M-1}^+ \\ & & & & & & & & \underline{\underline{a}}_M^- & \underline{\underline{a}}_M^{(0)} \end{bmatrix} \tag{3.3}$$

In order to construct the modified process $\tilde{J}^{(m)}(k)$ on $L(m)$, we introduce the two index sets

$$\begin{cases} G_m^- = \{1,\ldots,m-1\} & m \in \mathcal{M}\backslash\{1\} \\ G_m^+ = \{m+1,\ldots,M\} & m \in \mathcal{M}\backslash\{M\}. \end{cases} \tag{3.4}$$

For convenience we define $G_1^- = G_M^+ = \emptyset$. Given a set of probability vectors $\underline{y}_{L(n)}$ on $L(n)$, $n \in \tilde{m}$, let $\underline{\underline{b}}(m) = \underline{\underline{a}}_m(\underline{y}_{\tilde{L}(m)})$ be the one step transition probability matrix of $\tilde{J}^{(m)}(k)$. From (2.2) and the underlying row continuous structure, one then easily sees

that

$$\underline{\underline{b}}(m) = \left[\begin{array}{c|c|c} \underline{\underline{b}}_{G_m^- G_m^-} & \begin{matrix}\underline{\underline{0}} \\ \underline{\xi}_{m-1}^{+T}\end{matrix} & \underline{\underline{0}} \\ \hline \underline{\underline{0}} \quad \underline{\alpha}_m^- & \underline{\underline{b}}_{L(m)L(m)} & \underline{\alpha}_m^+ \quad \underline{\underline{0}} \\ \hline & \underline{\xi}_{m+1}^{-T} & \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{b}}_{G_m^+ G_m^+} \end{array}\right] \tag{3.5}$$

Here,

$$\left\{\begin{array}{l} \underline{\underline{b}}_{G_m^- G_m^-} = [\underline{y}_{L(i)}^T \underline{\underline{a}}_{L(i)L(j)} \underline{e}_{L(j)}]_{i,j \in G_m^-} \overset{\text{def}}{=} \begin{bmatrix} b_1^{(0)} & b_1^+ & & & \underline{\underline{0}} \\ b_2^- & b_2^{(0)} & b_2^+ & & \\ & \ddots & \ddots & \ddots & \\ & & b_{m-2}^- & b_{m-2}^{(0)} & b_{m-2}^+ \\ \underline{\underline{0}} & & & b_{m-1}^- & b_{m-1}^{(0)} \end{bmatrix} \\ \underline{\underline{b}}_{G_m^+ G_m^+} = [\underline{y}_{L(i)}^T \underline{\underline{a}}_{L(i)L(j)} \underline{e}_{L(j)}]_{i,j \in G_m^+} \overset{\text{def}}{=} \begin{bmatrix} b_{m+1}^{(0)} & b_{m+1}^+ & & & \underline{\underline{0}} \\ b_{m+2}^- & b_{m+2}^{(0)} & b_{m+2}^+ & & \\ & \ddots & \ddots & \ddots & \\ & & b_{M-1}^- & b_{M-1}^{(0)} & b_{M-1}^+ \\ \underline{\underline{0}} & & & b_M^- & b_M^{(0)} \end{bmatrix} \end{array}\right. \tag{3.6}$$

$$\left\{\begin{array}{l} \underline{\xi}_{m-1}^{+T} = \underline{y}_{L(m-1)}^T \underline{\underline{a}}_{m-1}^+; \\ \underline{\xi}_{m+1}^{-T} = \underline{y}_{L(m+1)}^T \underline{\underline{a}}_{m+1}^-; \end{array}\right. \tag{3.7}$$

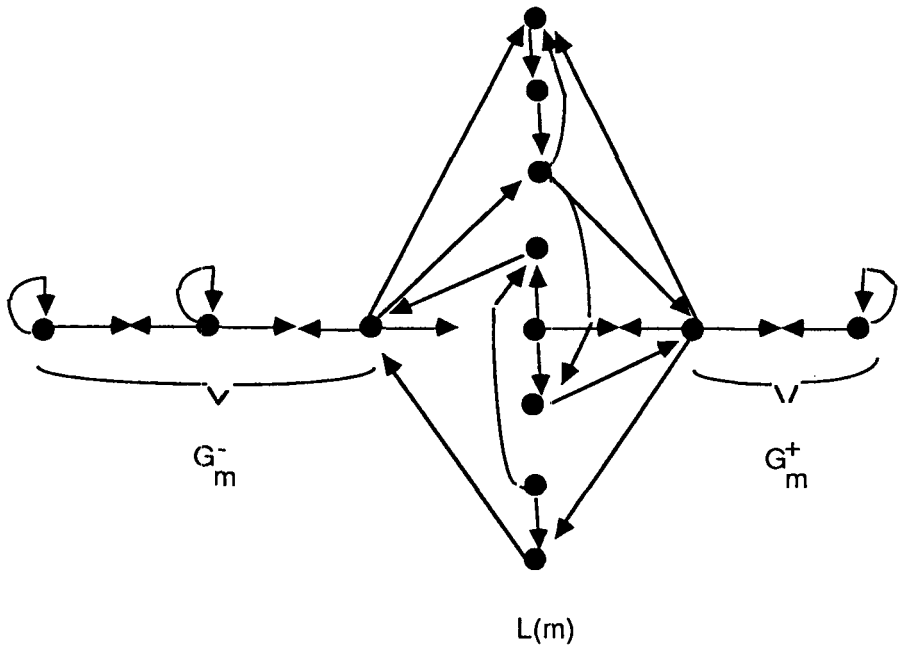
$$\left\{\begin{array}{l} \underline{\alpha}_m^- = \underline{\underline{a}}_m^- \underline{e}_{L(m-1)}; \\ \underline{\alpha}_m^+ = \underline{\underline{a}}_m^+ \underline{e}_{L(m+1)}; \end{array}\right. \tag{3.8}$$

and

$$\underline{\underline{b}}_{L(m)L(m)} = \underline{\underline{a}}_m^{(0)}. \tag{3.9}$$

A typical transition diagram for the process $\tilde{J}^{(m)}(k)$ is given in Figure 3.1.

Figure 3.1: Typical Transition Diagram for the Modified Process



The replacement process $\tilde{J}_m^{(m)}(k)$ obtained from this partially aggregated process $\tilde{J}^{(m)}(k)$ then has the one step transition probability matrix $\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)})$ which is given from (2.1) by

$$\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)}) = \underline{\underline{b}}_{L(m)L(m)} + \underline{\underline{b}}_{L(m)G(m)}[\underline{\underline{I}} - \underline{\underline{b}}_{G(m)G(m)}]^{-1}\underline{\underline{b}}_{G(m)L(m)}, \tag{3.10}$$

where $G(m) = G_m^- \cup G_m^+$. Exploiting the skip free structure of the transition probability matrix $\underline{\underline{b}}(m)$ in (3.5), one can simplify the representation of $\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)})$ substantially as we prove in the next theorem.

Theorem 3.1. *Let* $\underline{y}_{\tilde{L}(m)} = [\underline{y}_{L(n)}]_{n\in\tilde{m}}$, $\underline{y}_{L(n)} > \underline{0}$, $\underline{y}_{L(n)}^T \underline{e}_{L(n)} = 1$, $n \in \tilde{m}$. *Then* $\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)})$ *in (3.10) can be written as*

$$\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)}) = \underline{\underline{a}}_m^{(0)} + \frac{\alpha_m^- \underline{\xi}_{m-1}^{+T}}{\underline{\xi}_{m-1}^{+T}\underline{e}_{L(m)}} + \frac{\alpha_m^+ \underline{\xi}_{m+1}^{-T}}{\underline{\xi}_{m+1}^{-T}\underline{e}_{L(m)}}, \tag{3.11}$$

where for $m = 1$ the second term and for $m = M$ the last term on the right hand side is set to zero.

Proof: Let $\eta_m^- = (\underline{\xi}_{m-1}^{+T} \underline{e}_{L(m)})^{-1}$ and $\eta_m^+ = (\underline{\xi}_{m+1}^{-T} \underline{e}_{L(m)})^{-1}$. One then sees from (3.5) that

$$
\begin{aligned}
[\underline{\underline{I}} - \underline{\underline{b}}_{G(m)G(m)}] & \begin{bmatrix} \eta_m^- \underline{e}_{G_m^-} \underline{\xi}_{m-1}^{+T} \\ \\ \eta_m^+ \underline{e}_{G_m^+} \underline{\xi}_{m+1}^{-T} \end{bmatrix} \\
&= \begin{bmatrix} \eta_m^- [\underline{\underline{I}} - \underline{\underline{b}}_{G_m^- G_m^-}] \underline{e}_{G_m^-} \underline{\xi}_{m-1}^{+T} \\ \\ \eta_m^+ [\underline{\underline{I}} - \underline{\underline{b}}_{G_m^+ G_m^+}] \underline{e}_{G_m^+} \underline{\xi}_{m-1}^{+T} \end{bmatrix} \\
&= \begin{bmatrix} \underline{\underline{0}} \\ \underline{\xi}_{m-1}^{+T} \\ \\ \underline{\xi}_{m+1}^{-T} \\ \underline{\underline{0}} \end{bmatrix} = \underline{\underline{b}}_{G(m)L(m)}.
\end{aligned}
$$

Hence, it follows that

$$
[\underline{\underline{I}} - \underline{\underline{b}}_{G(m)G(m)}]^{-1} \underline{\underline{b}}_{G(m)L(m)} = \begin{bmatrix} \eta_m^- \underline{e}_{G_m^-} \underline{\xi}_{m-1}^{+T} \\ \\ \eta_m^+ \underline{e}_{G_m^+} \underline{\xi}_{m+1}^{-T} \end{bmatrix}. \tag{3.12}
$$

Substituting (3.12) into (3.10), one then sees that

$$
\begin{aligned}
\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)}) &= \underline{\underline{b}}_{L(m)L(m)} + \left[\underline{\underline{0}}, \underline{\alpha}_m^-, \underline{\alpha}_m^+, \underline{\underline{0}}\right] \begin{bmatrix} \eta_m^- \underline{e}_{G_m^-} \underline{\xi}_{m-1}^{+T} \\ \\ \eta_m^+ \underline{e}_{G_m^+} \underline{\xi}_{m+1}^{-T} \end{bmatrix} \\
&= \underline{\underline{a}}_{L(m)L(m)} + \eta_m^- \underline{\alpha}_m^- \underline{\xi}_{m-1}^{+T} + \eta_m^+ \underline{\alpha}_m^+ \underline{\xi}_{m+1}^{-T},
\end{aligned}
$$

proving the theorem. □

From Theorems 1.1 through 1.3 of Sumita and Rieders [21], the matrix $\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)})$ in (3.11) can then provide a basis for the replacement process approach applied to row continuous Markov chains. More specifically, $\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)})$ in line [7] of Algorithm 2.1 can be replaced by the expression in (3.11). Accordingly, the computation of $\underline{\underline{\beta}}$, the inverse

matrix $\underline{\underline{X}}_{\tilde{m}\,\tilde{m}} = [\underline{\underline{I}} - \underline{\underline{\beta}}_{\tilde{m}\,\tilde{m}}]^{-1}$ and $\underline{\underline{\xi}}_{\tilde{m}\,L(m)} = MX(\underline{y}^T_{\tilde{L}(m)}, \underline{\underline{a}}_{\tilde{L}(m)L(m)})$ can be totally eliminated from the loop of the algorithm. Furthermore, it is clear from Figure 3.1 that the lumping probability vector $\underline{\gamma}^T$ can be found as the ergodic probability vector of an ordinary truncated birth-death process. These observations lead to the following algorithm of surprising simplicity and efficiency.

Algorithm 3.2: Replacement Process Algorithm for Row Continuous Markov Chains

Input: a transition probability matrix: $\underline{\underline{a}}$ $(N \times N)$ in the form

$$\underline{\underline{\hat{a}}} = \begin{bmatrix} \underline{\underline{0}} & \underline{\underline{a}}^{(0)}_1 & \underline{\underline{a}}^{+}_1 \\ \underline{\underline{a}}^{-}_2 & \underline{\underline{a}}^{(0)}_2 & \underline{\underline{a}}^{+}_2 \\ \vdots & \vdots & \vdots \\ \underline{\underline{a}}^{-}_{M-1} & \underline{\underline{a}}^{(0)}_{M-1} & \underline{\underline{a}}^{+}_{M-1} \\ \underline{\underline{a}}^{-}_{M} & \underline{\underline{a}}^{(0)}_{M} & \underline{\underline{0}} \end{bmatrix}$$

lumping: $\mathcal{L} = \{L(1), \ldots, L(M)\}$;

vector: $\underline{y}^T_0 = [\underline{y}^T_{0:L(1)}, \ldots, \underline{y}^T_{0:L(M)}] > \underline{0}^T$, $\underline{y}^T_{0:L(n)}\underline{e}_{L(n)} = 1$, $n \in \mathcal{M}$;

level of accuracy: ϵ.

Output: a probability vector $\underline{q}^T = [\underline{q}^T_{L(1)}, \ldots, \underline{q}^T_{L(M)}]$ when it converges.

[1] Compute $\underline{\underline{\alpha}} = RS(\underline{\underline{\hat{a}}}) = \underline{\underline{\hat{a}}}\,\underline{\underline{E}} = [\underline{\underline{\alpha}}^{-}_{L(m)}, \underline{\underline{\alpha}}^{(0)}_{L(m)}, \underline{\underline{\alpha}}^{+}_{L(m)}]_{m \in \mathcal{M}}$. $(N \times 3)$

[2] Set $\underline{y}^T_1 = \underline{y}^T = \underline{y}^T_0$.

[3] LOOP1: Set $m = 1$.

[4] LOOP2: Compute $\underline{\xi}^{+T}_{m-1} = \underline{y}^T_{L(m-1)}\underline{\underline{a}}^{+}_{m-1}$ and $\underline{\xi}^{-T}_{m+1} = \underline{y}^T_{L(m+1)}\underline{\underline{a}}^{-}_{m+1}$.

[5] Set $\underline{\zeta}^{+T}_{m-1} = \underline{\xi}^{+T}_{m-1}/\underline{\xi}^{+T}_{m-1}\underline{e}_{L(m)}$ and $\underline{\zeta}^{-T}_{m+1} = \underline{\xi}^{-T}_{m+1}/\underline{\xi}^{-T}_{m+1}\underline{e}_{L(m)}$.

[6] Set $\underline{\underline{a}}^{**}_m = \underline{\underline{a}}^{(0)}_m + \underline{\underline{\alpha}}^{-}_{L(m)}\underline{\zeta}^{+T}_{m-1} + \underline{\underline{\alpha}}^{+}_{L(m)}\underline{\zeta}^{-T}_{m+1}$. $(|L(m)| \times |L(m)|)$

[7] Find $\underline{x}^T_{L(m)} > \underline{0}^T$ such that $\underline{x}^T_{L(m)} = \underline{x}^T_{L(m)}\underline{\underline{a}}^{**}_m$ and $\underline{x}^T_{L(m)}\underline{e}_{L(m)} = 1$.

[8] Set $\underline{\underline{Y}}_{m\,L(m)} = \underline{y}^T_{L(m)} = \underline{x}^T_{L(m)}$.

[9] JUMP2: If $m < M$, set $m = m + 1$ and go to LOOP2.

[10] If $\|y_1 - y\| \geq \epsilon$, set $\underline{y}_1 = \underline{y}$ and go to LOOP1.

[11] Compute $\underline{\underline{\beta}} = MX(\underline{y}^T, \underline{\underline{\alpha}}) = \underline{\underline{Y}}\,\underline{\underline{\alpha}}$. $(M \times 3)$

[12] Find $\underline{\gamma}^T > \underline{0}^T$ as the ergodic probability vector of the birth- death process with transition probabilities at state n given in the n^{th} row of $\underline{\underline{\beta}}$.

[13] Construct $\underline{q}^T = [\underline{q}^T_{L(1)}, \ldots, \underline{q}^T_{L(M)}]$ by setting $\underline{q}^T_{L(m)} = \gamma_m \underline{y}^T_{L(m)}$, $m \in \mathcal{M}$.

This algorithm differs from Algorithm 2.1 only in the way the one step transition probability matrix $\underline{\underline{a}}^{**}_m(\underline{y}_{\tilde{L}(m)})$ governing the replacement process is computed. In Algorithm 2.1, $\underline{\underline{a}}^{**}_m(\underline{y}_{\tilde{L}(m)})$ is computed in lines [5]–[7] involving the inverse matrix $\underline{\underline{X}}_{\tilde{m}\,\tilde{m}}$ of size

$M-1$. As we have seen in (3.11) of Theorem 3.1, the skip free structure enables one to construct the matrix $\underline{\underline{a}}_m^{**}(\underline{y}_{L(m)})$ without involving any inverse matrices at all. This computation is done in lines [4]–[6] of Algorithm 3.2. We also note that Algorithm 2.1 requires updating the matrix $\underline{\underline{\beta}}$ inside the loop (line [10]), while the new algorithm eliminates this computation.

Remark 3.3. *In actual computation, it is more efficient to combine lines [5] through [7] in Algorithm 3.2 by applying the power method in the following manner:*

$$\underline{v}_2^T = \underline{v}_1^T \underline{\underline{a}}_m^{(0)} + \frac{\underline{v}_1^T \underline{\alpha}_{L(m)}^-}{\underline{\xi}_{m-1}^{+T} \underline{e}_{L(m)}} \underline{\xi}_{m-1}^{+T} + \frac{\underline{v}_1^T \underline{\alpha}_{L(m)}^+}{\underline{\xi}_{m+1}^{-T} \underline{e}_{L(m)}} \underline{\xi}_{m+1}^{-T}.$$

Remark 3.4. *Algorithm 3.2 does not require all lump sizes (i.e. the number of states in each row) to be the same. In this regard, the notation* $\underline{\underline{a}}$ *and the operation RS applied to* $\underline{\underline{a}}$ *should be read symbolically.*

4. Takahashi's Modified Aggregation-Disaggregation Approach and Its Relevance to the New Algorithm

A modified aggregation-disaggregation algorithm has been proposed by Takahashi [23] for computing the ergodic probability vector for large scale row-continuous Markov chains. The purpose of this section is to describe this algorithm and to discuss its relevance to the new algorithm. For convenience, the notation in [23] is converted to the notation of this paper.

The idea behind the modification is to eliminate the aggregation procedure and to simplify the disaggregation procedure by taking advantage of the skip free structure. More specifically, given two probability vectors $\underline{y}_{L(m-1)}$ and $\underline{y}_{L(m+1)}$, the modified algorithm introduces a Markov chain $\hat{J}_m(k)$ on $L(m) \cup \{(m-1)^*, (m+1)^*\}$ governed by

$$\underline{\underline{\hat{a}}}_m = \begin{bmatrix} 1 - \underline{\xi}_{m-1}^{+T} \underline{e}_{L(m)} & \underline{\xi}_{m-1}^{+T} & 0 \\ \underline{\alpha}_m^- & \underline{\underline{a}}_m^{(0)} & \underline{\alpha}_m^+ \\ 0 & \underline{\xi}_{m+1}^{-T} & 1 - \underline{\xi}_{m+1}^{-T} \underline{e}_{L(m)} \end{bmatrix}. \tag{4.1}$$

Let $\underline{v}^T = [v^-, \underline{v}_{L(m)}^T, v^+]$ be the ergodic probability vector of $\underline{\underline{\hat{a}}}_m$ so that $\underline{v}^T = \underline{v}^T \underline{\underline{\hat{a}}}_m$. One then has

$$v^- = v^- (1 - \underline{\xi}_{m-1}^{+T} \underline{e}_{L(m)}) + \underline{v}_{L(m)}^T \underline{\alpha}_m^-, \tag{4.2}$$

and

$$\underline{v}_{L(m)}^T = v^- \underline{\xi}_{m-1}^{+T} + \underline{v}_{L(m)}^T \underline{\underline{a}}_m^{(0)} + v^+ \underline{\xi}_{m+1}^{-T}. \tag{4.3}$$

If we define $\underline{\pi}^{-T}_{L(m)} = \underline{\xi}^{+T}_{m-1}(\underline{I} - \underline{\underline{a}}^{(0)}_{m})^{-1}$ and $\underline{\pi}^{+T}_{L(m)} = \underline{\xi}^{-T}_{m+1}(\underline{I} - \underline{\underline{a}}^{(0)}_{m})^{-1}$, Equation (4.3) can be rewritten as

$$\underline{v}^{T}_{L(m)} = v^{-}\underline{\pi}^{-T}_{L(m)} + v^{+}\underline{\pi}^{+T}_{L(m)}. \tag{4.4}$$

By substituting (4.4) into (4.2), it then follows that

$$v^{+} = \frac{B}{A}v^{-}, \tag{4.5}$$

where

$$A = \underline{\pi}^{+T}_{L(m)}\underline{\alpha}^{-}_{m}; \quad B = \underline{\pi}^{-T}_{L(m)}\underline{\alpha}^{+}_{m}. \tag{4.6}$$

Substitution of (4.5) back into (4.4) then yields

$$\underline{v}^{T}_{L(m)} = \frac{v^{-}}{A}[A\underline{\pi}^{-T}_{L(m)} + B\underline{\pi}^{+T}_{L(m)}]. \tag{4.7}$$

One inner loop of the modified algorithm for lump $L(m)$ can now be described as follows.

Algorithm 4.1: Modified Aggregation-Disaggregation Algorithm (One inner loop for lump $L(m)$).

Input: two probability vectors $\underline{y}^{T}_{L(m-1)}$ on $L(m-1)$ and $\underline{y}^{T}_{L(m+1)}$ on $L(m+1)$.

Output: a probability vector $\underline{y}^{T}_{L(m)}$ on $L(m)$.

[1] Find $\underline{\pi}^{-T}_{L(m)}$ by solving

$$\underline{\pi}^{-T}_{L(m)} = \underline{\pi}^{-T}_{L(m)}\underline{\underline{a}}^{(0)}_{m} + \underline{\xi}^{+T}_{m-1}. \tag{4.8}$$

[2] Find $\underline{\pi}^{+T}_{L(m)}$ by solving

$$\underline{\pi}^{+T}_{L(m)} = \underline{\pi}^{+T}_{L(m)}\underline{\underline{a}}^{(0)}_{m} + \underline{\xi}^{-T}_{m+1}. \tag{4.9}$$

[3] Compute $\hat{\underline{\pi}}^{T}_{L(m)} = (A/B)\underline{\pi}^{-T}_{L(m)} + \underline{\pi}^{+T}_{L(m)}$, where A and B are as given in (4.6).

[4] Produce $\underline{y}^{T}_{L(m)}$ by $\underline{y}^{T}_{L(m)} = \hat{\underline{\pi}}^{T}_{L(m)}/\hat{\underline{\pi}}^{T}_{L(m)}\underline{e}_{L(m)}$.

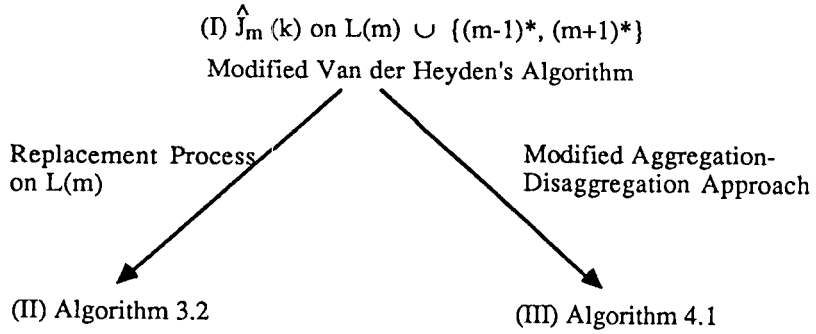
From (4.7), it can be readily seen that $\underline{y}^{T}_{L(m)}$ generated by Algorithm 4.1 is proportional to $\underline{v}^{T}_{L(m)}$ which is the ergodic probability vector of $\underline{\underline{\hat{a}}}_{m}$ on $L(m)$.

In Algorithm 4.1, for $m = 1$, Equation (4.8) is ignored and $\underline{\pi}^{-T}_{L(1)} = \underline{0}^{T}$ while for $m = M$, Equation (4.9) is ignored and $\underline{\pi}^{+T}_{L(M)} = \underline{0}^{T}$. Starting with initial probability vectors on individual lumps, Algorithm 4.1 can be applied repeatedly until convergence is reached. While Equations (4.8) and (4.9) preserve the form of the disaggregation procedure

in a simplified form, the aggregation procedure is totally eliminated. We next discuss the relevance of Algorithm 4.1 to the new algorithm.

It should be noted from (3.5) and (4.1) that the Markov chain $\hat{J}_m(k)$ governed by (4.1) can be obtained from the Markov chain $\tilde{J}^{(m)}(k)$ by aggregating all states in G_m^- into one state $(m-1)^*$ and all states in G_m^+ into one state $(m+1)^*$. Because of the skip free structure in G_m^- and G_m^+, one easily sees that the replacement process constructed from $\hat{J}_m(k)$ is statistically identical to the replacement process $\tilde{J}_m^{(m)}(k)$ constructed from $\tilde{J}^{(m)}(k)$. Hence, both Algorithm 3.2 via the replacement process approach and Algorithm 4.1 via the modified aggregation-disaggregation approach are built on the Markov chain $\hat{J}_m(k)$, as depicted in Figure 4.1.

Figure 4.1 Relationship between Algorithm 3.2 and Algorithm 4.1



We have seen that $\underline{y}^T_{L(m)}$ generated from Algorithm 4.1 is proportional to the ergodic probability vector of $\hat{J}_m(k)$ restricted to $L(m)$. Because of the characteristics of the replacement process, the output vector of Algorithm 3.2 also possesses this property. As Takahashi [24] suggested, a third alternative is available without using Algorithm 3.2 and Algorithm 4.1, where the power method is directly applied to $\underline{\hat{a}}_m$ of (4.1) governing $\hat{J}_m(k)$, followed by appropriate normalization. This corresponds to a modification of Van der Heyden's result reported in Haviv [4]. According to Figure 4.1, we refer to these three algorithms as RP-Algo (Algorithm 3.2), MAD-Algo (Algorithm 4.1), and VH-Algo (modified Van der Heyden's algorithm).

Although all three algorithms generate exactly the same probability vector on $L(m)$, the ideas behind the algorithms and their performance characteristics are quite different. If transitions within each row form a special structure (e.g. $\underline{a}^{(0)}_m$, $m \in \mathcal{M}$, are upper

diagonal matrices), MAD-Algo may be attractive since the inverse $[\underline{\underline{I}} - \underline{\underline{a}}_m^{(0)}]^{-1}$ may be computed straightforwardly. If successive substitution is employed, however, MAD-Algo is less efficient than VH-Algo and RP-Algo because two different systems of linear equations of size $|L(m)|$ have to be solved. In general, there exists a tradeoff between VH-Algo and RP-Algo. VH-Algo requires the power method applied to the matrix $\underline{\underline{\hat{a}}}_m$ of size $|L(m)|+2$, while RP-Algo employs that applied to $\underline{\underline{a}}_m^{**}(\underline{y}_{L(m)})$ of size $|L(m)|$. However, RP-Algo requires rank 2 modification. In what follows we examine the complexity of the three algorithms assuming no special structure for transitions within each row. For MAD-Algo, Equations (4.8) and (4.9) are solved by successive substitution which seems to be most efficient, see Schweitzer [16]. The number of iterations needed for Algorithm X is denoted by K_X. We also denote the size of lump $L(m)$ by $L = |L(m)|$. Tables 4.2 through 4.4 give a detailed complexity analysis for each of the three algorithms, by stating the number of additions, multiplications and divisions for each step in the algorithm. For comparison, these results are summarized in Table 4.5. One should note that the complexity analysis assumes that the row sum matrix $\underline{\underline{a}}$ is computed outside the loop and that MAD-Algo employs only $\underline{\alpha}_m^-$ whereas the other algorithms need both $\underline{\alpha}_m^-$ and $\underline{\alpha}_m^+$.

Table 4.2: Complexity of RP-Algo

Step	Additions	Multiplications	Divisions
$\underline{\xi}_{m+1}^{-T}$, $\underline{\xi}_{m-1}^{+T}$	$2(L-1)L$	$2L^2$	0
$\underline{\xi}_{m+1}^{-T}\underline{e}_{L(m)}$, $\underline{\xi}_{m-1}^{+T}\underline{e}_{L(m)}$	$2(L-1)$	0	0
Power method (see Remark 3.3)	$K_{RP}(L^2+3L-2)$	$K_{RP}(L^2+4L)$	$2K_{RP}$

Table 4.3: Complexity of MAD-Algo

Step	Additions	Multiplications	Divisions
$\underline{\xi}_{m+1}^{-T}$, $\underline{\xi}_{m-1}^{+T}$	$2(L-1)L$	$2L^2$	0
$\underline{\pi}_{L(m)}^{-T}$, $\underline{\pi}_{L(m)}^{+T}$	$2K_{MAD}L^2$	$2K_{MAD}L^2$	0
A	$(L-1)$	L	0
B	$(L-1)$	L	0
$\hat{\underline{\pi}}_{L(m)}^{T}$	L	L	1
$\underline{y}_{L(m)}^{T}$	$L-1$	0	L

Table 4.4: Complexity of VH-Algo

Step	Additions	Multiplications	Divisions
$\underline{\xi}_{m+1}^{-T}$, $\underline{\xi}_{m-1}^{+T}$	$2(L-1)L$	$2L^2$	0
$1-\underline{\xi}_{m+1}^{-T}\underline{e}_{L(m)}$, $1-\underline{\xi}_{m-1}^{+T}\underline{e}_{L(m)}$	$2L$	0	0
Power method	$K_{VH}(L+1)(L+2)$	$K_{VH}(L+2)^2$	0
Normalization	$L-1$	0	L

Table 4.5: Comparative Complexity Analysis

Method	Additions	Multiplications	Divisions
RP − Algo	$(K_{RP}+2)L^2+3K_{RP}L$ $-2(K_{RP}+1)$	$(K_{RP}+2)L^2$ $+4K_{RP}L$	$2K_{RP}$
MAD − Algo	$(2K_{MAD}+2)L^2+2L-3$	$(2K_{MAD}+2)L^2$ $+3L$	$L+1$
VH − Algo	$(K_{VH}+2)L^2+(3K_{VH}+1)L$ $+(2K_{VH}-1)$	$(K_{VH}-1)L^2$ $+4K_{VH}L+4K_{VH}$	L

Remark 4.2. *When successive substitution is employed, the number K_{RP} is basically determined by the second maximum eigenvalue of the stochastic replacement matrix $\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)})$ while K_{MAD} is determined by the maximum eigenvalue of $\underline{\underline{a}}_m^{(0)}$ and K_{VH} by the second maximum eigenvalue of Van der Heyden's stochastic matrix $\underline{\underline{\hat{a}}}_m$. It is quite difficult to theoretically order these numbers. As we will see, however, extensive numerical experiments suggest that K_{RP} is likely to be less than the other two numbers.*

Remark 4.3. *In MAD-Algo and in the ordinary aggregation-disaggregation algorithm it is possible to compute the inverse matrices $(\underline{\underline{I}}-\underline{\underline{a}}_m^{(0)})^{-1}$ outside the loop using LU-decomposition or similar techniques. If this is done, the systems of linear equations have to be solved only once and their complexity only depends on $|L(m)|$. When the size of the underlying problem is extremely large – a situation which the aggregation-disaggregation approach is originally intended for – the memory requirements for storing these inverse matrices may be prohibitive. The algorithms tested in this paper do not include this approach.*

5. Numerical Comparison

Although the complexity analysis given in Tables 4.2 through 4.5 is informative, one cannot draw any general conclusions regarding the computational efficiency of the three algorithms studied above. In the context of Ph/Ph/c queueing systems, Seelen [19] stated that possibly the main strength of the aggregation-disaggregation algorithm lies in the introduction of the aggregation aspect rather than the disaggregation aspect. Since the modified aggregation-disaggregation algorithm omits the aggregation part, one may therefore expect that the ordinary aggregation-disaggregation algorithm may work better than the modified version even for general row-continuous Markov chains. Because of this reason, it is interesting to include the ordinary aggregation-disaggregation algorithm (AD-Algo) in a numerical comparative analysis. For details of the ordinary aggregation- disaggregation algorithm, the reader is referred to Takahashi [23] or Sumita and Rieders [21].

In order to examine the efficiency and accuracy of RP-Algo in comparison with MAD-Algo, AD-Algo and VH-Algo, extensive numerical experiments have been conducted. In all four algorithms, successive substitution is employed for solving systmes of linear equations involved. As initial distributions, uniform probability vectors are used. For implementing successive substitution after the initial cycle however, the intermediary probability vectors obtained from the previous cycle are used as initial probability vectors of the current cycles. In doing this, RP-Algo requires no extra efforts while all other algorithms do require additional storage space. The cost of additional memory is minimal which is likely to be dominated by the advantage of accelerated convergence. This precaution was not incorporated in the preliminary report given in [22]. The purpose of this section is to report the numerical results comparing the four algorithms. We begin this by describing the frame of the numerical experiments.

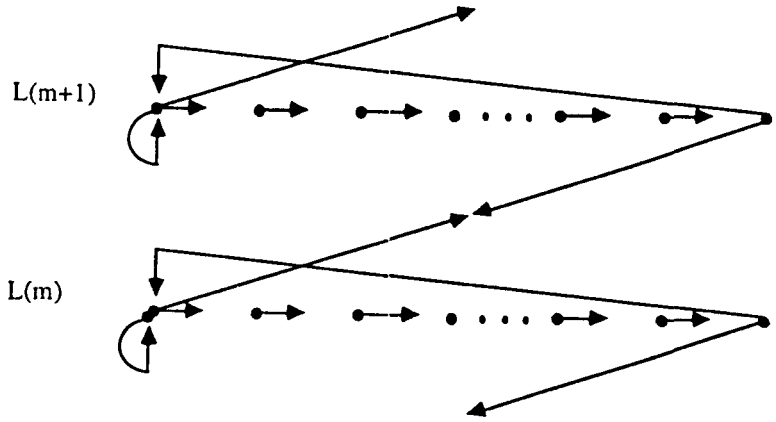
(A) State Space Size and Lumping

State Space Size N	No. of Rows M	Row Size L
100	10	10
200	10	20
300	10	30
400	10	40
500	10	50
600	10	60
700	10	70
800	10	80
900	10	90
1000	10	100

(B) Sparsity Level ρ

In order to assure ergodicity, the transition structure of an almost periodic Markov chain is used as a basis for each $\underline{\underline{a}}_m^{(0)}$, see Figure 5.1. Additional transitions among the states are then generated randomly. The levels of sparsity for those submatrices are 0, 20, 40, 60, and 80%, where sparsity level 0% corresponds to completely dense matrices $\underline{\underline{a}}_m^-$, $\underline{\underline{a}}_m^{(0)}$ and $\underline{\underline{a}}_m^+$, while 80% implies the matrices have only 20% nonzero entries.

Figure 5.1: Basic Markov Chain



(C) Number of Experiments

For each pair of N and sparsity level ρ, three random experiments have been conducted, totalling 150 experiments.

(D) Level of Accuracy ϵ

In all experiments, the level of accuracy ϵ is $10^{-5}/N$, where N is the state space size. For the successive substitution employed in the lump computations of all algorithms, an accuracy level of ϵ/M is chosen where M is the number of lumps.

(E) Performance Measures

As performance measures for comparing the four algorithms, we use the number of nonzero operations for additions, multiplications and divisions, separately. Given a common accuracy level, the three algorithms RP-Algo, MAD- Algo and VH-Algo always have the same number of outer loops, while this number may be different for AD-Algo. The speed of convergence of the successive substitution for individual lump computations can be quite different, which is also of our interest.

Table 5.2 summarizes the average number of outer loops for each pair of sparsity level ρ and state space size N. The upper entries are for the three algorithms (RP,MAD,VH) and the lower entries are for AD-Algo. As can be seen in Table 5.2, the number of outer loops decreases as N increases and sparsity level ρ decreases. AD-Algo almost always requires a smaller number of outer loops than the other three algorithms. However, this does not immediately imply the performance superiority because the number of iterations on individual lumps may be different, as we see next.

The average numbers of iterations required for one lump computation are summarized in Table 5.3 for the four different algorithms. We note that these numbers for MAD-Algo correspond to $2K_{MAD}$ in the complexity analysis of Table 4.3. In principle, general trends similar to those of Table 5.2 can be observed. It should be noted that RP-Algo requires the least number of iterations for one lump computation. This indicates that the second maximum eigenvalue of the stochastic replacement matrix $\underline{\underline{a}}_m^{**}(\underline{y}_{\tilde{L}(m)})$ corresponding to a lump $L(m)$ is likely to be less than the maximum eigenvalue of $\underline{\underline{a}}_m^0$ or the second maximum eigenvalue of Van der Heyden's stochastic matrix $\underline{\underline{\hat{a}}}_m$. From Tables 5.2 and 5.3, one expects that both RP-Algo and AD-Algo dominate the other two algorithms MAD-Algo and VH-Algo. RP-Algo and AD-Algo, however, compete against each other in different aspects, with the number of outer loops favoring AD-Algo and the number of iterations for lump computations favoring RP-Algo. In what follows, we continue the comparative analysis with further details.

Table 5.2: Average Number of Outer Loops

$\rho \backslash N$	100	200	300	400	500	600	700	800	900	1000
0.0	6.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	4.0
	5.0	5.0	4.7	4.0	4.0	4.0	4.0	4.0	4.0	4.0
0.2	7.0	6.0	6.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
	6.0	5.0	5.0	5.0	5.0	5.0	4.0	4.0	4.0	4.0
0.4	7.3	7.0	6.0	6.0	6.0	6.0	6.0	5.0	5.0	5.0
	6.7	6.0	5.3	5.0	5.0	5.0	5.0	5.0	5.0	5.0
0.6	9.0	8.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	6.0
	8.0	7.0	6.0	6.0	6.0	5.3	5.0	5.0	5.0	5.0
0.8	11.0	10.0	9.0	9.0	8.0	7.0	7.0	7.0	7.0	7.0
	10.3	8.7	8.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0

Table 5.3: Average Number of Iterations on Lumps

$\rho \backslash N$		100	200	300	400	500	600	700	800	900	1000
0.0	RP	3.8	3.8	3.7	3.4	3.1	3.0	3.0	2.9	2.9	3.3
	MAD	14.6	14.6	14.1	13.7	13.2	13.0	12.6	12.7	12.6	14.0
	AD	7.2	6.1	6.2	6.6	6.5	6.3	6.1	6.0	5.8	5.6
	VH	12.7	13.8	13.3	13.1	12.7	12.8	12.3	12.5	12.4	14.9
0.2	RP	4.1	4.1	3.5	3.9	3.8	3.8	3.7	3.6	3.2	3.1
	MAD	14.8	14.3	13.7	14.4	14.2	14.0	13.7	13.7	13.2	13.1
	AD	7.2	7.0	6.6	6.2	5.9	5.7	6.9	6.6	6.5	6.2
	VH	12.9	13.0	12.6	14.7	14.2	13.5	13.5	13.7	13.3	13.1
0.4	RP	4.9	4.1	4.3	4.1	3.6	3.5	3.4	3.9	3.8	3.8
	MAD	15.7	14.2	14.4	14.0	13.8	13.3	12.8	14.1	14.1	13.8
	AD	7.4	6.8	6.9	6.8	6.8	6.2	6.3	5.9	5.8	5.7
	VH	13.5	12.7	14.1	13.7	13.5	13.0	12.8	14.7	14.9	14.6
0.6	RP	5.4	4.8	4.7	4.2	4.4	4.3	4.2	4.1	3.8	3.6
	MAD	15.8	14.4	14.7	14.0	14.8	14.4	14.4	13.9	13.6	13.5
	AD	7.9	7.1	7.2	6.2	6.1	6.4	6.8	6.5	6.5	6.3
	VH	13.8	13.0	14.0	13.2	15.0	14.8	14.6	14.0	13.7	13.8
0.8	RP	7.9	6.0	5.4	4.8	5.0	4.8	4.7	4.7	4.5	4.2
	MAD	18.7	16.7	14.8	14.1	14.7	15.2	14.7	14.4	14.0	13.7
	AD	9.4	8.5	7.1	7.2	6.9	7.1	6.9	7.1	6.4	6.0
	VH	15.1	14.5	13.9	13.4	14.5	14.9	14.6	14.9	14.0	14.0

In terms of the number of nonzero operations, we have found that VH-Algo is totally outperformed by the other three algorithms for any pair of (N, ρ). Hence, we omit VH-Algo from the further detailed analysis.

In Figures 5.4 through 5.6, the ratios of the number of nonzero operations for RP-Algo versus that for MAD-Algo are plotted as a function of N for $\rho = 0.0$ through $\rho = 0.8$, for additions, multiplications and divisions, respectively. For additions and multiplications, RP-Algo dominates MAD-Algo uniformly in such a way that this efficiency gap increases as the state space size increases or the sparsity level decreases. For $N = 1000$, RP-Algo outperforms MAD-Algo roughly by a factor of three for $\rho = 0.0$ and by a factor of two for $\rho = 0.8$. For divisions, one observes a similar trend in the performance of RP-Algo versus MAD-Algo. While for $N = 100$, the performance of the two algorithms is comparable, RP-Algo dominates MAD-Algo uniformly for all other chains. In particular, the ratio is dominated by 0.08 for $N = 1000$.

Figures 5.7 through 5.9 illustrate similar computational results comparing RP- Algo with AD-Algo. As before, a general trend of RP-Algo performing better as N increases or as the sparsity level decreases can be observed. For $N > 500$ and $0.0 \leq \rho \leq 0.4$, RP-Algo outperforms AD-Algo with a 20% improvement. This difference is larger for divisions with RP-Algo dominating AD-Algo with an improvement of approximately 80% in the same parameter region.

In order to get some feeling about the actual savings, the absolute numbers of nonzero operations are summarized for $N = 100$ and $N = 1000$ in Table 5.10.

Figure 5.4: Number of Additions: Ratio RP-Algo/MAD-Algo

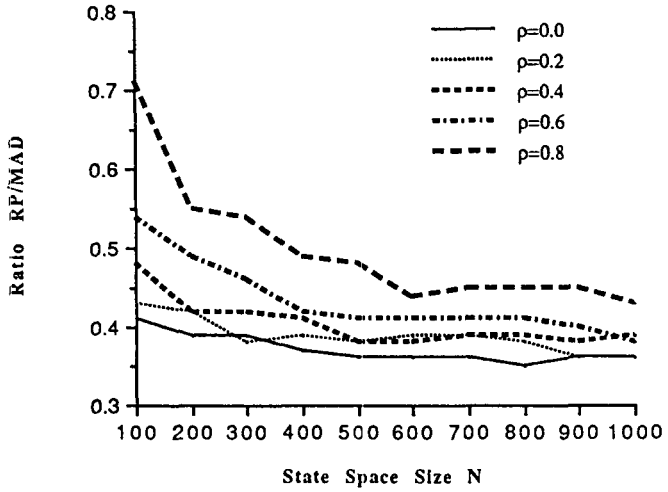


Figure 5.5: Number of Multiplications: Ratio RP-Algo/MAD-Algo

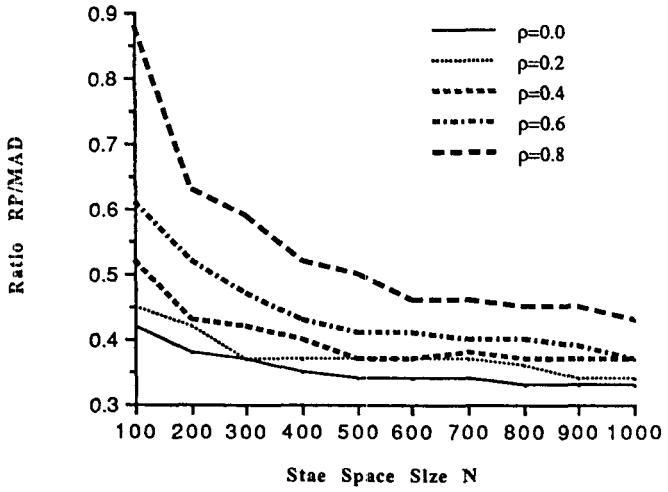


Figure 5.6: Number of Divisions:
Ratio RP-Algo/MAD-Algo

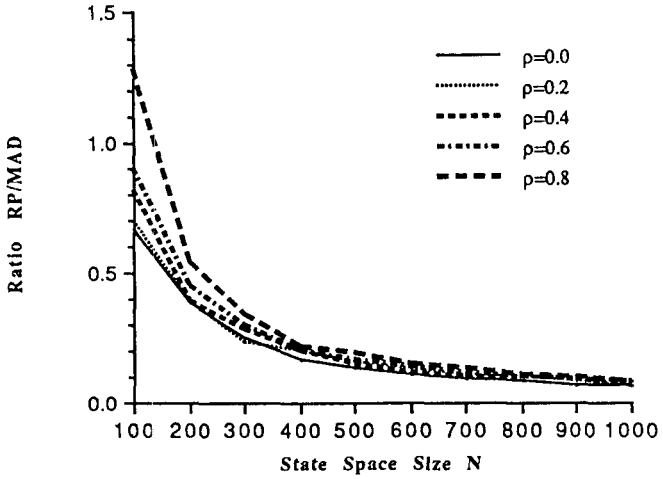


Figure 5.7: Number of Additions:
Ratio RP-Algo/AD-Algo

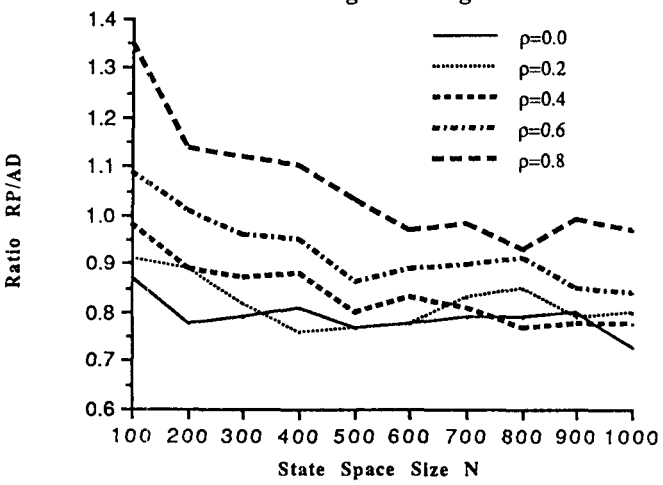


Figure 5.8: Number of Multiplications: Ratio RP-Algo/AD-Algo

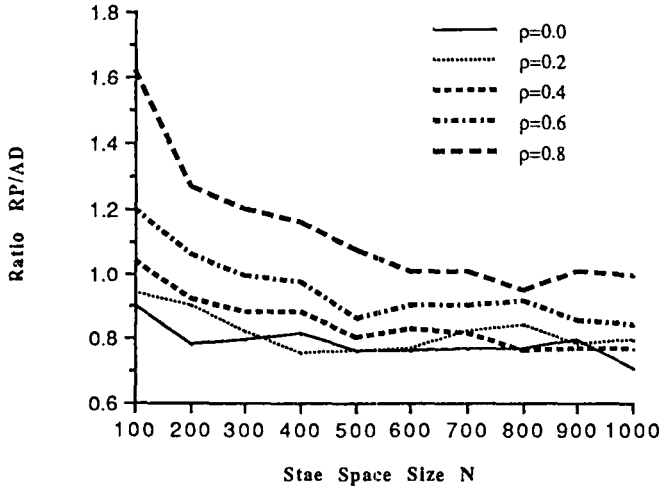


Figure 5.9: Number of Divisions: Ratio RP-Algo/AD-Algo

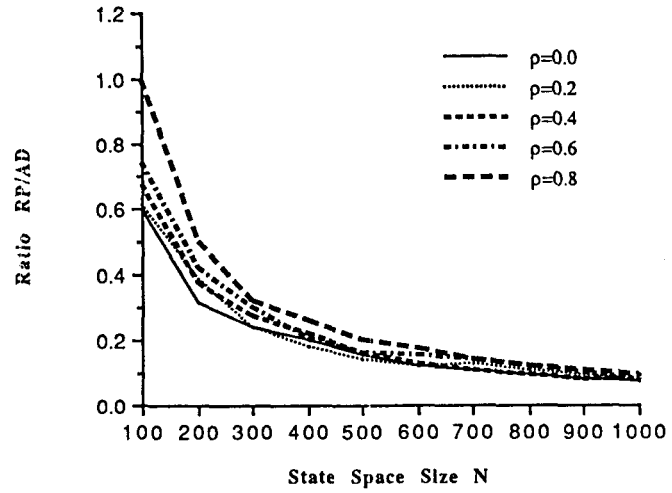


Table 5.10: Absolute Numbers of Nonzero Operations

ρ	0	0.4	0.8
N=100			
Additions (given in thousands)			
RP-Algo	45.9	47.5	59.7
MAD-Algo	111.4	99.5	84.6
AD-Algo	52.7	48.4	44.3
Multiplications (given in thousands)			
RP-Algo	42.3	44.7	57.3
MAD-Algo	100.2	86.5	65.0
AD-Algo	47.2	42.8	35.4
Divisions			
RP-Algo	467.7	690.7	1595.7
MAD-Algo	703.0	847.0	1243.0
AD-Algo	775.0	1033.3	1155.0
N=1000			
Additions (given in thousands)			
RP-Algo	2480.6	1951.4	1044.4
MAD-Algo	6938.3	5007.2	2422.0
AD-Algo	3307.8	2488.4	1071.2
Multiplications (given in thousands)			
RP-Algo	2195.8	1774.6	977.1
MAD-Algo	6603.2	4770.6	2271.6
AD-Algo	3011.2	2296.5	986.2
Divisions			
RP-Algo	303.0	397.0	581.0
MAD-Algo	4423.0	5095.0	7111.0
AD-Algo	4220.0	5275.0	6330.0

6. Concluding Remarks

The new algorithm developed in this paper can be applied for computing the ergodic probability vector of row-continuous Markov chains in continuous time via the uniformization procedure of Keilson [7] and the ergodic probability vector of row-continuous semi-Markov processes via embedded Markov chains, see e. g. Ross [15].

When successive substitution is employed for solving systems of linear equations, extensive numerical experiments on randomly generated transition probability matrices suggest that both RP-Algo and AD-Algo dominate MAD-Algo and VH-Algo uniformly. Furthermore, RP-Algo outperforms AD-Algo as both state space size and density of the underlying matrices increase.

These numerical experiments are still far from being complete and further numerical study is needed in order to acquire better understanding of the performance characteristics of the algorithms. Various strategic issues such as overrelaxation and dynamic lumping discussed in Schweitzer [17] and Seelen [19] can also be incorporated in the replacement process algorithm. A comparison of the replacement process algorithm modified in such a manner with Seelen's algorithm will be of particular interest. Furthermore, it will be interesting to test these algorithms for substantially larger Markov chains with special structure, e.g. Ph/Ph/c queueing systems and tandem queueing systems. These studies are in progress and will be reported elsewhere.

Acknowledgement

The authors wish to thank Yukio Takahashi, Paul Schweitzer, the editor and the anonymous referees for stimulating discussions and many helpful comments which have deepened the understanding of the algorithms involved and improved the presentation of the paper substantially.

REFERENCES

[1] Courtois, P.J. amd P. Semal (1984),"Bounds for the Positive Eigenvectors of Nonnegative Matrices and for Their Approximation by Decomposition", *J. Ass. Comp. Mach.*, Vol. 31, No. 4, 804–825.

[2] Hajek, B. (1982),"Birth and Death Processes on the Integers with Phases and General Boundaries", *J. Applied Probability*, 19, 488–499.

[3] Haviv, M. (1986),"An Approximation to the Stationary Distribution of a Nearly Completely Decomposable Markov Chain and Its Error Analysis", *SIAM J. Alg. Disc. Meth.*, Vol. 7, No. 4, 589–593.

[4] Haviv, M. (1987),"Aggregation/Disaggregation Methods for Computing the Stationary Distribution of a Markov Chain", *SIAM J. Numer. Analysis*, Vol. 24,No. 4, 952–966 .

[5] Haviv, M. and Y. Ritov (1986),"An Approximation to the Stationary Distribution of a Nearly Completely Decomposable Markov Chain and Its Error Bounds", *SIAM J. Alg. Disc. Meth.*, Vol. 7, No. 4, 583–588.

[6] Haviv, M. and L. Van der Heyden (1984),"Perturbation Bounds for the Stationary Probabilitites of a Finite Markov Chain", *Advances in Applied Probability*, Vol. 16, 804–818.

[7] Keilson, J. (1979), *Markov Chain Models— Rarity and Exponentiality*, Springer-Verlag, New York.

[8] Keilson, J., U. Sumita and M. Zachmann (1987),"Row Continuous Finite Markov Chains – Structure and Algorithms", *J. Oper. Res. Society of Japan*, Vol. 30, No. 3, 291–314.

[9] Keilson, J. and M. Zachmann (1988),"Homogeneous Row-Continuous Bivariate Markov Chains with Boundaries", *J. of Appl. Prob.*, Special Vol. No. 25a, 237–256.

[10] Kemeny, J.G., L.J. Snell and A.W. Knapp (1966), *Denumerable Markov Chains*, Van Nostrand, New York.

[11] Latouche, D. M., P. A. Jacobs and D. P. Gaver (1984),"Finite Markov Chain Models Skip-Free in One Direction", *Naval Res. Logistics Quarterly*, 31, 571–588.

[12] Meyer, Carl D. (1989),"Stochastic Complementation, Uncoupling Markov Chains, and the Theory of Nearly Reducible Systems", *SIAM Review*, 31, 240–272.

[13] Neuts, M. F. (1981), *Matrix-Geometric Solutions in Stochastic Models — An Algorithmic Approach*, The Johns Hopkins University Press, Baltimore, MD.

[14] Ramaswami, V. and D. M. Lucantoni (1984),"Algorithms for the Multiserver Queue with Phase Type Service", Technical Report, Bell Comm. Res., Inc., and AT&T Bell Labs.

[15] Ross, S. M. (1982), *Stochastic Processes*, Wiley, New York.

[16] Schweitzer, P. (1984),"Aggregation Methods for Large Markov Chains", Mathematical Computer Performance and Reliability, G. Iazeolla, P. J. Courtois, A. Hordijk, eds. , Elsevier North Holland, Amsterdam, 275–286.

[17] Schweitzer, P. (1990),"A Survey of Aggregation-Disaggregation in Large Markov Chains", *Proceedings of the First International Conference on the Numerical Solution of Markov Chains*, January 8–10, North Carolina State University, Raleigh, North Carolina, 57–80.

[18] Schweitzer, P., M. Puterman and K. Kindle (1985),"Iterative Aggregation-Disaggregation Procedures for Discounted Semi-Markov Reward Processes", *Opns. Research*, 33, 589–605.

[19] Seelen, L.P. (1986),"An Algorithm for Ph/Ph/c Queues", *European J. of Operat. Res.*, 23, 118–127.

[20] Sumita, U. and M. Rieders (1989),"Lumpability and Time Reversibility in the Aggregation-Disaggregation Method for Large Markov Chains", *Stochastic Models*, Vol. 5, 63–81.

[21] Sumita, U. and M. Rieders (1990),"A New Algorithm for Computing the Ergodic Probability Vector for Large Markov Chains: Replacement Process Approach", *Probability in the Engineering and Information Sciences*, Vol. 4, 89–116.

[22] Sumita, U. and M. Rieders (1990),"Numerical Comparison of the Replacement Process Approach with the Aggregation- Disaggregation Algorithm for Row-Continuous Markov Chains", *Proceedings of the First International Conference on the Numerical Solution of Markov Chains*, January 8–10, North Carolina State University, Raleigh, North Carolina, 309–327.

[23] Takahashi, Y. (1975),"A Lumping Method for Numerical Calculations of Stationary Distributions of Markov Chains", Res. Rep. No. B-18, Dept. of Information Sciences, Tokyo Institute of Technology.

[24] Takahashi, Y. (1988), private communication.

[25] Takahashi, Y. and Takami, Y. (1976),"A Numerical Method for the Steady-State Probabilities of a GI/G/c Queueing System in a General Case", *J. Oper. Res. Soc. Japan*, 19(2), 147–157.

Ushio Sumita

Graduate School of International Management
The International University of Japan
Niigata, Japan

and

William E. Simon Graduate School
of Business Administration
University of Rochester
Rochester, New York 14627