STRUCTURAL DECOMPOSITION OF MULTIPLE TIME SCALE MARKOV PROCESSES*

J. R. ROHLICEK
BBN Laboratories Incorporated,
10 Moulton St., Cambridge, MA 02238.

A. S. WILLSKY
Laboratory for Information and Decision Systems,
Massachusetts Institute of Technology,
Cambridge, MA 02139.

Abstract

A straightforward algorithm for the multiple time scale decomposition of singularly perturbed Markov processes has been presented in [1,2]. That algorithm provides a uniform approximation of the probability transition function over the interval $t \geq 0$ through the construction of a sequence of aggregate models valid at successively slower time scales. When only the *structure* of these models is desired, the algorithm can be expressed simply in terms of graphs associated with each of the aggregated models. The major computation then becomes computing shortest paths in these graphs. This representation of the algorithm furthermore allows analysis of more complex systems where there are multiple perturbation parameters with unknown relative orders of magnitude.

Keywords: Markov processes, singular perturbation, multiple time scales, graph theory.

1 INTRODUCTION

In this paper, an algorithm for the analysis of the multiple time scale *structure* of a perturbed Markov process is developed which builds directly on the multiple time scale decoposition algorithm presented in [1,2]. Structure of a perturbed Markov process refers to the complete multiple time scale decomposition of the type performed in [1,2,3,4] where the evolution of the state probabilities is approximated using a sequence of aggregated models, each valid a successively slower time scales. In this case, however, only the position of the nonzero transition rates of each of the unperturbed, aggregated time scale models, and the sets of states which constitute the aggregate classes, are determined. Although the detailed behavior of the system cannot be recovered from these descriptions, much useful information is retained. It will be shown that this subset of information about the decomposition can be obtained using a very simple graph-theoretic algorithm which is implicit in the algorithm presented in [1,2].

Use of these structural aspects of a Markov chain can have many applications. For instance, an algorithm similar to that presented in this paper has been applied to the analysis of the behavior of "simulated annealing" optimization methods [5]. Other applications include computing order of magnitudes for the time of events in systems with rare transitions. Specific applications include analysis of the failure time of a fault-tolerant system or the error rate of a communication protocol.

The nature of the decomposition algorithms presented in this paper are very different from those which attempt to approximate the detailed behavior of system. In those algorithms, numerical calculations of dominant eigenvectors of unperturbed systems, and of ϵ-dependent "trapping" probabilities were required. The algorithms in this paper basically consist of computing various types of connectivity in labeled graphs. Since the computations involve only integer quantities, issues of numerical stability are not relevant. Also, by introducing the graph-theoretic formalism, it is possible to employ standard graphical algorithms for computing quantities such as shortest paths between vertices of a graph.

This paper is organized as follows. In the next section, the decomposition algorithm and uniform approximation result present in [1,2] is stated and a simple example is provided. In the next section, the graphical formalism is introduced and the new graphical decomposition algorithm is presented. The final section includes conclusions and a discussion of the relationship of this approach to other work. Also, the application of this algorithm to models with multiple perturbation parameters is discussed with application to a fault-tolerant system model.

*This research was conducted under the support of the Air Force Office of Scientific Research under grant AFOSR-82-0258 and the Army Research Office under grant DAAG-29-84-K005.

2 Markov Process Decomposition Algorithm

In this section, the basic algorithm presented in [1,2] is stated. The reader is referred to those works for a detailed interpretation of this algorithm, proof of the uniform convergence of the approximation, and demonstration of its relationship to the algorithms of Courtois [4] and Coderch [3]. Following the algorithm, a simple example is provided.

Algorithm 1 (Continuous Time) *Begin with the generator*[1] $A^{(0)}(\epsilon)$ *of a finite-state Markov process with one ergodic class for* $\epsilon > 0$. $a_{ji}^{(0)}(\epsilon)$ *is the probability transition rate from state* i *to state* j. *Set* $k \leftarrow 0$.

1. *Partition the state set into the ergodic classes* $E_1, E_2, \ldots, E_N$ *and the transient set* T *generated by* $A^{(k)} = A^{(k)}(0)$. *If there is only a single class* ($N = 1$), *the iteration is terminated.*
2. *For each class* E_I, *compute the ergodic probabilities* $u_{iI}^{(k)}$, $\forall i \in E_I$ *of the member states corresponding to the generator* $A^{(k)}$. *Also set* $u_{jI}^{(k)} = 0$, $\forall j \notin E_I$.
3. *For each state* j *and each class* E_I, *compute terms* $\tilde{v}_{Ij}^{(k)}(\epsilon)$ *such that*

$$\tilde{v}_{Ij}^{(k)}(\epsilon) = v_{Ij}^{(k)}(\epsilon)(1 + \mathrm{O}(\epsilon)) \tag{1}$$

$$\sum_{K=1}^{N} \tilde{v}_{Kj}^{(k)}(\epsilon) = 1 \tag{2}$$

where

$$v_{Ij}^{(k)}(\epsilon) \equiv \Pr\left(\eta^{(k)}(\epsilon, t^*) \in E_I \;\middle|\; \eta^{(k)}(\epsilon, 0) = j,\ t^* = \inf_{t \geq 0}(t \mid \eta^{(k)}(\epsilon, t) \notin T)\right) \tag{3}$$

and $\eta^{(k)}(\epsilon, t)$ *is a sample path of the Markov process generated by* $A^{(k)}(\epsilon)$.

4. *Form the* $n \times N$ *and* $N \times n$ *matrices*

$$U^{(k)} = \left[u_{ij}^{(k)}\right] \quad \textit{and} \quad \tilde{V}^{(k)}(\epsilon) = \left[\tilde{v}_{Ij}^{(k)}(\epsilon)\right] \tag{4}$$

Then

$$A^{(k+1)}(\epsilon) = \frac{1}{\epsilon}\tilde{V}^{(k)}(\epsilon)A^{(k)}(\epsilon)U^{(k)} \tag{5}$$

Set $k \leftarrow k + 1$. *Go to 1.*

The overall approximation of the evolution of the transition probabilities $\phi_{ji}^{(0)}(\epsilon, t)$ *(from state* i *to state* j*) can be written in matrix form as*

$$\begin{aligned}
\Phi^{(0)}(\epsilon, t) \;=\; & \Phi^{(0)}(t) + \\
& \left(U^{(0)}\Phi^{(1)}(\epsilon t)V^{(0)} - U^{(0)}V^{(0)}\right) + \\
& \left(U^{(0)}U^{(1)}\Phi^{(2)}(\epsilon^2 t)V^{(1)}V^{(0)} - U^{(0)}U^{(1)}V^{(1)}V^{(0)}\right) + \\
& \qquad \vdots \\
& \left(U^{(0)}\ldots U^{(k-1)}\Phi^{(k)}(\epsilon^k t)V^{(k-1)}\ldots V^{(0)} - \right. \\
& \qquad \left. U^{(0)}\ldots U^{(k-1)}V^{(k-1)}\ldots V^{(0)}\right) + \mathrm{O}(\epsilon)
\end{aligned} \tag{6}$$

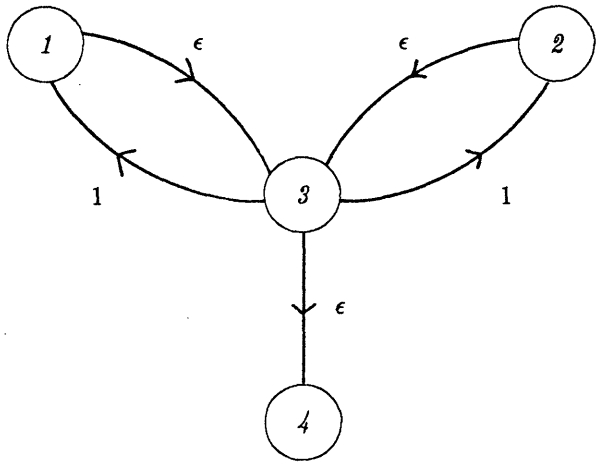
where

$$\Phi^{(k)}(\epsilon, \tau) = \mathrm{e}^{A^{(k)}(\epsilon)\tau},\ \Phi^{(k)}(\tau) = \mathrm{e}^{A^{(k)}\tau},\ \textit{and}\ V^{(k)} = \tilde{V}^{(k)}(0) \tag{7}$$

and where $\mathrm{O}(\epsilon)$ *is a function of* ϵ *and* t *which converges uniformly to zero over* $t \geq 0$. □

A simple application of this algorithm is presented here. The same system will be analyzed using the graphical decomposition algorithm presented in the next section.

[1] The state probabilities $p(\epsilon, t)$ evolve according to $\frac{d}{dt}p(\epsilon, t) = A^{(0)}(\epsilon)p(\epsilon, t)$.



$$A^{(0)}(\epsilon) = \begin{bmatrix} -\epsilon & 0 & 1 & 0 \\ 0 & -\epsilon & 1 & 0 \\ \epsilon & \epsilon & -2-\epsilon & 0 \\ 0 & 0 & \epsilon & 0 \end{bmatrix}$$

Figure 1: Perturbed Markov process

Example 1 *Consider the generator of a Markov process*

$$A^{(0)}(\epsilon) = \begin{bmatrix} -\epsilon & 0 & 1 & 0 \\ 0 & -\epsilon & 1 & 0 \\ \epsilon & \epsilon & -2-\epsilon & 0 \\ 0 & 0 & \epsilon & 0 \end{bmatrix} \tag{8}$$

associated with the state transition diagram in Figure 1. The ergodic classes and transient set are

$$E_1 = \{1\},\ E_2 = \{2\},\ E_3 = \{4\},\ T = \{3\} \tag{9}$$

The ergodic probabilities are all degenerate in this case, therefore

$$u_{11}^{(0)} = u_{22}^{(0)} = u_{43}^{(0)} = 1 \quad or \quad U^{(0)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Suitable terms $\tilde{v}^{(0)}(\epsilon)$ which satisfy (1)-(2) above are

$$\begin{matrix} \tilde{v}_{13}^{(0)}(\epsilon) = \tilde{v}_{23}^{(0)}(\epsilon) = 1/2 - \epsilon/4 \\ \tilde{v}_{33}^{(0)}(\epsilon) = \epsilon/2 \end{matrix} \quad or \quad \tilde{V}^{(0)}(\epsilon) = \begin{bmatrix} 1 & 0 & 1/2-\epsilon/4 & 0 \\ 0 & 1 & 1/2-\epsilon/4 & 0 \\ 0 & 0 & \epsilon/2 & 1 \end{bmatrix} \tag{11}$$

Using these terms, $A^{(1)}(\epsilon)$ computed using (5) generates the process illustrated in Figure 2 and given by

$$A^{(1)}(\epsilon) = \begin{bmatrix} -1/2-\epsilon/2 & 1/2 & 0 \\ 1/2 & -1/2-\epsilon/2 & 0 \\ \epsilon/2 & \epsilon/2 & 0 \end{bmatrix} \tag{12}$$

The procedure is iterated to produce $A^{(1)}(\epsilon)$ and $A^{(2)}(\epsilon)$. $A^{(2)}(0)$ generates a single ergodic class and therefore the iteration is terminated. The set of ϵ-independent Markov models from which the approximation is derived is shown in Figure 3. □

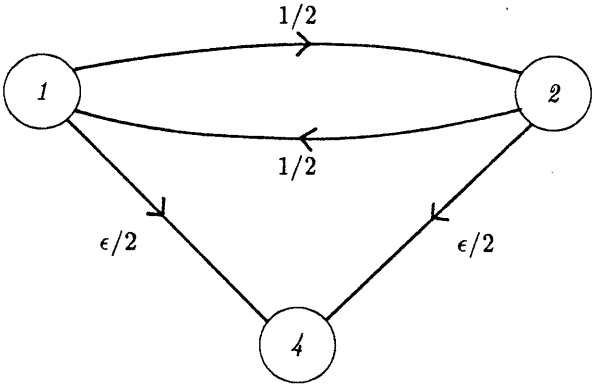


Figure 2: Order $1/\epsilon$ time scale model for Example 1

3 Graphical Decomposition

The uniform approximation (6) produced using Algorithm 1 is expressed as a combination of the behavior of a set of ϵ-independent, aggregated Markov processes with generators $A^{(0)}(0), \ldots, A^{(k)}(0)$ and some associated matrices (the $U^{(i)}$ and $V^{(i)}$) which characterize the recurrent and transient states of these processes. The algorithm can be greatly simplified if only the constituent states of the aggregate classes and the allowable (i.e. nonzero rate) state transitions are desired. In this section, the structure of the graphical representation is presented. Then, the graphical algorithm for the derivation of this entire structural decomposition is described.

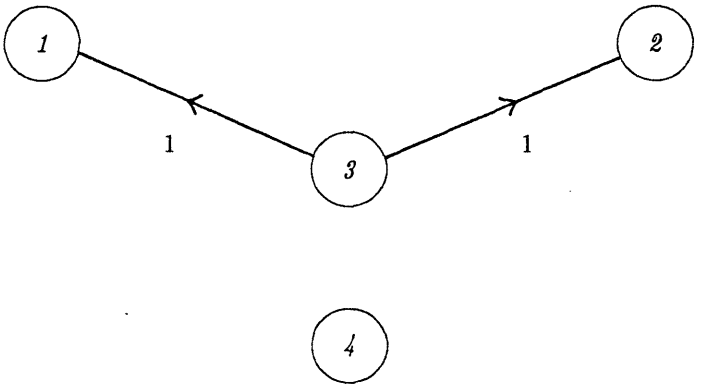
Given a perturbed Markov generator $A^{(k)}(\epsilon)$, an associated graph $G^{(k)} = (\mathcal{N}^{(k)}, \mathcal{E}^{(k)})$ can be constructed as follows. If $A^{(k)}(\epsilon)$ generates an n state process, then the set of vertices is $\mathcal{N}^{(k)} = \{1, 2, \ldots, n\}$. Each edge is a triple, $e = (i, j, w) \in \mathcal{E}^{(k)}$ of the initial vertex, i, the final vertex, j, and a nonnegative integer weight, w. For each nonzero entry $a_{ji}^{(k)}(\epsilon)$, $j \neq i$, a directed, weighted link from vertex i to vertex j is introduced. If $a_{ji}^{(k)}(\epsilon)$ is strictly $O(\epsilon^w)$, the weight of the edge is w. It is also useful to construct the "zero weight" graph $G_0^{(k)} = (\mathcal{N}^{(k)}, \mathcal{E}_0^{(k)})$ from the generator $A^{(k)}(0)$. The set of vertices is unchanged, and the new set of edges, $\mathcal{E}_0^{(k)} \subseteq \mathcal{E}^{(k)}$, corresponds to the subset of zero-weight edges in $\mathcal{E}^{(k)}$.

Several properties of the graph $G = (\mathcal{N}, \mathcal{E})$ will be used. First, define the distance between two vertices $d(i, j)$ as the minimum sum of weights along directed paths from i to j. If there is no such path, then $d(i, j) = \infty$. A *communicating class* C is a maximal set of states such that $i, j \in C$ if and only if $d(i, j) < \infty$. A class C is *final* if there is no vertex $j \notin C$ such that $d(i, j) < \infty$ for any $i \in C$. Also define the distance $\tilde{d}_E(i, C)$ as the minimum weight path from i to the class C with *no* intermediate vertices in the set E. There is a direct association between the Markov process generated by A and the graph G. Most important, an ergodic class generated by A exactly corresponds to a final class of G. Also, a transient state of A corresponds to a vertex of G in some non-final class.

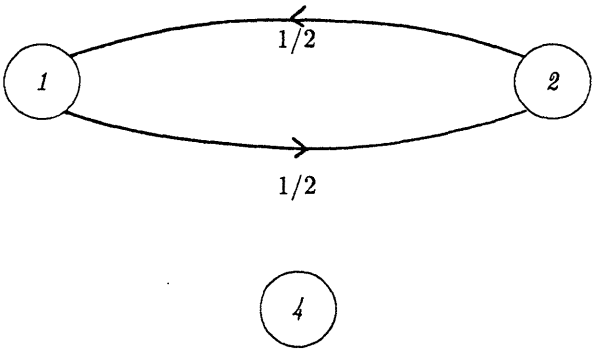
The basic structural decomposition algorithm can be specified in the graphical terms described above. We begin with a graph $G^{(0)}$ constructed using the Markov generator $A^{(0)}(\epsilon)$. Then a sequence of graphs associated with successive time scales, $G_0^{(0)}, G_0^{(1)}, \ldots, G_0^{(K)}$ is constructed such that $G_0^{(i)}$ is associated with the generator $A^{(i)}(0)$ constructed using Algorithm 1. Also, the constituent states in an aggregate class associated with each of the vertices in the graph are computed. The set $\mathcal{U}_i^{(k)}$ is the set of states of $\eta^{(0)}(\epsilon, t)$ associated with the aggregate class represented by vertex i in the graph $G_0^{(k)}$. The set $\mathcal{V}_i^{(k)}$ is the set of states which have an $O(1)$ probability of entering the aggregate class $\mathcal{U}_i^{(k)}$ by the k^{th} time scale.

Algorithm 2 (Structural Decomposition) *Begin with the graph $G^{(0)} = (\mathcal{N}^{(0)}, \mathcal{E}^{(0)})$ constructed from the Markov generator $A^{(0)}(\epsilon)$ of the n state process $\eta^{(0)}(\epsilon, t)$.*
Define $\mathcal{U}_i^{(0)} = \{i\}, \mathcal{V}_i^{(0)} = \{i\}$, $i = 1, 2, \ldots, n$.
Set $k \leftarrow 0$.

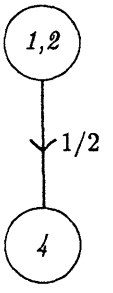
1. *Construct the graph $G_0^{(k)} = (\mathcal{N}^{(k)}, \mathcal{E}_0^{(k)})$ using the zero-weight subset of edges $\mathcal{E}_0^{(k)} \subseteq \mathcal{E}^{(k)}$. Identify*



(a) O(1) *time scale model* — $A^{(0)}$



(b) O$(1/\epsilon)$ *time scale model* — $A^{(1)}$



(c) O$(1/\epsilon^2)$ *time scale model* — $A^{(2)}$

Figure 3: Multiple time scale models for Example 1

the final communicating classes of $G_0^{(k)}$, $E_I^{(k)}$, $I = 1, 2, \ldots, N$. *The complementary set*

$$T^{(k)} \equiv \mathcal{N}^{(k)} - E^{(k)}\ , \quad E^{(k)} \equiv \bigcup_I E_I^{(k)} \tag{13}$$

is the set of vertices in non-final classes. If there is only one final class ($N = 1$), *then the decomposition is completed.*

2. *Form the new aggregate sets*

$$\mathcal{U}_I^{(k+1)} = \bigcup_{i \in E_I^{(k)}} \mathcal{U}_i^{(k)} \tag{14}$$

3. *Define*

$$\tilde{E}_I^{(k)} \equiv \{i \mid \tilde{d}_{E^{(k)}}(i, E_I^{(k)}) = 0\} \tag{15}$$

and form the new sets

$$\mathcal{V}_I^{(k+1)} = \bigcup_{i \in \tilde{E}_I^{(k)}} \mathcal{V}_i^{(k)} \tag{16}$$

4. *Construct a new graph* $\bar{G}^{(k+1)} = (\mathcal{N}^{(k+1)}, \bar{\mathcal{E}}^{(k+1)})$, $\mathcal{N}^{(k+1)} = \{1, \ldots, N\}$. *For each pair of vertices* (I, J) *of* $\bar{G}^{(k+1)}$, $I \neq J$, *the following set of edges comprise* $\bar{\mathcal{E}}^{(k+1)}$.
   (a) *For each edge,* $e = (v_1, v_2, w) \in \mathcal{E}^{(k)}$, *where* $v_1 \in E_I^{(k)}$, $v_2 \in E_J^{(k)}$, *an edge* $(I, J, w - 1)$ *is added to the set* $\bar{\mathcal{E}}^{(k+1)}$.
   (b) *For each edge,* $e = (v_1, t, w) \in \mathcal{E}^{(k)}$, *where* $v_1 \in E_I^{(k)}$, $t \in T^{(k)}$, *such that* $\tilde{d}_{E^{(k)}}(t, E_J^{(k)}) < \infty$, *the edge* $(I, J, w - 1 + \tilde{d}_{E^{(k)}}(t, E_J^{(k)}))$ *is added to the set* $\bar{\mathcal{E}}^{(k+1)}$.
5. *The graph* $\bar{G}^{(k+1)}$ *is "reduced" by retaining only the lowest weight link between any pair of vertices producing the graph* $G^{(k+1)} = (\mathcal{N}^{(k+1)}, \mathcal{E}^{(k+1)})$. *Specifically,*
   (a) *Set* $\mathcal{E}^{(k+1)} \leftarrow \bar{\mathcal{E}}^{(k+1)}$.
   (b) *For each edge* $e = (v_1, v_2, w) \in \mathcal{E}^{(k+1)}$, *remove edge* e *if there is some other edge* $e' = (v_1, v_2, w')$, $w' \leq w$ *remaining in* $\mathcal{E}^{(k+1)}$

   Set $k \leftarrow k + 1$. *Go to step 1*

□

Note that the steps involved are numbered in a fashion to parallel the steps in Algorithm 1 with the exception that the last step is not performed in the previous algorithm. The major computational requirement of this algorithm involves determining the shortest paths, the $\tilde{d}$ terms. This can be accomplished using a variety of standard graphical algorithms. Since all shortest paths from non-final vertices to all final classes are required, an algorithm such as a modification of the Floyd algorithm [6] can be employed.

Algorithm 3 (Shortest path) *Begin with a graph* $G = (\mathcal{N}, \mathcal{E})$, $\mathcal{N} = \{1, \ldots, N\}$, $\mathcal{E} = \{(i, j, d_{ij}) \mid i, j \in \mathcal{N}\}$, *and a set of vertices* E *which may not be used as intermediated vertices.*

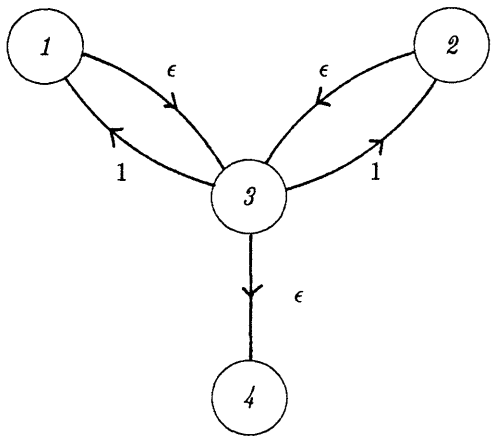
1. *Set* $D_{ij}^{(0)} = \min(d_{ij} \mid (i, j, d_{ij}) \in \mathcal{E})$. $D_{ij}^{(0)} = \infty$ *if there is no edge from* i *to* j.
2. *For* $n = 0, \ldots, N - 1$
   if $n \in E$ *then* $D^{(n+1)} = D^{(n)}$ *otherwise*

$$D_{ij}^{(n+1)} = \min\left(D_{ij}^{(n)}, D_{in}^{(n)} + D_{nj}^{(n)}\right) \quad \forall i \neq j$$

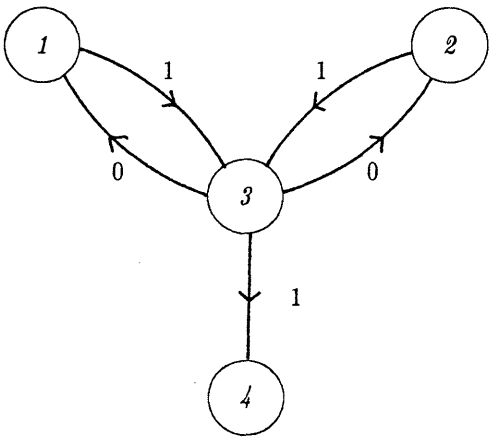
3. $\tilde{d}_E(i, j) = D_{ij}^{(N)}$

□

A simple example is considered using Algorithm 3 in order to clarify the approach. The Markov process considered appears in Example 1 where Algorithm 1 is demonstrated.

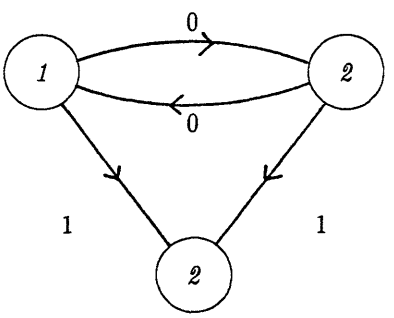


(a) $A^{(0)}(\epsilon)$



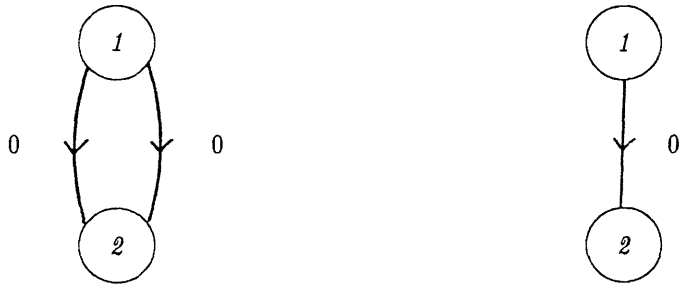
(b) $G^{(0)}$

Figure 4: Perturbed Markov process and associated graphs



$\bar{G}^{(1)} = G^{(1)}$

Figure 5: $O(1/\epsilon)$ time scale graph



(a) $\bar{G}^{(2)}$ (b) $G^{(2)}$

Figure 6: $O(1/\epsilon^2)$ time scale graphs

Example 2 *The generator $A^{(0)}(\epsilon)$ and the associated graph $G^{(0)}$ are shown in Figure 4. The graph $G_0^{(0)}$ has only the edges $(3,1,0)$ and $(3,2,0)$. The final classes of the graph $G_0^{(0)}$ are $E_1^{(0)} = \{1\}$, $E_2^{(0)} = \{2\}$, and $E_3^{(0)} = \{4\}$ and the non-final set is $T^{(0)} = \{3\}$. Using the definition in equation (15), $\tilde{E}_1^{(0)} = \{1,3\}$, $\tilde{E}_2^{(0)} = \{2,3\}$, and $\tilde{E}_3^{(0)} = \{4\}$. The distances in the graph $G^{(0)}$ to the final classes of $G_0^{(0)}$ are $\tilde{d}_{E^{(0)}}(3,1) = \tilde{d}_{E^{(0)}}(3,2) = 0$ and $\tilde{d}_{E^{(0)}}(3,4) = 1$. The sets $\mathcal{U}^{(1)}$ and $\mathcal{V}^{(1)}$ are $\mathcal{U}_1^{(1)} = \{1\}$, $\mathcal{U}_2^{(1)} = \{2\}$, and $\mathcal{U}_3^{(1)} = \{4\}$, and $\mathcal{V}_1^{(1)} = \{1,3\}$, $\mathcal{V}_2^{(1)} = \{2,3\}$, and $\mathcal{V}_3^{(1)} = \{4\}$.*

The graph $\bar{G}^{(1)}$, which in this simple example is already reduced and therefore also the graph $G^{(1)}$, is shown in Figure 5. The sets constructed in this iteration on the algorithm ($k = 1$) are $E_1^{(1)} = \tilde{E}_1^{(1)} = \{1,2\}$, $E_2^{(1)} = \tilde{E}_2^{(1)} = \{3\}$, $T^{(1)} = \{\}$, $\mathcal{U}_1^{(2)} = \{1,2\}$, $\mathcal{V}_1^{(2)} = \{1,2,3\}$, and $\mathcal{U}_2^{(2)} = \mathcal{V}_2^{(2)} = \{4\}$.

Finally, the graph $\bar{G}^{(2)}$ is constructed as shown in Figure 6(a). The reduction in this case corresponds to removing one of the duplicated links from vertex 1 to 2. The sets constructed in this iteration are $E_1^{(2)} = \{2\}$, $\tilde{E}_1^{(2)} = \{1,2\}$, and $T^{(2)} = \{1\}$. The set of graphs $G_0^{(i)}$ and the sets $\mathcal{U}^{(i)}$ and $\mathcal{V}^{(i)}$ are illustrated in Figure 7. This should be compared to the set of Markov processes illustrated in Figure 3 which is obtained using Algorithm 1. □

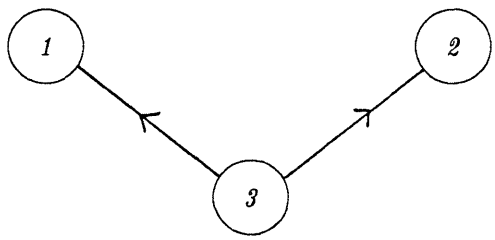
4 Conclusions

The graphical algorithm presented in this paper is significant in that it allows construction of an approximation of the behavior of a singularly perturbed Markov process in which the approximation preserves the multiple time scale structure of the system while not attempting to preserve the detailed dynamics at any time scale. Furthermore, there is no restriction on the ergodic structure of the process at any time scale. Specifically, there may be transient states at any of the intermediate time scale models.

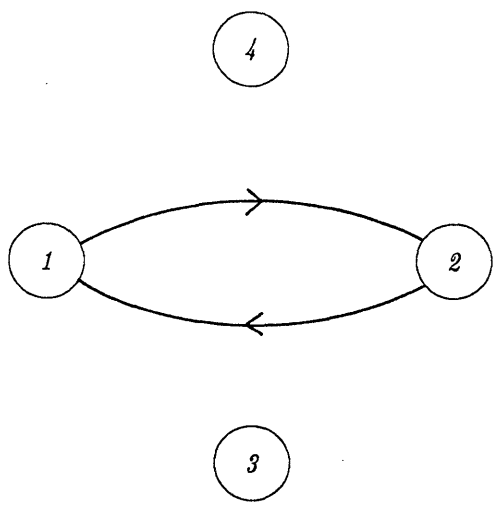
An accurate approximation of the multiple time scale structure has many applications. In particular, complex systems which inherently have transient states can be analyzed using the new algorithm. Such models arise in the analysis of fault-tolerant systems. If failure of the system is associated with some combination of failures of critical components, then complicated multiple time scale structure may occur. The graphical algorithm presented in this paper provides a way to determine the critical sequences of events in the system which may lead to failure and the order of magnitude of the time till such events occur.

Another aspect of this algorithm which is introduced in [1] is that with a modest extension, the multiple time scale structure of systems with multiple perturbation parameters can be considered. Though perturbed Markov processes with two perturbation terms have been considered [7], that method requires making assumptions about the relative orders of magnitude of these parameters. In the extension of the graph algorithm, a set of multiple time scale structural decompositions can be derived where each decomposition is associated with a set of constraints on the relative orders of magnitudes of the perturbation terms. These constraints are associated with different paths in the graphs being associated with the shortest paths between sets of states.

Finally, the graphical algorithm allows identification of state transitions rates which are sufficiently small so that they cannot affect the full multiple time scale approximation formed using Algorithm 1. This observation would allow simplification of each of the generators $A^{(k)}(\epsilon)$ by setting certain transition

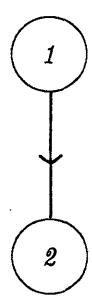


$G_0^{(0)}$



$G_0^{(1)}$

$$\mathcal{U}_1^{(1)} = \{1\} \quad \mathcal{V}_1^{(1)} = \{1,3\}$$
$$\mathcal{U}_2^{(1)} = \{2\} \quad \mathcal{V}_2^{(1)} = \{2,3\}$$
$$\mathcal{U}_3^{(1)} = \{4\} \quad \mathcal{V}_3^{(1)} = \{4\}$$



$G_0^{(2)}$

$$\mathcal{U}_1^{(2)} = \{1,2\} \quad \mathcal{V}_1^{(2)} = \{1,2,3\}$$
$$\mathcal{U}_2^{(2)} = \{4\} \quad \mathcal{V}_2^{(2)} = \{4\}$$

Figure 7: Multiple time scale graphs

rates to zero between iterations. The graphical decomposition algorithm could in this way be used as an initial analysis to reduce to computation required to construct the complete uniform approximation.

REFERENCES

[1] J. R. Rohlicek. *Aggregation and Time Scale Analysis of Perturbed Markov Systems.* PhD thesis, MIT, 1987.

[2] J. R. Rohlicek and A. S. Willsky. The reduction of perturbed Markov generators: an algorithm exposing the role of transient states. September 1985. Submitted to the *Journal of the ACM.*

[3] M. Coderch, A. S. Willsky, S. S. Sastry, and D. A. Castinon. Hierarchical aggregation of singularly perturbed finite state Markov processes. *Stochastics*, 8:259–289, 1983.

[4] P. J. Courtois. *Decomposability: Queuing and Computer Systems Applications.* Academic Press, New York, 1977.

[5] J. N. Tsitsiklis. *Markov Chains with Rare Transitions and Simulated Annealing.* Technical Report LIDS-P-1497, MIT, September 1985.

[6] R. W. Floyd. Algorithm 97 — shortest path. *Communications of ACM*, 5:345, 1962.

[7] G. S. Ladde and D. D. Siljak. Multiparameter singular perturbations of linear systems with multiple time scales. *Automatica*, 19(4):385–394, 1983.