JKAU: Eng. Sci., **vol. 6**, pp. 115-135 (1414 A.H./1994 A.D.)

Aggregation/Disaggregation Methods for Computing the Stationary Distribution of Markov Chains with Application to Multiprogramming System

RABAH W. ALDHAHERI
Department of Electrical and Computer Engineering,
Faculty of Engineering, King Abdulaziz University,
Jeddah, Saudi Arabia

ABSTRACT. This paper studies the aggregation/disaggregation of nearly completely decomposable Markov chains that have many applications in queueing networks and packet switched networks. A general class of similarity transformation that transforms the stochastic transition probability matrix into a reduced order aggregated matrix is presented. This transformation is used to develop an aggregation algorithm to compute the exact stationary probability distribution, as well as $0(\varepsilon^k)$ approximation of it. The proposed aggregation method is applied to a multiprogramming computer system with six active terminals and the capacity of the CPU and the secondary memory is 3. This example is used to compare our algorithm with three well-known algorithms. The simulation studies showed that our algorithm usually converges in less number of iterations and CPU time. Moreover, it is shown that the other algorithms do not converge in some cases while our algorithm usually converges.

1. Introduction

Finite Markov chain has many applications in so many engineering problems. Queueing networks which are extensively used for modeling computer systems and evaluating their performance are one of the applications which received a great deal of attention of the researchers[1-18]for the past decades. The problem arises in these systems is determining the busy/idle period of the servers and the probability distribution of the queue length. This information can be expressed in terms of the

stationary probability distribution (s.p.d.) vector of a finite Markov chain representing the queueing model.

Another important application which can be modeled as a Markovian queueing system is the packet switched network[9-10]. The fundamental problem in this application is again, to compute the stationary probability distribution of homogeneous sytem of linear equations.

In these applications where this stationary probability distribution plays an important role, the number of states in the Markov chain can be huge, *i.e.*, much more than 10,000. Fortunately, in such large matrices the states can be clustered into small groups such that there is a strong interaction within each group while the interaction among the groups is weak compared to the interaction within the groups. This class of Markov chain is known in the literature as nearly completely decomposable Markov chain. Simon and Ando[11] were the first to propose this class of systems and they gave examples from economics to illustrate this class of systems. Since the seventies, there have been several studies that contributed to the development of decomposition-aggregation method that exploits the strong-weak interaction structure to obtain reduced-order approximation. Courtois[1,4] developed the theory of the nearly completely decomposable Markov chain and introduced the technique of aggregation in queueing network analysis and computer system performance evaluation[1-3]. Courtois developed an aggregation procedure that produce an $0(\varepsilon)$ approximation to the stationary probability distribution, where ε is a small parameter that represents the intergroup interactions. This procedure has been extended to obtain an $0(\varepsilon^2)$ approximations[7]. Other numerical techniques which combine block decomposition and iterative methods, like block Gauss-Seidel or block Jacobi methods have been reported in references[8, 18, 19, 21]. Phillips and Kokotovic[12] gave a singular perturbation interpretation to Courtois' aggregation. They developed a similarity transformation that transforms the system into a singularly perturbed form, whose slow model coincides with Courtois' aggragated matrix. The use of singular perturbation in aggragation of finite state Markov chains has been pursued further in references[13-15].

The paper is organized as follows: In the next section, we review briefly the nearly completely decomposable Markov chains and introduce some of the notations and conventions. In Section 3, a general class of transformation that transforms the nearly completely decomposable Markov chains into a reduced-order aggregated matrix is given. This transformation is more general than the one considered in reference6[12]. It is shown that within this transformation we can choose a transformation that is independent of the system parameters. This is very important when we develop our iterative algorithm in Section 4 because in this case we do not need to form the aggregated matrix at each iteration. In this section also, we use the above transformation to develop an aggregation method to compute the exact stationary distribution from a reduced order aggregated system. Properties of the aggregated system is discussed and the uniqueness of the solution is derived. Moreover, it is shown that all the transformations that satisfy the conditions of Section 3 produce the same

$0(\varepsilon)$ approximation of the aggregated matrix. In Section 4, a class of transformation is given to reduce the computational effort. This class of transformation will be used to develop an iterative algorithm to compute the aggregated matrix. It is shown that stopping the iteration after k steps gives an $0(\varepsilon^{k+1})$ approximation of the stationary probability distribution. Finally, in Section 5 we use the results of Sections 3 and 4 to obtain an exact solution for the stationary behaviour of the multiprogramming system. This example is used to compare our iterative method with the methods considered in references[8, 18, 19].

2. Nearly Completely Decomposable Markov Chains

The nearly completely decomposable Markov chain is a class of finite Markov chains in which the transition probability matrix (t.p.m.) $P \in \Re^{n \times n}$ takes the form

$$P = I_n + A + \varepsilon B \tag{2.1}$$

where I_n is the nth order identity matrix, ε is the maximum degree of coupling and

$$\mathrm{A} = \begin{bmatrix} A_{11} & & & \\ & A_{22} & & \\ & & \ddots & \\ & & & A_{NN} \end{bmatrix} \tag{2.2}$$

The diagonal blocks A_{ii} are square and of order n_i, $i = 1, 2, \cdots, N$.

Therefore

$$\sum_{i=1}^{N} n_i = n$$

The matrices P and $(I + A_{ii})$, $i = 1, 2, \cdots, N$ are stochastic. Hence the row sums of A and B are zero. Notice here that there is an uncountable set of block diagonal stochastic matrices A which makes the row sums of A and B zero. All these choices give the same stationary probability distribution π which will be defined shortly. It is assumed that the Markov chain has a single ergodic class. Thus P has a unique unity eigenvalue and there is a unique positive left eigenventor π, which satisfies

$$\pi P = \pi, \ \pi_i > 0, \ \pi e_n = 1 \tag{2.3}$$

where e_n is a column vector consisting of n ones. The row vector π is called the stationary probability distribution or the Perron-Frobenius eigenvector (P-F eigenvector). It is also assumed that each of the matrices $(I + A_{ii})$ is irreducible. Hence $(I + A_{ii})$ has a unique unity eigenvalue. For samll ε, the matrix $A + \varepsilon B$ has one zero eigenvalue and N-1 eigenvalues which are very close to zero. This might cause ill-conditioning problem in solving (2.3). Moreover, the convergence of the standard iterative methods for solving (2.3) will be very bad.

Our first objective in this paper is to find a transformation that reduce the higher order n-dimensional Markov chain system to a lower order N-dimensional system where $N << n$.

This transformation makes use of the two-time scale property inherited in this model[12,20]. The transformation decomposes the system into slow and fast parts and as a result of this, the high and ill-conditioned Markov chain system is reduced to a low order and well-conditioned one.

In the following sections the transformation that aggregates the Markov chain system is proposed. Moreover, a special class of "block diagonal transformation", is given to simplify and reduce the amount of computations required to form the aggregated matrix. This transformation depends only on the dimensions of the subsystems. It is independent of the system parameters. This special class of transformation will be used to develop a new algorithm to compute the exact and the $0(\varepsilon^k)$ approximation of the stationary probability distribution vector π.

3. Exact Aggregation of Nearly Completely Decomposable Markov Chains

In this section, we propose a transformation that is more general than the transformation considered in reference[12]. The transformation can be described as follows :

Partition the transformation Γ such that

$$\Gamma = [\, W_1 \;\; W_2 \,] \tag{3.1}$$

where $W_1 \in \Re^{n \times N}$ and $W_2 \in \Re^{n \times (n-N)}$

2. Choose W_1 such that $AW_1 = 0$. Since A is block diagonal, W_1 can be chosen as $W_1 = \text{diag}\,[\, e_{n_1}, e_{n_2}, \cdots, e_{n_N} \,]$, where e_{n_i}, $i = 1, \cdots N$ is a column vector of n_i *ones*. W_2 can be any matrix such that the transformation Γ is nonsingular. The matrix Γ^{-1} is written as

$$\Gamma^{-1} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \tag{3.2}$$

where $V_1 \in \Re^{N \times n}$ and $V_2 \in \Re^{(n-N) \times n}$. Therefore, $V_1 W_1 = I_N$, $V_2 W_2 = I_{n-N}$, $V_1 W_2 = 0$ and $V_2 W_1 = 0$.

Apply this transformation to equation (2.3). Let $\beta = \pi W_1$ and $\gamma = \pi W_2$. Then, equation (2.3) becomes

$$\pi\, [W_1 \;\; W_2] \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} P = \pi \tag{3.3}$$

Multiply (3.3) from the right by Γ to obtain

$$[\beta\ \gamma]\begin{bmatrix} I_N + \varepsilon V_1 B W_1 & V_1 (A + \varepsilon B) W_2 \\ \varepsilon V_2 B W_1 & V_2 (A + \varepsilon B)\ W_2 + I_{n-N} \end{bmatrix} = [\beta\ \gamma] \quad (3.4)$$

Notice here, because A has N zero eigenvalues, so

$$\Gamma^{-1} A \Gamma = \begin{bmatrix} 0 & V_1 A W_2 \\ 0 & V_2 A W_2 \end{bmatrix}$$

Where the upper left corner of the above matrix is $N \times N$ zero matrix. Since the eigenvalues are invariant under the similarity transformation, therefore, $V_2 A W_2$ is nonsingular. This implies that for ε sufficiently small, $V_2 (A + \varepsilon B)\ W_2$ is nonsingular. From (3.4) we have

$$\beta (I_N + \varepsilon V_1 B W_1) + \varepsilon \gamma V_2 B W_1 = \beta \quad (3.5)$$

and

$$\beta V_1 (A + \varepsilon B)\ W_2 + \gamma (V_2 (A + \varepsilon B)\ W_2 + I_{n-N}) = \gamma \quad (3.6)$$

Therefore

$$\gamma = -\beta V_1 (A + \varepsilon B)\ W_2 [V_2 (A + \varepsilon B)\ W_2]^{-1} \quad (3.7)$$

Substitute (3.7) into (3.5) to obtain

$$\beta \gg I_N + \varepsilon (V_1 B W_1 - V_1 (A + \varepsilon B)\ W_2 (V_2 (A + \varepsilon B)\ W_2)^{-1} V_2 B W_1)\ ] = \beta \quad (3.8)$$

Let

$$Q_s = [V_1 - V_1 (A + \varepsilon B)\ W_2 (V_2 (A + \varepsilon B)\ W_2)^{-1} V_2]\ B W_1 = V_s B W_1 \quad (3.9)$$

and

$$P_{ex} = I_N + \varepsilon Q_s \quad (3.10)$$

where

$$V_s = [V_1 - V_1 (A + \varepsilon B)\ W_2 (V_2 (A + \varepsilon B)\ W_2)^{-1} V_2] \quad (3.11)$$

Then

$$\beta P_{ex} = \beta. \quad (3.12)$$

or simply

$$\beta Q_s = 0 \tag{3.13}$$

Again from (2.3), $\pi e_n = 1$, therefore,

$$\pi \begin{bmatrix} W_1 & W_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} e_n = 1 \tag{3.14}$$

Hence

$$\beta e_N = 1 \tag{3.15}$$

Where e_N is a column vector of N ones. Thus, the solution of the full order system (2.3) is completely characterized by

$$\pi = \beta V_1 + \gamma V_2 = \beta V_s \tag{3.16}$$

where β is defined by (3.13) and (3.15), and γ is given by (3.7). Therefore, the high-order n-dimensional system is reduced to a lower-order N-dimensional system. It is shown in reference [20] that the solution of (3.13) and (3.15) is well-conditioned no matter how small ε is.

The existence of a unique solution of (3.13) and (3.15) is discussed in Section 3.1. In Section 3.2, we propose a class of transformation that reduces the amount of computations required to form the aggregated matrix. Before discussing the existence of a unique solution to (3.13) and (3.15), let us raise this question: Is the aggregated matrix P_{ex} is a stochastic matrix? The answer is no, in general. The following example is a counter example which shows that the matrix P_{ex} may not be stochastic.

Example 3.1

Consider the irreducible transition probability matrix

$$P = \begin{bmatrix} 0.6 & 0.4-\varepsilon & 0 & 0 & 0 & \varepsilon \\ 0.6-\varepsilon & 0.4-\varepsilon & 0 & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 0.5-\varepsilon & 0.5-\varepsilon & 0 & 0 \\ 0 & \varepsilon & 0.5 & 0.5-\varepsilon & 0 & 0 \\ 0.5\varepsilon & 0.5\varepsilon & 0.5\varepsilon & 0.5\varepsilon & 0.7-\varepsilon & 0.3-\varepsilon \\ 0 & 0.5\varepsilon & 0.0 & 0.5\varepsilon & 0.7 & 0.3-\varepsilon \end{bmatrix}$$

From (2.1) and (2.2), the matrices A and B are

$$A = \begin{bmatrix} -0.4 & 0.4 & 0 & 0 & 0 & 0 \\ 0.6 & -0.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.3 & 0.3 \\ 0 & 0 & 0 & 0 & 0.7 & -0.7 \end{bmatrix} \quad \text{and } B = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 1 \\ -1 & -1 & 0 & 0 & 1 & 1 \\ 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0.5 & 0.5 & 0.5 & 0.5 & -1 & -1 \\ 0 & 0.5 & 0 & 0.5 & 0 & -1 \end{bmatrix}$$

Let Γ be equal to

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

From (3.9)

$$Q_s = \begin{bmatrix} -1.4 + 0.5\varepsilon & -0.5\varepsilon & 1.4 \\ 1.5 & -1.5 - \varepsilon & \varepsilon \\ 0.85 - 0.5\varepsilon & 0.85 + \varepsilon & -1.7 - 0.5\varepsilon \end{bmatrix}$$

It is clear that the matrix $P_{ex} = I_N + \varepsilon Q_s$ is not a stochastic matrix because of the negative sign of the (12) element.

Since P_{ex} is not necessary stochastic, the existence of a unique solution of (3.13) and (3.15) should be shown. In the following section the properties of Q_s is established and from which the uniqueness of the solution can be concluded.

3.1 Properties of the aggregated matrix

The aggragated matrix Q_s is a reduced form of the original matrix $(A + \varepsilon B)$ and it inherits some of its properties. In this section some properties of Q_s are given in the form of theorems.

Theorem 3.1

The row sums of the matrix Q_s are equal to zero.

Proof

To prove this theorem it is enough to show that $Q_s e_N = 0$.

Multiply (3.9) from the right by e_N to obtain

$$Q_s e_N = V_s B W_1 e_N = V_s B e_n$$

Recall from the definition of B that each row sum of B equal to zero. Therefore,

$$Q_s e_N = V_s B W_1 e_N = V_s B e_n = 0$$

and this completes the proof. As a result of this theorem, the matrix Q_s has a zero eigenvalue. Another useful property of Q_s is given in the following theorem.

Theorem 3.2

Let the matrix Q_s be defined as in (3.9) and ψ_{s_i} be defined as the matrix Q_s with its ith column replaced by the vector e_N . Then

a) The matrix Q_s has a unique zero eigenvalue.
b) The matrix ψ_{s_i} is nonsingular, for $i = 1, 2, \cdots N$.
c) Equations (3.13) and (3.15) have a unique solution and $\beta_i > = 1, 2, \cdots, N$.

The proof of this theorem is given in reference [20]. Here, we will give a simpler proof to part (a) only.

a) From the ergodicity assumption.

$$\text{rank}\,(A + \varepsilon B) = n - 1 = \text{rank}\,(\Gamma^{-1}(A + \varepsilon B)\,\Gamma) = \text{rank}\left\{\begin{bmatrix} \varepsilon V_1 B W_1 & V_1 (A + \varepsilon B) W_2 \\ \varepsilon V_2 B W_1 & V_2 (A + \varepsilon B) W_2 \end{bmatrix}\right\} =$$

$$\text{rank}\left\{\begin{bmatrix} \varepsilon Q_s & V_1 (A + \varepsilon B) W_2 \\ 0 & V_2 (A + \varepsilon B) W_2 \end{bmatrix}\begin{bmatrix} I_N & 0 \\ (V_2 (A + \varepsilon B) W_2)^{-1} \varepsilon V_2 B W_1 & I_{n-N} \end{bmatrix}\right\} = \text{rank}\,(U.L.)$$

Since the matrix L is nonsingular,

$$n - 1 = \text{rank}\,(\varepsilon Q_s) + \text{rank}\,(V_2 (A + \varepsilon B) W_2) = \text{rank}\,(\varepsilon Q_s) + n - N$$

Therefore

$$\text{rank}\,(Q_s) = N - 1 \tag{3.17}$$

Since e_N span the null space of Q_s, (3.17) implies that Q_s has a unique zero eigenvalue.

From the above theorem, we notice that P_{ex} preserves some of the properties of P whether P_{ex} is stochastic or not, like one row sums, the uniqueness of the unity eigenvalue and the property that the left eigenvector corresponding to the unity eigenvalue is unique, positive and its sum equals 1. Moreover, although P_{ex} is not unique, depends on the choice of the similarity transformation Γ, the left eigenvector corresponding to the unity eigenvalue β is unique for all classes of transformations satisfying (3.1) since we fixed the choice of W_1.

3.2 Block Diagonal Transformation

From Section 3, it is clear that there is a wide class of transformations which give the same aggregated matrix. Moreover, any order of approximation of the P–F eigenvector π can be achieved via an expansion of the exact reduced order system, (3.9). In this section a subclass of the general transformation matrix discussed previously is given. This transformation has a block diagonal structure. The idea of the transformation is to specialize the choice of W_2 in (3.1). Since W_1 is block diagonal, we may choose W_2 to be block diagonal as well. Such a choice will result in block diagonal matrices V_1 and V_2. So, the matrices V_1, V_2, W_1 and W_2 defined in section 3 become :

$$V_1 = \begin{bmatrix} V_1^{(1)} & 0 & 0 & 0 & 0 \\ 0 & V_1^{(2)} & 0 & 0 & \cdot \\ 0 & 0 & V_1^{(3)} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & V_1^{(N)} \end{bmatrix} \tag{3.18}$$

$$V_2 = \begin{bmatrix} V_2^{(1)} & 0 & 0 & 0 & 0 \\ 0 & V_2^{(2)} & 0 & 0 & \cdot \\ 0 & 0 & V_2^{(3)} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & V_2^{(N)} \end{bmatrix} \tag{3.19}$$

$$W_1 = \begin{bmatrix} e_{n_1} & 0 & 0 & 0 & 0 \\ 0 & e_{n_2} & 0 & 0 & 0 \\ 0 & 0 & e_{n_3} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & e_{n_N} \end{bmatrix} \tag{3.20}$$

and

$$W_2 = \begin{bmatrix} W_2^{(1)} & 0 & 0 & 0 & 0 \\ 0 & W_2^{(2)} & 0 & 0 & \cdot \\ 0 & 0 & W_2^{(3)} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & W_2^{(N)} \end{bmatrix} \tag{3.21}$$

where

$V_1^{(i)} W_1^{(i)} = 1$, $V_1^{(i)} W_2^{(i)} = 0$, $V_2^{(i)} W_1^{(i)} = 0$ and $V_2^{(i)} W_2^{(i)} = I_{m_i}$. The superscript (i) denotes the ith diagonal block of the respective matrix and $m_i = n_i - 1$. In this transformation, computation of V_0 is easier and more efficient. Recall from (3.11) that :

$$V_0 = V_s|_{\varepsilon = 0} = V_1 - V_1 A W_2 (V_2 A W_2)^{-1} V_2 \tag{3.22}$$

In this case V_1, V_2, $(V_1 A W_2)$ and $(V_2 A W_2)$ are block diagonal. Each diagonal block of the matrix $V_2 A W_2$ is equal to $(V_2^{(i)} A_{ii} W_2^{(i)}) \in \Re^{m_i \times m_i}$. This simplifies the matrix multiplications and inversion involved in computing V_0. The transformation given in reference [12] belongs to this class of transformations. In reference [22] it is shown that V_0 is a closed form expression for the Perron-Frobenius eigenvectors for the stochastic matrices $(A + I_n)$. Moreover, it is shown that all the transformations that satisfy (3.1) produce the same V_0, *i.e.*, the same $0(\varepsilon)$ aggregated matrix.

We now propose a specific choice of $W_2^{(i)}$ that renders a particular simple transformation. This choice is the same as the one considered in reference [22] and it proceeds as follows :

For A_{ii}, $i = 1, 2, \cdots, N$, let $W_1^{(i)} \in \Re^{n_i \times 1}$ and $W_2^{(i)} \in \Re^{n_i \times m_i}$ be

$$W_1^{(i)} = e_{n_i} \text{ and } W_2^{(i)} = \begin{bmatrix} I_{m_i} \\ 0_{m_i} \end{bmatrix} \tag{3.23}$$

where 0_{m_i} is a row vector consisting of m_i zeros.

For this choice, $V_1^{(i)} \in \Re^{1 \times n_i}$ and $V_2^{(i)} \in \Re^{m_i \times n_i}$ can be computed to be

$$V_1^{(i)} = [\, 0, 0, \cdots, 1 \,] \tag{3.24}$$

and

$$V_2^{(i)} = [\, I_{m_i} \quad -e_{m_i} \,] \tag{3.25}$$

Our choice of transformation (3.23) – (3.25) is much simpler than the one proposed in reference [12] because it is independent of the system parameters. This is very important when we develop our algorithm, because we don't need to form the aggregated matrix in each iteration. Moreover, the transformation $\Gamma^{-1} A \Gamma$ can be obtained by inspection.

4. An Iterative Algorithm for Computing the Exact S.P.D.

From Section 3.2, we notice that choosing the block diagonal transformation (3.23) – (3.25) will reduce the amount of computations required to form the aggregated matrix. Moreover, when this transformation is used, the matrix $(V_2 (A + \varepsilon B) W_2)$ approaches a block diagonal matrix as ε tends to zero. This suggests that one should exploit the closeness to a block diagonal matrix to simplify the inversion of $(V_2 (A + \varepsilon B) W_2)$. In this section we can employ this fact in developing an iterative algorithm to compute the exact stationary probability distribution (S.P.D.), as well as $0(\varepsilon^k)$ approximation of it. This algorithm is based on computing $(V_2 (A + \varepsilon B) W_2)^{-1}$ iteratively. It is shown in reference [22] that V_s can be written as

$$V_s = V_0 - \varepsilon V_0 B W_2 (V_2 (A + \varepsilon B) W_2)^{-1} V_2 \tag{4.1}$$

Substituting (4.1) into (3.9), we get

$$Q_s = V_0 B W_1 - \varepsilon V_0 B W_2 (V_2 (A + \varepsilon B) W_2)^{-1} V_2 B W_1 \tag{4.2}$$

The procedure of this algorithm is described as follows :

Let $S \overset{\Delta}{=} V_2 A W_2$, $R \overset{\Delta}{=} V_2 B W_2$ and

$$Z = (V_2 (A + \varepsilon B) W_2)^{-1} = (S + \varepsilon R)^{-1} \tag{4.3}$$

where S is a block diagonal nonsingular matrix. Multiplying (4.3) from the right by $(S + \varepsilon R)$ to obtain

$$Z = S^{-1} - \varepsilon Z R S^{-1} \overset{\Delta}{=} f(Z) \tag{4.4}$$

where f maps Z into Z.

The solution of this equation can be obtained via successive approximation if $f(\cdot)$ is a contraction mapping. We have

$$f(Z_1) - f(Z_2) = \varepsilon [Z_2 - Z_1] Y$$

where $Y \overset{\Delta}{=} RS^{-1}$ and $Z_1, Z_2 \in Z$. *Thus*

$$\| f(Z_1) - f(Z_2) \| \leq \varepsilon \| Z_2 - Z_1 \| \| Y \|$$

Since A, B, V_2 and W_2 are $0(1)$, $\| Y \| = 0(1)$. This implies that $\varepsilon \| Y \| < 1$ for sufficiently small ε. Therefore, f is a contraction mapping. Thus equation (4.4) has a unique solution which gives $f(Z) = Z$.

The successive approximation for computing $(A_2\,(A + \varepsilon B)\,W_2)^{-1}$ proceeds as follows.

- Let $Z^{(-1)} = 0$
- For $k = 0, 1, 2, \cdots$

$$Z^{(k)} = S^{-1} - \varepsilon Z^{(k-1)}\,Y \tag{4.5}$$

- As $k \to \infty$, $Z^{(k)} \to (V_2\,(A + \varepsilon B)\,W_2)^{-1}$. Moreover,

$$\| Z - Z^{(k)} \| = 0(\varepsilon^{k+1}) \tag{4.6}$$

To show (4.6), substract (4.5) from (4.4) to get

$$Z - Z^{(k)} = \varepsilon\,[Z^{(k-1)} - Z]\,Y$$

Now, for $k = 0$, $Z - Z^{(0)} = -\varepsilon ZY = 0(\varepsilon)$ and for $k = 1$,

$$Z - Z^{(1)} = \varepsilon\,[Z^{(0)} - Z]\,Y = \varepsilon^2 ZY^2 = 0(\varepsilon^2)$$

and by induction we can show that (4.6) is true.

Notice that the only matrix inversion required in this process is computing the inverse of the block diagonal matrix S.

Replacing $(V_2\,(A + \varepsilon B)\,W_2)^{-1}$ in (4.1) by $Z^{(k-1)}$ of (4.5) yields

$$V_k = V_0 - \varepsilon V_0 B W_2\,Z^{(k-1)}\,V_2\,,\ k \geq 1 \tag{4.7}$$

The $(k+1)$th order approximation of Q_s, $k \geq 0$ is given by

$$\begin{aligned} Q_0 &= V_0 B W_1 \\ Q_k &= V_k B W_1\,,\ k \geq 1 \end{aligned} \tag{4.8}$$

where V_0 and V_k are defined by (3.22) and (4.7) respectively. Hence

$$Q_s = Q_k + 0\,(\varepsilon^{k+1}) \tag{4.9}$$

Approximating Q_s in (3.9) by Q_k yields

$$\beta_k Q_k = 0\,,\ \beta_k e_N = 1 \tag{4.10}$$

Let

$$P_{k+1} = I_N + \varepsilon Q_k\,,\ k \geq 0 \tag{4.11}$$

then equation (4.10) can be rewritten as

$$\beta_k P_{k+1} = \beta_k\,,\ \beta_k e_N = 1 \tag{4.12}$$

Theorem 4.1

Let P_{k+1} be defined as in equation (4.11), then :

a) P_{k+1} is an $0(\varepsilon^{k+2})$ approximation of P_{ex}.

b) β_k *is an* $0(\varepsilon^{k+1})$ approximation of β.

c) π_k is an $0(\varepsilon^{k+1})$ approximation of π, where,

$$\pi_k = \beta_k V_1 + \gamma_k V_2 = \beta_k V_1 - \beta_k V_1 (A + \varepsilon B) W_2 Z^{(k)} V_2 = \beta_k V_k \tag{4.13}$$

and

$$\gamma_0 = -\beta_0 V_1 A W_2 Z^{(0)}$$
$$\gamma_k = -\beta_k V_1 (A + \varepsilon B) W_2 Z^{(k)}, \; k \geq 1 \tag{4.14}$$

Proof

a) From (3.10) and (4.9)

$$P_{ex} = I_N + \varepsilon Q_s = I_N + \varepsilon [Q_k + 0(\varepsilon^{k+1})] = I_N + \varepsilon Q_k + 0(\varepsilon^{k+2})$$

b) From Theorem 3.2

$$\beta \psi_{si} = 1_i \tag{4.15}$$

where 1_i is an N-dimensional row vector with the ith element equal to 1 and the rest of the elements equal to zero, *i.e.*,

$$\beta [q_s^1, q_s^2, \cdots, q_s^{i-1}, e_N, q_s^{i+1}, \cdots, q_s^N] = [0, 0, \cdots, 0, 1, 0, \cdots, 0] \tag{4.16}$$

where q_s^i is the ith column of Q_s.

Similarly, equation (4.10) can be rewritten as

$$\beta_k \psi_{ki} = 1_i \tag{4.17}$$

where ψ_{ki} is obtained from Q_k by replacing its ith column by e_N.

From (4.9), (4.15) and (4.17) it follows that

$$\psi_{si} = \psi_{ki} + 0(\varepsilon^{k+1}) \tag{4.18}$$

Hence, ψ_{ki} is nonsingular for sufficiently small ε, which implies that (4.12) or equivalently, (4.10) has a unique solution. Therefore,

$$(\beta \partial \beta_k) \psi_{ki} = 0(\varepsilon^{k+1}) \tag{4.19}$$

Since ψ_{ki} is nonsingular,

$$\beta = \beta_k + 0(\varepsilon^{k+1}) \tag{4.20}$$

c) From (3.13), (4.7) and (4.20)

$$\pi = \beta V_s = (\beta_k + 0(\varepsilon^{k+1}))(V_k + 0(\varepsilon^{k+1})) = \beta_k V_k + 0(\varepsilon^{k+1}) = \pi_k + 0(\varepsilon^{k+1})$$

Now, the proposed iterative algorithm can be summarized as follow :

a) Compute $Z^{(k)}$ iteratively from (4.5) for any degree of accuracy you choose.

Notice that stopping the iteration after k steps results in 0 (ε^{k+1}) approximation of π.

b) Compute V_k and Q_k by using equations (4.7) and (4.8).

c) Solve for the row vector β_k which is uniquely determined by (4.10) or equivalently, (4.12).

d) Compute π_k as defined in (4.13).

e) Conduct a convergence test. If the accuracy is sufficient then stop and take π_k to be the required solution. Otherwise, go to step (a).

4.1 Simulation Results

In this section, several numerical examples are given. These examples are solved by the exact method presented in Section 3 and by the iterative algorithm presented in Section 4. Moreover, these examples are solved by the three algorithms discussed in reference[17]. The computer program for each algorithm was executed on PC/386-20. The software was written in Pascal and compiled using the Turbo Pascal compiler. The program is terminated when $\| \pi - \pi_k \|_2 \leq 10^{-7}$. If the convergence has not been reached, the program is terminated at 50 iterations.

The simulation results showed that our proposed algorithm converges more rapidly than the other algorithms. Moreover, it is shown that these algorithms do not converge in some cases while our algorithm usually converges in few number of iterations.

Example 4.1

Consider the following transition probability matrix given in reference[16]. This matrix represents a nearly completely decomposable Markov chain with three blocks, *i.e.*, $N = 3$

$$P = \begin{bmatrix} 0.434 & 0.548 & 0.010 & 0.000 & 0.006 & 0.002 \\ 0.340 & 0.645 & 0.000 & 0.005 & 0.010 & 0.000 \\ 0.004 & 0.005 & 0.219 & 0.765 & 0.007 & 0.000 \\ 0.002 & 0.000 & 0.213 & 0.785 & 0.000 & 0.000 \\ 0.008 & 0.002 & 0.000 & 0.000 & 0.667 & 0.323 \\ 0.005 & 0.001 & 0.000 & 0.010 & 0.725 & 0.259 \end{bmatrix}$$

The exact stationary probability distribution is shown in the following five columns :

1. Exact Method: is the solution by using our exact aggregation method presented in section 3.
2. Kouri: is the solution by using (K-M-S) algorithm[18].
3. Takahashi: is the solution by using Takahashi algorithm[19].
4. Vantilborh: is the solution by using Vantilborgh algorithm[8].
5. Prop. Algor.: is the solution by using our proposed algorithm discussed in Section 4.

TABLE 1.

Exact method	Kouri	Takahashi	Vantilborgh	Prop. Algor.
π Vector	π_k Vector	π_k Vector	π_k Vector	π_k Vector
0.09609287	0.09609287	0.09609287	0.09609287	0.09609287
0.15107140	0.15107140	0.15107140	0.15107140	0.15107140
0.10736545	0.10736545	0.10736545	0.10736545	0.10736545
0.38916190	0.38916190	0.38916190	0.38916190	0.38916190
0.17831986	0.17831986	0.17831986	0.17831986	0.17831986
0.07798853	0.07798853	0.07798853	0.07798853	0.07798853
CPU time is 0 s, 50 ms	CPU time is 0 s, 160 ms	CPU time is 0 s, 100 ms	CPU time is 0 s, 100 ms	CPU time is 0 s, 100 ms
	Iterations 7	Iterations 4	Iterations 4	Iterations 3

Notice from Table 1 that the four iterative methods converge to the exact solution in a few number of iterations. Our method converges in less number of iterations.

The CPU time in this example and in the next ones is an approximate and sometimes there is slight difference in the CPU time from one run to another. Also, in all the examples we considered the CPU time for our exact aggregation method is less than the one for the iterative methods. This should be expected since we deal with small size matrices. In our iterative method, part of the CPU time is required for the transformation step which is performed before applying the iterative algorithm. This made the CPU time of our algorithm in the previous example is comparable with the others although the number of iterations is less. The significance of our algorithm does not lie in the CPU time only, but, in its usual convergence. The other algorithms do not converge in some cases as we will see in the next example and in the multiprogramming system considered in Section 5.1.

Example 4.2

This example deals with evaluating computer system performance. The transition probability matrix of the page fault probability of the LRU (least recently used) replacement algorithm is given in reference[1].

Due to the size of this matrix, we refer the interested reader to the work of Courtois[1].

The S.P.D. solved by the exact exaggeration method and the iterative algorithms is given in Table 2.

Table 2 shows that Kouri, Takahashi and Vantilborgh algorithms did not converge to the exact solution even if we increase the number of iterations, while our algorithm takes only 4 iterations to converge.

TABLE 2.

Exact method	Kouri	Takahashi	Vantilborgh	Prop. Algor.
π Vector	π_k Vector	π_k Vector	π_k Vector	π_k Vector
0.07150547	0.07051917	0.07150499	0.07150499	0.07150547
0.06311665	0.06224461	0.06311652	0.06311652	0.06311665
0.05842784	0.05762605	0.05842773	0.05842773	0.05842784
0.03768036	0.03715108	0.03767996	0.03767996	0.03768036
0.07382848	0.07281037	0.07382839	0.07382839	0.07282848
0.05689172	0.05610132	0.05689156	0.05689156	0.05689172
0.06071739	0.06027951	0.06071581	0.06071581	0.06071739
0.06005172	0.05961566	0.06005032	0.06005032	0.06005172
0.06036241	0.05992708	0.06036115	0.06036115	0.06036241
0.06005154	0.05961404	0.06005003	0.06005003	0.06005154
0.06023276	0.05979708	0.06023143	0.06023143	0.06023276
0.10777089	0.10559154	0.10771570	0.10771570	0.10777089
0.07186578	0.07047838	0.07189624	0.07189624	0.07186578
0.08164878	0.08006275	0.08167336	0.08167336	0.08164878
0.07584821	0.07436089	0.07585680	0.07585680	0.07584821
CPU time is 0 s, 430 ms	CPU time is 6 s, 90 ms	CPU time is 6 s, 750 ms	CPU time is 3 s, 620 ms	CPU time is 1 s, 40 ms
	Iterations 50	Iterations 50	Iterations 50	Iterations 4

5. Multiprogramming Computer System

We have mentioned in the introduction that the nearly completely decomposable Markov chain has many applications in queueing networks. In this section we consider a queueing network model of a computer system studied by Courtois[1-3] and Vantilborgh[7].

This model represents a multiprogramming computer system. A diagram of this model is shown in Fig. 1. The system consists of

1. A set of L terminals from which L users generate random command according to Poisson process with a rate of λ.
2. A central processing unit (CPU) represented by S_1 with state dependent exponential mean service rate μ_1 (i) where $_1$ is the number of jobs in queue S_1.
3. A paged secondary memory S_2 with state dependent service rate μ_2 (i_2) where i_2 is the number of jobs in queue S_2

When a command is generated, the user at the terminal remains inactive until the system responds to that command. Command from terminal enters the multiprogramming mix if $i_1 + i_2 < M$, where M is the maximum capacity of the multiprogramming mix, otherwise it is kept waiting in queue Q_3. At the end of service at CPU it goes to secondary memory with probability δ and to Q_3 with probability φ. The state

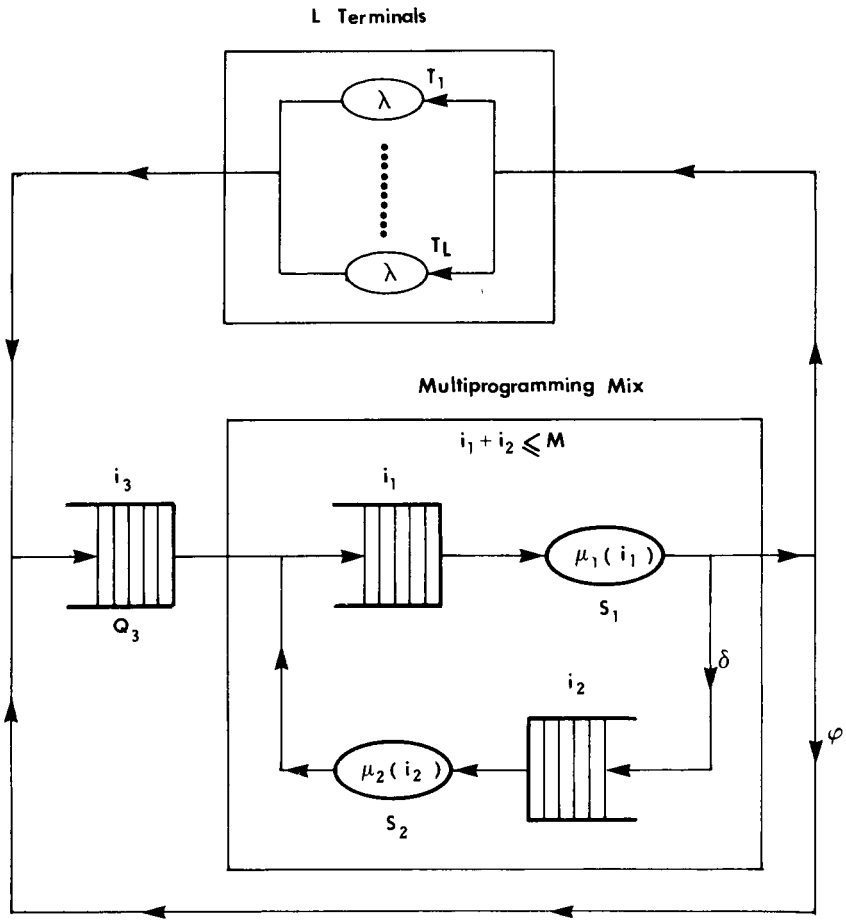

FIG. 1. Multiprogramming computer system.

of this system is uniquely defined by the triplet (i_1, i_2, i_3) and the transition probability matrix $Q(L, M)$ is formed by ordering the states as follows: First comes all the states with $i_3 = 0$. The state with constant $i_1 + i_2$ represents one group in the whole system which is represented by the stochastic matrix $Q(L, M)$. The group with smaller $i_1 + i_2$ comes first and within this group, states with smaller value of i_2 come first. Then states with $i_3 = 1, 2 \cdots, L - M$ are ordered and grouped by the same procedure. For more details of forming $Q(L, M)$, we refer the reader to the work of Vantilborgh[7]. The service rates for the CPU and the secondary memory are chosen as in reference[7].

$$\mu_1(i_1) = \frac{64}{i + 16} \text{ and } \mu_2(i_2) = \frac{3i}{i + 6} \tag{5.1}$$

5.1 Numerical Example

The purpose of this section is to apply our method in computing the exact stationary probability distribution described in Sections 3 and 4 to the model of the multiprogramming computer system discussed in the previous section. As we did in examples 4.1 and 4.2 we will compare our iterative algorithm with the ones in references[8, 18, 19], in terms of the number of iterations required to converge to the exact solution and the CPU time. In this example, we consider the multiprogramming system with 6 terminals and the capacity of CPU and the secondary memory is 3. This system can be modeled as a Markov chain with 22×22 probability transition matrix. The computation of this transition matrix with $\delta = 0.9$, $\lambda = 0.2$ and $\varphi = 0$ is given in reference[22]. The time dependent service rates of the CPU and the secondary memory are chosen as in equation (5.1). The stochastic matrix derived from this model is block tridiagonal and the blocks themselves along the diagonal are tridiagonal. The maximum degree of coupling in this example, ε is 0.12 and the maximum number of iterations and the tolerance of the convergence is chosen as in Examples 4.1 and 4.2.

TABLE 3.

Exact method	Kouri	Takahashi	Vantilborgh	Prop. Algor.
π Vector	π_k Vector	π_k Vector	π_k Vector	π_k Vector
0.00000829	0.00002204	0.00003999	0.00000829	0.00000829
0.00002643	0.00007024	0.00012746	0.00002643	0.00002643
0.00020551	0.00006074	0.00011022	0.00020551	0.00020551
0.00007848	0.00010142	0.00010142	0.00007848	0.00007848
0.00054198	0.00053554	0.00053553	0.00054198	0.00054197
0.00231382	0.00229925	0.00229924	0.00231382	0.00231382
0.00024677	0.00025167	0.00025167	0.00024677	0.00024676
0.00144625	0.00144809	0.00144809	0.00144625	0.00144624
0.00464864	0.00464662	0.00464661	0.00464864	0.00464864
0.00984417	0.00983990	0.00983989	0.00984418	0.00984418
0.00088077	0.00088164	0.00088163	0.00088078	0.00088075
0.00551710	0.00551779	0.00551778	0.00551710	0.00551709
0.01979751	0.01979681	0.01979677	0.01979752	0.01979752
0.05213223	0.05212870	0.05212860	0.05213226	0.05213224
0.00239458	0.00239462	0.00239462	0.00239459	0.00239459
0.01631713	0.01631670	0.01631666	0.01631714	0.01631714
0.06566995	0.06566751	0.06566735	0.06566997	0.06566997
0.20279844	0.20279034	0.20278989	0.20279851	0.20279845
0.00356108	0.00356102	0.00356101	0.00356114	0.00356110
0.02687135	0.02687041	0.02687031	0.02687139	0.02687136
0.12399969	0.12399503	0.12399459	0.12399966	0.12399968
0.46069982	0.46068230	0.46068067	0.46069958	0.46069979
CPU time is 0 s, 760 ms	CPU time is 6 s, 150 ms	CPU time is 8 s, 70 ms	CPU time is 5 s, 710 ms	CPU time is 2 s, 360 ms
	Iterations *50*	Iterations 50	Iterations 50	Iterations 8

Notice in Table 3 that Koury, Takahashi and Vantilborgh algorithms do not converge, within the specified tolerance, while the proposed algorithm takes only 8 iterations to converge to the exact solution. We tried to increase the number of iterations but the stationary probability distribution for the other algorithms did not change. Notice also, that Vantilborgh algorithm is very close to the exact, but the other two algorithms are way off.

6. Conclusions

In this paper we have proposed a general transformation that transforms the stochastic systems which can be modeled as large finite-state Markov chains into reduced order systems. This enabled us to compute the exact stationary probability distribution of the nearly completely decomposable Markov chain by solving a reduced order well-conditioned aggregated problem. A block diagonal transformation, which is a subclass of the general one, is proposed to simplify and reduce the amount of computations. This transformation is used to develop an iterative algorithm to compute the exact as well as the $0(\varepsilon^k)$ approximation of the stationary probability distribution. This algorithm is compared with three well-known algorithms known in the literature. It is shown that our algorithm converges in all the examples we considered while the other algorithms do not converge in some of the examples even if we increase the number of iterations. In the examples where the other algorithms converge, our proposed algorithm was found to converge in less number of iterations.

The aggregation procedure is applied to the aggregative analysis of a queueing network model of a computer system to illustrate our approach and its significance.

Acknowledgement

Part of this work is supported by the Scientific Research Administration, Faculty of Engineering, King Abdulaziz University under grant No. 053-410. This support is gratefully acknowledged.

References

[1] **Courtois, P.J.,** *Decomposability Queueing and Computer System Applications,* New York, Academic Press (1977).

[2] **Courtois, P.J.,** Decomposability, instabilities and saturation in multiprogramming systems, *Commun. ACM,* **18:** 371-377 (1975).

[3] **Courtois, P.J.,** *On the Near-Complete Decomposability of Networks of Queues and of Stochastic Models of Multiprogramming Computing Systems,* Rep. CMU-CS-72, 111, Carnegie-Mellon Univ., Pittsburgh (1972).

[4] **Courtois, P.J.** and **Semal, P.,** Computable bounds for conditional steady state probabilities in large Markov chains and queueing models, *IEEE Selected Areas in Communications,* **4**(6): 926-937 (1986).

[5] **Brandwajn, A.,** A model of time virtual memory systems solved using equivalence and decomposition methods, *Acta Informatica,* **4:** 11-47 (1974).

[6] **Konheim, A.G.** and **Reiser, M.,** Finite capacity queuing systems with applications in computer modeling, *SIAM J. Computing,* **7**(2): 210-229 (1978).

[7] **Vantilborgh, H.,** Aggregation with an error of $0(\varepsilon^2)$, *J. ACM,* **32:** 162-190 (1985).

[8] **Vantilborgh, H.,** *The Error of Aggregation, A Contribution to the Theory of Decomposable Systems, and Application,* Ph.D. Thesis, Univ. Cath. Louvian (1981).

[9] **Wunderlich, E.F., Kaufman, L.** and **Gopinath, B.,** The Control of Store and Forward Congestion in Packet-Switched Networks, *Proc. Fifth ICCC*, pp. 2-17 (1980).

[10] **Kaufman, L., Gopinath, B.** and **Wunderlich, E.,** Analysis of Packet Network Congestion Control Using Sparse Matrix Algorithms, *IEEE Trans. Comm.*, **COM-29** 253-465 (1981).

[11] **Simon, H.** and **Ando, A.,** Aggregation of variables in dynamic systems, *Econometrica*, **29:** 111-138 (1961).

[12] **Phillips, R.G.** and **Kokotovic, P.V.,** A singular perturbation approach to modeling and control of Markov chains, *IEEE Trans. Automat. Contr.*, **AC-26:** 1087-1094 (1981).

[13] **Coderch, M., Willsky, A.S., Sastry, S.S.** and **Castanon, D.A.,** Hierarchical aggregation of singularly perturbed finite state Markov processes, *Stochastics*, **8:** 259-289 (1983).

[14] **Lou, X.C., Rohlicek, J., Coxson, P., Verghese, G.** and **Willsky, A.,** Time Scale Decomposition: The Role of Scaling in Linear Systems and Transient States in Finite Markov Processes, *Proc. ACC*, pp. 1408-1413 (1985).

[15] **Rohlicek, J.** and **Willsky, A.,** The reduction of perturbed Markov generators: An algorithm exposing the role of transient states, *J. ACM*, **35:** 675-696 (1988).

[16] **McAllister, D., Stewart, G.W.** and **Stewart, W.J.,** On a Rayleigh-Ritz refinements technique for nearly uncoupled stochastic matrices, *Linear Alg. Applications*, **60:** 1-25 (1984).

[17] **Cao, W.L.** and **Stewart, W.J.,** Iterative aggregation/disaggregation techniques for nearly uncoupled Markov chains, *J. ACM*, **32:** 702-719 (1985).

[18] **Koury, J.R., McAllister, D.F.** and **Stewart, W.J.,** Iterative methods for computing stationary distributions of nearly completely decomposable Markov chains, *SIAM J. Alg. Disc. Math.*, **5:** 164-186 (1984).

[19] **Takahashi, Y.,** *A Lumping Method for Numerical Calculations of Stationary Distribution of Markov Chains,* Res. Rep. B-18, Dept. of Inf. Sci., Tokyo Inst. of Tech., Tokyo, Japan, June (1975).

[20] **Aldhaheri, R.W.** and **Khalil, H.K.,** Aggregation of the Policy Iteration Method for Nearly Completely Decomposable Markov Chains, *IEEE Trans. Automat. Contr.*, **AC 36:** 178-187 (1991).

[21] **Krieger, U.R., Muller-Clostermann, B.** and **Sczittnick, M.,** Modeling and analysis of communication systems based on computational methods for Markov chains, *IEEE Selected Areas in Communications*, **8**(9): 1630-1648 (1990).

[22] **Aldhaheri, R.W.,** *Aggregation of Nearly Completely Decomposable Markov Chains with Application to Multiprogramming Computer System,* Final Report, Research Project No. 051-410, Scien. Res. Admin., Faculty of Engineering, King Abdulaziz University (1990).

الطرق التبسيطية والتجميعية لحساب التوزيع الاحتمالي المستقر لسلاسل ماركوف مع التطبيق على حاسب آلي متعدد البرامج

رباح واصل الظاهري

قسم الهندسة الكهربائية وهندسة الحاسبات ، كلية الهندسة ، جامعة الملك عبد العزيز

جـــــدة - المملكة العربية السعودية

المستخلص . يدرس هذا البحث تبسيط وتجميع نوع من سلاسل ماركوف المعروفة بسلاسل ماركوف الضعيفة التقارن . ولهذا النوع من السلاسل تطبيقات هندسية كثيرة مثل شبكات الانتظار التي تستخدم في نمذجة شبكات الحاسب الآلي وشبكات الاتصالات وغيرها من التطبيقات .

تم التوصل في هذا البحث إلى صيغة رياضية تحويلية عامة يمكن باستخدامها تبسيط النماذج الكبيرة والمعلولة رياضيًا إلى نماذج صغيرة وغير معلولة رياضيًا . وقد وجد أن الصيغ الرياضية المعروفة الآن ماهي إلا حالة خاصة من التحويلة الرياضية العامة التي قدمت في هذا البحث . كما تم إيجاد صيغة رياضية مشتقة من الصيغة الرياضية العامة التي ذكرت آنفا تبسط وتقلل من العمليات الحسابية اللازمة لإيجاد النموذج المصغر من سلسلة ماركوف . وقد استخدمت هذه الصيغة في عمل خوارزمية تحسب التوزيع الاحتمالي المستقر الذي يستخدم في دراسة تقويم الأداء لشبكات الانتظار . وأخيراً تم تطبيق هذه الطريقة التبسيطية والخوارزمية التكرارية على حاسب آلي متعدد البرامج ذي ٦ طرفيات ، كما يمكن للذاكرة الرئيسية والفرعية أن تستوعب ٣ برامج في آن واحد . أيضًا قورنت هذه الخوارزمية مع ثلاث خوارزميات تكرارية معروفة في هذا المجال وقد أثبتت المقارنة بأن هذه الخوارزمية تحتاج إلى عدد أقل من التكرار في كل الأمثلة التي درست كي تصل إلى القيمة الصحيحة بينما الخوارزميات الأخرى لا تصل إلى القيمة الصحيحة في بعض الأمثلة حتى وإن زيد في عدد التكرار .