Efficient Solutions for a Class of Non-Markovian Models

Edmundo de Souza e Silva[1]
UCLA Computer Science Department
Los Angeles, CA 90024

H. Richard Gail
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598

Richard R. Muntz[1]
UCLA Computer Science Department
Los Angeles, CA 90024

Abstract

The use of embedded Markov chains has been known for some time. However the application of this technique has been very ad hoc and not established as a standard approach for a wide class of models. Recently however, there has been progress in the direction of identifying an interesting class of models which are not Markovian but which can yield to a well defined solution method based on the analysis of an embedded Markov chain. Example applications that yield to this approach include polling models with deterministic timeout periods. In this paper we derive efficient methods for the computation of the transition probabilities for the embedded chain, and performance measures expressable as Markov reward functions. Computation of the transition probabilities for the embedded chain requires transient analysis and our computation procedures are based on uniformization for computing transient behavior. Examples are given to demonstrate the effectiveness of the methods and the extended class of models that are solvable with these techniques.

[1]The work of E. de Souza e Silva and R. R. Muntz was partially supported by grants from National Science Foundation CCR-9215064 and from CNPq(Brazil).

1 Introduction

Embedded Markov chain models have been widely used as a tool to obtain performance measures of interest for non-Markovian models. Given an initial process $\mathcal{X} = \{X(t) : t \geq 0\}$ for which measures have to be computed the approach can be outlined in three basic steps. (1) First, embedded points are identified which define a discrete time Markov process $\mathcal{Y}$ that represents the state of the system at these time points. Such a chain is called the embedded Markov chain. For any given non-Markovian model it may not be evident how to find embedded points in order to obtain a Markov process $\mathcal{Y}$. However, for many models, this identification can be done automatically, from a high level model specification. (2) Once the embedded points are identified, the problem is to compute the transition probability matrix for $\mathcal{Y}$. This may not be a trivial task and may require elaborate computational procedures. (3) The third step is to obtain the measures of interest from the solution of the Markov process $\mathcal{Y}$.

It is not uncommon that step (2) is attacked using transient solution techniques. In fact, several recent papers which solve non-Markovian models using the embedded Markov chain approach have used transient analysis as part of the overall solution. We survey a few of these efforts.

Grassmann [9] addressed the problem of finding the stationary state probabilities for a GI/PH/1 queue. In this case the embedded chain is easily identified at arrival points, and the problem reduces to solving step (2) indicated above. The state of the system at embedded points is composed of the number of customers in the system and the service phase of the customer being served. In order to calculate the transition probability matrix of $\mathcal{Y}$, one must obtain the probability $p_{i,j,n}$ that the server served n customers between successive arrivals and that the service phase changed from i to j. The interarrival time distribution is given and transient analysis was used to calculate the entries in the embedded matrix. In particular the uniformization technique [14] was shown to provide a useful approach to making the required calculations.

In [5] scheduled maintenance policies of computer systems were analyzed. Embedded points were found at times where scheduled maintenance is performed. In two of the policies studied, the time between embedded points was not constant. Furthermore, these times did not have a given distribution as in the example above. Instead the interval between embedded points depends on the manner in which the model evolves with time. For example, in one of the policies a maintenance visit occurs if the system goes down before a regular scheduled maintenance. It was shown that the transition probabilities can also be obtained from the transient solution of a continuous Markov chain model that describes the failure and repair behavior of the system between arrivals of the scheduled repair person, and the uniformization technique was used as the foundation of the approach. The theory of Markov reward models was invoked to calculate the measure of interest (step (3) above), namely the system availability.

Marsan *et al* [1, 2] introduced the so-called Deterministic and Stochastic Petri Net (DSPN) models. DSPNs are a class of timed Petri Net models with deterministic and exponential transitions such that an embedded Markov chain can be automatically identified from the high level Petri Net model and the transition probabilities calculated. The class of models considered includes Petri net models where at most one deterministic transition is enabled at a time. The features allowed are such that the underlying model obtained is a semi-Markov chain. Among the features included are exponential transitions that can be enabled concurrently with a deterministic transition and deterministic transitions that can be disabled when certain exponential transitions fire. The epochs of the embedded chain are those instants at which a deterministic transition is either disabled or fires (after being enabled for a period of time) and, instants when an exponential transition fires and no deterministic transition is enabled. Once the embedded points are identified, the state space is partitioned into two subsets: one partition contains all states (markings in Petri net terminology) in which a deterministic transition is enabled and the other partition contains the remaining states. The transition probabilities among states of the second partition are easily calculable since, for states in this partition, no deterministic transition is enabled. The problem of determining the transition probability from a state in the first partition to any other state is decomposed by conditioning on the event that the enabled deterministic transition fires and on the complementary event that the deterministic transition is disabled before it can fire. The Chapman-Kolmogorov equations are used to obtain these probabilities, and thus calculations involving matrix exponentials are necessary. In the same manner the residence time in each state of the model is computed. From the mean residence times and the solution to the embedded model the final solution of the semi-Markov model is computed. It should be noted that the overall solution cost is substantial, since the transient analysis needed for obtaining transition probabilities requires calculating the exponential of matrices, where the number of nonzero entries is of the same order of magnitude as the cardinality of the state space of the original model. Furthermore, due to the solution method employed, any sparseness in the original model is destroyed.

In [2] it was recognized that the cost of the transient solution necessary to calculate the transition probabilities of the embedded chain could be reduced if the original DSPN model had so-called "independent subnetworks". Roughly, an independent subnetwork evolves independently of other subnetworks in the model when a deterministic transition is enabled. As a consequence the transient analysis necessary to compute the transition probabilities and expected sojourn times can be done independently for each subnetwork. This is done after finding a matrix resulting from the merging of one of these subnetworks with the "preemptive subnetwork" (a preemptive subnetwork is one whose evolution may preempt an enabled deterministic transition). The final model solution requires the combination of the results from different subnetworks. In general this procedure is nontrivial, and the approach described in [2] is complex.

Several papers followed based on [1, 2]. In [15], the uniformization technique was used for the transient analysis needed to compute the transition probabilities, instead of the approach used in [1, 2]. Only non-preemptive deterministic transitions were considered in [15]. In [8] the method of supplementary variables was used to handle models containing transitions with general distributions and models in which two or more non-exponential transitions can be enabled simultaneously. The solution of the partial differential equations obtained was done numerically by discretizing time. Choi *et al* [4] showed that the underlying embedded model need not be restricted to semi-Markov models, but may belong to a broader class of Markov regenerative models. The transient analysis is performed by taking the Laplace transform of the differential equations describing the time dependent behavior of the model to calculate the transition probabilities of the embedded chain. The transform equations can then be numerically inverted. The steady state solution is obtained by using results from Markov reward models, as was done in [5].

In this paper we show how the approach of [7] for solving polling models with server timeouts can be easily generalized to develop a simple and efficient solution method for solving a broad class of non-Markovian models. We consider the case where only one non-exponential transition can be active at a time, similar to the DSPN case.

In Section 2 we present the necessary backgound material and introduce the class of models that can be handled. In Section 3 we define the embedded chain and find recursions for calculating quantities necessary for the intermediate steps of the solution. Section 4 presents recursions for calculating the measures of interest. In Section 5 we describe the computational complexity of the approach. Section 6 extends the class of models that can be handled. An example which illustrates the proposed approach is presented in Section 7, and Section 8 concludes the paper.

2 Background Material and Model Paradigm

We begin this section with an informal description of the model paradigm adopted, based on the notion of interacting objects and events (see [3]). The system model consists of a set of objects which interact with each other via a message passing mechanism. Each object has an internal state which evolves over time. The state of an object may change as a result of an *event* which is generated by the object itself or due to a message received from another object. The state of an object determines the events that can occur for that object, i.e., the set of events which are *enabled*. Once enabled, an event occurs after an amount of time which is either: (a) exponentially distributed; or (b) deterministically distributed. (The case of events with general distribution are considered later.) Events that are enabled may be disabled due to changes in the internal state of the model. The occurrence of an event may simply cause a state change in the object that generates the event, or it may also cause one or more messages to be sent to other objects. Messages are delivered in zero time and are used as a mechanism to cause objects to *react* (change state) to an event executed by another object. The state of the system is a vector containing the internal states of each object in the model and the list of undelivered messages. Since messages are delivered in zero time, some of the states are transitory and have zero holding time. These correspond to the *vanishing* states in stochastic Petri net terminology. States with positive mean holding time are called *tangible*. For the purposes of this paper we assume that the vanishing states have been eliminated and only the tangible states are present in the model.

We now give a more formal definition of the processes we consider. We assume there are E deterministic events φ_j, $j = 1, \ldots, E$, and at most one deterministic event can be enabled at any time. A deterministic event is one for which, once enabled, the time until it executes is deterministic, unless it is disabled due to another event. The index $j = 0, \ldots, E$ is an indicator of which deterministic event is enabled, where $j = 0$ represents the case when no deterministic transition is enabled. Now consider a continuous time process $\mathcal{X} = \{X(t) : t \geq 0\}$ with finite state space. The state space of $\mathcal{X}$ will be denoted by $\mathcal{S}$. Each state in $\mathcal{S}$ is an appropriately dimensioned vector in which the ith component contains the state of the ith object in the system model. Note that $X(t)$ is not a Markov process, since there are non-exponential events in the model. Let $\eta_1, \eta_2, \ldots$ be the times when either: (1) a deterministic event is disabled and another deterministic event is enabled; (2) a deterministic event is disabled and no deterministic event is enabled; or (3) a deterministic event is enabled when no deterministic event was previously enabled. The times $\eta_1, \eta_2, \ldots$ are not regeneration points for $\mathcal{X}$, but the values of $\mathcal{X}$ at these points yield an embedded discrete time vector Markov chain $\mathcal{Y} = \{Y_k : k = 1, 2, \ldots\}$ given by $Y_k = X(\eta_k)$.

For each $j = 1, \ldots, E$ there is a finite set of transitions corresponding to φ_j going from disabled to enabled or a firing of φ_j which results in a state in which φ_j is again enabled. The union of all "destination states" of these transitions will be denoted $\mathcal{S}'^{(j)}$. These are the initial states of what we will call a Δ_j mini-interval, i.e., an interval in which deterministic event φ_j is enabled. There is a similar definition for $\mathcal{S}'^{(0)}$, which is the set of initial states for a Δ_0 mini-interval, i.e., an interval in which no deterministic event is enabled. The state space for the embedded Markov chain $\mathcal{Y}$ is $\mathcal{S}' = \bigcup_{j=0}^{E} \mathcal{S}'^{(j)}$. The sets $\mathcal{S}'^{(j)}$ are disjoint, since at most one deterministic event can be enabled at a time.

The main problem we face is to determine the transition probabilities of the embedded Markov chains defined above. Due to the assumption that only one deterministic event is enabled at a time, if we consider an interval that starts when a deterministic event, say event φ_j, is enabled and ends when φ_j is disabled (either due to the expiration of the deterministic time associated with this event or to the execution of an exponential event which leads the model to a state where φ_j is disabled) the process is Markovian for that interval of time. This is clear since only events with exponential holding times can execute while φ_j is active. That is, although the process $\mathcal{X}$ is not Markovian, it is Markovian between η_k and η_{k+1}.

Our approach is to isolate the Markov chain for each type of mini-interval and analyse each chain to determine the transition probabilities for the embedded Markov chain $\mathcal{Y}$. The state space of the Markov chain corresponding to a mini-interval Δ_j is denoted by $\mathcal{S}^{(j)}$, and it contains all of the intial states $\mathcal{S}'^{(j)}$ for a mini-interval Δ_j plus all of the states that are reachable by transitions that do not correspond to disabling φ_j nor a firing of φ_j. From analysis of the Markov chain for Δ_j intervals we will be able to obtain the required transition probabilities for $\mathcal{Y}$. Due to the deterministic transitions, this requires transient analysis. However, it is often the case that the Markov chain for a mini-interval can be decomposed into independent chains. One or more of the component subchains lead to transitions that can disable the deterministic transition. Other subchains will evolve independently until the end of the mini-interval (which occurs when either the deterministic transition fires or it is disabled). We will be able to take advantage of the decomposition of the Markov process that governs the evolution of the system in a mini-interval to reduce computation costs using the transient analysis methods described in the next section.

The uniformization technique gives the foundation of the approach we use. Briefly, a continuous time Markov process $\mathcal{X} = \{X(t) : t \geq 0\}$ with generator matrix $\mathbf{Q}$ and finite state space can be transformed into a discrete time Markov chain $\mathcal{Z} = \{Z_n : n = 0, 1, \ldots\}$ with the same state space as $\mathcal{X}$ and with transition probability matrix $\mathbf{P} = \mathbf{Q}/\Lambda + \mathbf{I}$, where the uniformization rate $\Lambda \geq \max\{q_i\}$, and the transition times occur according to a Poisson process $\mathcal{N} = \{N(t) : t \geq 0\}$ independent of $\mathcal{Z}$ and with rate Λ. In other words, $X(t) = Z_{N(t)}$. Details can be found in [6, 11, 12] and the references therein.

Finally, we note that algorithms for automatically determining the states of the embedded Markov chain $\mathcal{Y}$ and the states in each type of mini-subinterval is not a difficult problem. An existing package which assumes that all events are exponentially distributed (even those that are deterministic) can be used to obtain all reachable tangible states. It is easy to see that if only one deterministic event is enabled at a time, then the set of reachable states is not changed if we replace the deterministic event by an exponential event. Extra information is supplied to the chain generator to indicate the deterministic events. Clearly, from this information and the information concerning the state representation, it is easy to determine all transitions that correspond to a deterministic event becoming enabled and therefore the elements in each subset $\mathcal{S}'^{(j)}$ are determined. The states in $\mathcal{S}^{(j)}$ and the transition rates between them are likewise easily determined. So all of the "pieces" are easily determined with variants of existing algorithms.

3 The Methodology

Our method of analysis involves studying the behavior of the embedded Markov chain defined at times $\eta_1, \eta_2, \ldots$. Let T_j be the constant time associated with deterministic event φ_j. We consider events with generally distributed holding times in Section 6.

3.1 Preliminaries

As described in the previous section, we proceed by considering mini-intervals in which a given deterministic event is enabled or in which no deterministic event is enabled. Let Δ_j denote a mini-interval in which deterministic event φ_j is enabled, and let $\mathbf{G}^{(j)}$ denote the transition rate matrix for the continuous time Markov chain $\mathcal{V}^{(j)}$ that represents the evolution of the system model while in such an interval. The state space of this Markov chain includes $\mathcal{S}^{(j)}$, which was defined in the previous section to include all the states of the original model in which event φ_j is enabled. In addition, we will include the transitions and target states for transitions that can end a Δ_j interval. These states are convenient to include, since the transition probabilities for the embedded Markov chain correspond to the probabilities of which absorbing state (initial state of a new mini-interval) the system model ends up in, given each initial state for the mini-interval.

The matrix $\mathbf{G}^{(j)}$ has the form

$$\mathbf{G}^{(j)} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{G}^{(j,0)} & \mathbf{G}^{(j,1)} & \cdots & \mathbf{G}^{(j,j)} & \cdots & \mathbf{G}^{(j,E)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}. \tag{1}$$

As mentioned in the previous section, the states and transitions are a subset of those of the original model and are easily extracted from the original model.

Since we will use uniformization to do transient analysis on the Markov chain $\mathcal{V}^{(j)}$, we form a new discrete Markov chain with transition probability matrix $\mathbf{F}^{(j)} = \mathbf{G}^{(j)}/\Lambda^{(j)} + I$

(where $\Lambda^{(j)}$ is the uniformization rate) in the usual manner to obtain

$$\mathbf{F}^{(j)} = \begin{bmatrix} I & 0 & \cdots & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{F}^{(j,0)} & \mathbf{F}^{(j,1)} & \cdots & \mathbf{F}^{(j,j)} & \cdots & \mathbf{F}^{(j,E)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & I \end{bmatrix}. \tag{2}$$

The full transition probability matrix for $\mathcal{Y}$ can be expressed as

$$\mathbf{H} = \begin{bmatrix} \mathbf{D}^{(0,0)} & \mathbf{D}^{(0,1)} & \mathbf{D}^{(0,2)} & \cdots & \mathbf{D}^{(0,E)} \\ \mathbf{D}^{(1,0)} & \mathbf{D}^{(1,1)} & \mathbf{D}^{(1,2)} & \cdots & \mathbf{D}^{(1,E)} \\ \mathbf{D}^{(2,0)} & \mathbf{D}^{(2,1)} & \mathbf{D}^{(2,2)} & \cdots & \mathbf{D}^{(2,E)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}^{(E,0)} & \mathbf{D}^{(E,1)} & \mathbf{D}^{(E,2)} & \cdots & \mathbf{D}^{(E,E)} \end{bmatrix}. \tag{3}$$

In the above matrix we have ordered the states first according to the partition imposed by the mini-intervals, and within each partition of states some arbitrary order is used. The problem is to develop efficient computational algorithms to compute the elements of $\mathbf{H}$. For every nonzero element $d_{s',s}$ in $\mathbf{H}$, s' is a state in $\mathcal{S}'^{(j)}$ for some j, $0 \leq j \leq E$, and similarly s is a state in $\mathcal{S}'^{(k)}$ for some k, $0 \leq k \leq E$. For s' a state in $\mathcal{S}'^{(j)}$, the probabilities will be determined from transient analysis of the uniformized Markov chain corresponding to $\mathcal{V}^{(j)}$ with transition probability matrix $\mathbf{F}^{(j)}$.

We calculate the entries of the jth block row of $\mathbf{H}$ by conditioning on the length of time until φ_j executes or is disabled. Let δ_j, $j = 1, \ldots, E$, be a random variable equal to the length of a randomly chosen mini-interval Δ_j during which φ_j is enabled. Likewise, δ_0 is a random variable equal to the length of a mini-interval Δ_0 during which no deterministic event is enabled. We first determine the distribution for δ_j, $j \neq 0$. Since no deterministic event other than φ_j can be enabled during mini-interval Δ_j (for $j = 1, \ldots, E$) and no deterministic events are enabled during Δ_0, the behavior of the original process $\mathcal{X}$ during a mini-interval is Markovian. We can identify two mutually exclusive and exhaustive outcomes for mini-interval Δ_j, $j \neq 0$: (a) the deterministic event associated with the mini-interval fires, and in this case $\delta_j = T_j$; (b) the execution of another event in the system disables φ_j, and in this case $\delta_j < T_j$.

For $j = 0, \ldots, E$, the system behavior during Δ_j is given by a continuous time Markov chain whose state space consists of all states in $\mathcal{S}^{(j)}$ plus absorbing states $\mathcal{S}'^{(k)}$, $k \neq j$, which can be reached in a direct transition from a state in $\mathcal{S}^{(j)}$. We assume that this process evolves as a vector-valued Markov chain with $M_j + 1$ components, where these entries correspond to M_j+1 independent chains $\mathcal{W}_0^{(j)}, \ldots \mathcal{W}_{M_j}^{(j)}$. That is, lumping the states of the original vector-valued chain with the same ith component, $i=0,\ldots,M_j$, yields M_j+1 chains. The numbering of the independent subchains is such that $\mathcal{W}_0^{(j)}$ is the preemptive chain, i.e., the subchain that leads to the absorbing states in the $\mathcal{S}'^{(k)}$, while chains $1,\ldots,M_j$ correspond to the nonpreemptive chains. In other words, for each j, $\mathcal{S}^{(j)}$ can be written as the product $\mathcal{S}^{(j)} = \mathcal{S}_{0,j} \times \cdots \times \mathcal{S}_{M_j,j}$, where $\mathcal{S}_{i,j}$ is the set of states of the ith independent chain $\mathcal{W}_i^{(j)}$.

We have assumed that M_j+1 independent subchains can be identified corresponding to the mini-interval Δ_j. In [2] an algorithm was proposed to identify independent chains from a Petri Net description. Here we do not discuss this issue and assume that the independent chains are given. We should also note that if such a factorization into independent subchains exists but is not recognized, then the procedure we describe below still is applicable, but the computational complexity will be greater than using the independent subchains. In [2] it was recognized that such independent chains can lead to computational savings when calculating the entries of the embedded matrix of the DSPN model. The entries were calculated by combining each of the independent chains in turn with the preemptive chain responsible for disabling the corresponding deterministic transition. However, the necessary step to combine the solutions of the independent chains was very complex and costly. Instead, we calculate the entries by using the distribution of δ_j, the length of the mini-interval Δ_j. We will develop simple recursions which significantly simplify the overall solution.

Consider $\mathcal{W}_0^{(j)}$ for $j=1,\ldots,E$. For notational convenience we drop the subscript 0 of $\mathcal{W}_0^{(j)}$ and the subscript j of M_j whenever the meaning is clear from the context. This chain begins to evolve when φ_j is enabled, and it continues until either φ_j has executed for a deterministic time T_j or φ_j is disabled when $\mathcal{W}^{(j)}$ reaches an absorbing state, whichever comes first. Likewise, for $j=0$, the continuous time Markov chain $\mathcal{W}^{(0)}$ has states where no deterministic transition is enabled plus absorbing states which are reached after the execution of an event which enables any deterministic transition.

The distribution of δ_j can be calculated from the preemptive chain $\mathcal{W}^{(j)}$ as follows. Let

$$F(t,\boldsymbol{\nu}) \stackrel{\text{def}}{=} P[\delta_j \le t|\boldsymbol{\nu}], \qquad 0 \le t < T_j$$

be the distribution for δ_j given an initial distribution $\boldsymbol{\nu}$ for chain $\mathcal{W}^{(j)}$. Equivalently, $F(t,\boldsymbol{\nu})$ is the probability of being in an absorbing state at time t. Using basic uniformization results, for $0 \le t < T_j$ we have

$$F(t,\boldsymbol{\nu}) = \sum_{n=0}^{\infty} e^{-\Lambda_j t} \frac{(\Lambda_j t)^n}{n!} \pi_{\text{abs}}(n,\boldsymbol{\nu}). \tag{4}$$

Here Λ_j is the uniformization rate for chain $\mathcal{W}^{(j)}$, $\boldsymbol{\pi}(0,\boldsymbol{\nu}) = \boldsymbol{\nu}$, $\boldsymbol{\pi}(n,\boldsymbol{\nu}) = \boldsymbol{\pi}(n-1,\boldsymbol{\nu})\mathbf{W}^{(j)}$ where $\mathbf{W}^{(j)}$ is the transition matrix of the uniformized chain $\mathcal{W}^{(j)}$, and $\pi_{\text{abs}}(n,\boldsymbol{\nu})$ is the probability of being in an absorbing state at step n of the uniformized chain $\mathcal{W}^{(j)}$. The density $F'(t,\boldsymbol{\nu})$ for $0 < t < T_j$ is given by

$$F'(t,\boldsymbol{\nu}) = \sum_{n=1}^{\infty} e^{-\Lambda_j t} \frac{(\Lambda_j t)^{n-1}}{(n-1)!} \Lambda_j \left\{ \pi_{\text{abs}}(n,\boldsymbol{\nu}) - \pi_{\text{abs}}(n-1,\boldsymbol{\nu}) \right\}. \tag{5}$$

The quantity

$$\pi_{\text{abs}}(n,\boldsymbol{\nu}) - \pi_{\text{abs}}(n-1,\boldsymbol{\nu}) \tag{6}$$

is the probability of being absorbed at exactly the nth step of the uniformized Markov chain $\mathcal{W}^{(j)}$ given an initial distribution $\boldsymbol{\nu}$.

Let s'_c and s_c be the state of independent subchain c at the beginning and the end of interval δ_j, respectively, for $c = 0,\ldots,M$. The index 0 is used to refer to states of the preemptive chain $\mathcal{W}^{(j)}$. Since the $M+1$ chains are independent, we clearly have after conditioning on the length of the mini-interval $\delta_j = t$ and the initial state s'

$$d^{(j)}_{s',s}|t = \prod_{c=0}^{M} P[s'_c \to s_c|t],$$

where the system state $\mathbf{s} = \langle s_0, s_1, \ldots, s_M \rangle$ is a vector for which the entries are the states of the $M+1$ independent chains. For $0 \le t < T_j$, after uniformizing each independent chain, we have using (5)

$$d^{(j)}_{s',s}|t = \left[\sum_{k_0=0}^{\infty} e^{-\Lambda_0 t} \Lambda_0 \frac{(\Lambda_0 t)^{k_0}}{k_0!} \phi_{s_0}(k_0+1) \right] \prod_{c=1}^{M} \sum_{k_c=0}^{\infty} e^{-\Lambda_c t} \frac{(\Lambda_c t)^{k_c}}{k_c!} \pi_{s_c}(k_c).$$

Here Λ_c is the uniformization rate, $\pi_{s_c}(k_c)$ is the probability that the uniformized chain c is in state s_c after k_c steps from initial state s'_c, and $\phi_{s_0}(k_0+1)$ is the probability that the uniformized chain $\mathcal{W}^{(j)}$ is absorbed in state s_0 at exactly step k_0+1 starting from initial state s'_0. Thus

$$d^{(j)}_{s',s}|t = \sum_{n=0}^{\infty} \sum_{\mathbf{k}\in\mathcal{K}_n} \left[\prod_{c=0}^{M} e^{-\Lambda_c t} \frac{(\Lambda_c t)^{k_c}}{k_c!} \right] \left[\prod_{c=1}^{M} \pi_{s_c}(k_c) \right] \Lambda_0 \phi_{s_0}(k_0+1),$$

where $\mathbf{k} = \langle k_0, \ldots, k_M \rangle$ and $\mathcal{K}_n = \{\mathbf{k} : \|\mathbf{k}\| = n\}$ with $\|\mathbf{k}\| = k_0 + \cdots + k_M$. Unconditioning on the length of the interval δ_j, for $0 \le t < T_j$, yields

$$d^{(j)}_{s',s} = \sum_{n=0}^{\infty} \sum_{\mathbf{k}\in\mathcal{K}_n} \left(\frac{\Lambda_0}{\gamma}\right) n! \left[\prod_{c=0}^{M} \frac{(\Lambda_c/\gamma)^{k_c}}{k_c!} \right] \prod_{c=1}^{M} \pi_{s_c}(k_c) \phi_{s_0}(k_0+1) \int_0^{T_j} e^{-\gamma t} \gamma \frac{(\gamma t)^n}{n!} dt \tag{7}$$

where $\gamma = \sum_{c=0}^{M} \Lambda_c$. After performing the integration and doing some simple algebraic manipulations, we have

$$d^{(j)}_{s',s} = \sum_{n=1}^{\infty} e^{-\gamma T_j} \frac{(\gamma T_j)^n}{n!} \left(\frac{\Lambda_0}{\gamma}\right) \sum_{\sigma=0}^{n-1} \sum_{\mathbf{k}\in\mathcal{K}_\sigma} \sigma! \left[\prod_{c=0}^{M} \frac{(\Lambda_c/\gamma)^{k_c}}{k_c!} \right] \prod_{c=1}^{M} \pi_{s_c}(k_c) \phi_{s_0}(k_0+1). \tag{8}$$

Equation (8) can also be interpreted in the following way. Consider a superposition of $M+1$ Poisson processes representing transitions of the independent chains. Let n be the number of transitions of the aggregate process during $(0, T_j)$, and let $\sigma+1$ be the transition at which absorption occurred. Setting $\sigma = \sum_{c=0}^{M} k_c$, the expression in (8) gives the probability that k_c of the transitions before absorption are from nonpreemptive chain c, $c = 1, \ldots, M$, and k_0 are transitions of chain $\mathcal{W}^{(j)}$ (so that absorption occurs at step k_0+1 of the preemptive chain). Given the number of transitions of each chain, the probability of a transition from state s' to s is simply $\prod_{c=1}^{M} \pi_{s_c}(k_c)\phi_{s_0}(k_0+1)$. Recalling that we have a total of n transitions, when $\sigma + 1 < n$ the additional transitions of the composite process are ignored, since the mini-interval j ends when absorption occurs. Unconditioning on n gives (8).

Equation (8) contains the combinatorial term $\sum_{\sigma=0}^{n-1}\sum_{\mathbf{k}\in\mathcal{K}_\sigma}$. In what follows we show that these terms can be aggregated to produce a simple recursion. Initially, we define two functions and find recursions for each.

For $\mathbf{k} \in \mathcal{K}_\sigma$ (i.e., $\sigma = \sum_{c=0}^{M} k_c$), let

$$f(\sigma, \mathbf{k}) = \sigma! \prod_{c=0}^{M} \frac{(\Lambda_c/\gamma)^{k_c}}{k_c!}. \tag{9}$$

Then $f(\sigma, \mathbf{k})$ can be evaluated recursively as follows.

Lemma 1 *For* $||\mathbf{k}|| = \sigma > 0$,

$$f(\sigma, \mathbf{k}) = \sigma \frac{\Lambda_c/\gamma}{k_c} f(\sigma - 1, \mathbf{k} - \mathbf{1}_c) \tag{10}$$

for any c such that $k_c > 0$.

Proof: We can rewrite (9) as

$$f(\sigma, \mathbf{k}) = \sigma \frac{\Lambda_c/\gamma}{k_c}\left[(\sigma-1)! \prod_{\substack{i=0\\ i\neq c}}^{M} \frac{(\Lambda_i/\gamma)^{k_i}}{k_i!} \cdot \frac{(\Lambda_c/\gamma)^{k_c-1}}{(k_c-1)!}\right]$$

Recognizing the term in brackets as $f(\sigma-1, \mathbf{k}-\mathbf{1}_c)$, we obtain (10). □

Now for $\mathbf{k} \in \mathcal{K}_\sigma$, define

$$\Gamma_{\mathbf{s}}[\sigma, \mathbf{k}] = \prod_{i=1}^{M} \pi_{s_i}(k_i)\phi_{s_0}(k_0+1), \tag{11}$$

where $\mathbf{s} = \langle s_0, \ldots, s_M \rangle$.

Lemma 2 *For* $||\mathbf{k}|| = \sigma > 0$,

$$\Gamma_{\mathbf{s}}[\sigma, \mathbf{k}] = \sum_{s_c^* \in \mathcal{S}_c} p^{(c)}_{s_c^*, s_c} \Gamma_{\mathbf{s}^*(c)}[\sigma - 1, \mathbf{k} - \mathbf{1}_c] \tag{12}$$

for any $c \neq 0$ *such that* $k_c > 0$, *where* $\mathbf{s}^*(c) = \langle s_0, \ldots, s_{c-1}, s_c^*, s_{c+1}, \ldots, s_M \rangle$ *and* $\mathbf{P}^{(c)} = [p^{(c)}_{ij}]$ *is the transition matrix of the uniformized chain c. The initial conditions are:*

$$\Gamma_{\mathbf{s}}[0, \mathbf{0}] = \begin{cases} \prod_{c=1}^{M} \pi_{s_c'}(0) \phi_{s_0'}(1) & \textit{for } \mathbf{s} = \mathbf{s}' \\ 0 & \textit{otherwise} \end{cases} \tag{13}$$

where $\mathbf{s}' = \langle s_0', \ldots, s_M' \rangle$ *is the initial state.*

Proof: For any $c \neq 0$ such that $k_c > 0$,

$$\begin{aligned} \Gamma_{\mathbf{s}}[\sigma, \mathbf{k}] &= \prod_{\substack{i=1 \\ i \neq c}}^{M} \pi_{s_i}(k_i) \phi_{s_0}(k_0 + 1) \left[\sum_{s_c^* \in \mathcal{S}_c} \pi_{s_c^*}(k_i - 1) p^{(c)}_{s_c^*, s_c} \right] \\ &= \sum_{s_c^* \in \mathcal{S}_c} p^{(c)}_{s_c^*, s_c} \prod_{\substack{i=1 \\ i \neq c}}^{M} \pi_{s_c^*}(k_c - 1) \pi_{s_i}(k_i) \phi_{s_0}(k_0 + 1) \\ &= \sum_{s_c^* \in \mathcal{S}_c} p^{(c)}_{s_c^*, s_c} \Gamma_{\mathbf{s}^*(c)}[\sigma - 1, \mathbf{k} - \mathbf{1}_c] \end{aligned} \tag{14}$$

The steps are identical for $c = 0$. □

3.2 The Main Recursion

For $\sigma = 0, \ldots, n-1$, we first define subsets of $\mathcal{K}_\sigma$ that will be used for aggregating terms in the expression for $d_{s',s}$.

Definition 1 *For* $1 \leq g \leq \sigma$, $0 \leq l \leq M$, *let*

$$G_{g,l}[\sigma] = \{\mathbf{k} \in \mathcal{K}_\sigma : k_l = g, k_j = 0 \textit{ for } j < l\}.$$

For $g = 0$ *we set* $G_{0,l}[0] = \emptyset$ *for* $0 \leq l \leq M - 1$, *and* $G_{0,M}[0] = \{\langle 0, \ldots, 0 \rangle\}$.

It follows that $G_{g,l}[\sigma]$ is a partition of $\mathcal{K}_\sigma$. As an example, if $\sigma = 3$ and $M = 2$, we have

$$G_{3,0}[3] = \{\langle 3, 0, 0 \rangle\}$$

$$
\begin{aligned}
G_{2,0}[3] &= \{\langle 2,1,0\rangle, \langle 2,0,1\rangle\} \\
G_{1,0}[3] &= \{\langle 1,2,0\rangle, \langle 1,1,1\rangle, \langle 1,0,2\rangle\} \\
G_{3,1}[3] &= \{\langle 0,3,0\rangle\} \\
G_{2,1}[3] &= \{\langle 0,2,1\rangle\} \\
G_{1,1}[3] &= \{\langle 0,1,2\rangle\} \\
G_{3,2}[3] &= \{\langle 0,0,3\rangle\}.
\end{aligned}
$$

Note that $G_{g,M}[\sigma] = \emptyset$ for $g < \sigma$, since $\sum_{c=0}^{M} k_c = \sigma$.

From the definitions above, the following equalities hold.

(a) For $g = 2, \ldots, \sigma$, $0 \leq l \leq M$

$$\{\mathbf{k} : \mathbf{k} \in G_{g,l}[\sigma]\} = \{\mathbf{k} + \mathbf{1}_l : \mathbf{k} \in G_{g-1,l}[\sigma - 1]\}. \tag{15}$$

(b) For $g = 1$, $0 \leq l \leq M - 1$

$$\{\mathbf{k} : \mathbf{k} \in G_{1,l}[\sigma]\} = \bigcup_{h=l+1}^{M-1} \bigcup_{u=1}^{\sigma-1} \{\mathbf{k} + \mathbf{1}_l : \mathbf{k} \in G_{u,h}[\sigma - 1]\} \bigcup \{\mathbf{k} + \mathbf{1}_l : \mathbf{k} \in G_{\sigma-1,M}[\sigma - 1]\}. \tag{16}$$

To show (a) we note that, since $g \geq 2$, then $g - 1 \geq 1$. Thus adding 1 to position l of all vectors $\mathbf{k} \in G_{g-1,l}[\sigma - 1]$ clearly gives all vectors in $G_{g,l}[\sigma]$, since the new k_l is equal to g. To see that the second equality holds, we note that, by definition, all vectors in $G_{u,h}[\sigma - 1]$ (for $l+1 \leq h \leq M-1$ and $1 \leq u \leq \sigma-1$) have entries $k_0 = k_1 = \cdots = k_l = 0$. Therefore, adding 1 in position k_l of all vectors in these sets will generate all possible vectors with $k_l = 1$ (and $k_0 = k_1 = \cdots = k_{l-1} = 0$). The last term follows since $G_{u,M}[\sigma - 1] = \emptyset$ for $u < \sigma - 1$.

We now develop a recursion over the partitions of vectors $\mathbf{k}$, instead of considering each vector in isolation. This motivates the following definition.

Definition 2 *For* $1 \leq g \leq \sigma$, $0 \leq l \leq M$, *let*

$$\Omega_{\mathbf{s}}[g, l, \sigma] = \sum_{\mathbf{k} \in G_{g,l}[\sigma]} f(\sigma, \mathbf{k}) \Gamma_{\mathbf{s}}[\sigma, \mathbf{k}]. \tag{17}$$

Note that $\Omega_{\mathbf{s}}[g, l, \sigma] = 0$ for $\langle s_0, \ldots, s_{l-1}\rangle \neq \langle s'_0, \ldots, s'_{l-1}\rangle$. In this case, $G_{g,l}[\sigma] = \emptyset$, since any vector $\mathbf{k} \in G_{g,l}[\sigma]$ must satisfy $k_0 = \cdots = k_{l-1} = 0$, i.e., the Markov chains corresponding to the first l entries must remain in their initial states for such a $\mathbf{k}$.

We now show that $\Omega_{\mathbf{s}}$ can be calculated by a simple recursion.

Lemma 3 *Let* $\mathbf{s} \in \mathcal{S}$, $\mathbf{s} = \langle s_0, \ldots, s_M \rangle$, $\mathbf{s}^*(c) = \langle s_0, \ldots, s_{c-1}, s_c^*, s_{c+1}, \ldots, s_M \rangle$. *Furthermore, let* s_j' *be the initial state of Markov chain* j, *and* $\mathbf{s}'$ *be the initial state of the combined Markov chain. For* $\sigma = 0, 1, \ldots$, $1 \leq g \leq \sigma$, *and* $0 \leq l \leq M$

$$\Omega_{\mathbf{s}}[g, l, \sigma] = \begin{cases} \sigma \dfrac{\Lambda_l/\gamma}{g} \sum_{s_l^* \in \mathcal{S}_l} p_{s_l^*, s_l}^{(l)} \Omega_{\mathbf{s}^*(l)}[g-1, l, \sigma-1] & 2 \leq g \leq \sigma \\ \sigma(\Lambda/\gamma) p_{s_l', s_l}^{(l)} \Upsilon_{\mathbf{s}'(l)}[\sigma-1, l+1, \sigma-1] & g = 1 \end{cases} \tag{18}$$

where $\Upsilon[g, l, \sigma]$ *is calculated by the recursion*

$$\Upsilon_{\mathbf{s}}[g, l, \sigma] = \begin{cases} \Upsilon_{\mathbf{s}}[g-1, l, \sigma] + \Omega_{\mathbf{s}}[g, l, \sigma] & \text{if } g > 1 \\ \Upsilon_{\mathbf{s}}[\sigma, l+1, \sigma] + \Omega_{\mathbf{s}}[1, l+1, \sigma] & \text{if } g = 1 \end{cases} \tag{19}$$

The initial conditions are ($\sigma = g = 0$)

$$\Omega_{\mathbf{s}}[0, M, 0] = \begin{cases} \Gamma_{\mathbf{s}}[0, 0] & \text{for } \mathbf{s} = \mathbf{s}' \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Proof: Assume that $2 \leq g \leq \sigma$, $0 \leq l \leq M$. Using (10), (12) and Definition 2,

$$\begin{aligned} \Omega_{\mathbf{s}}[g, l, \sigma] &= \sum_{\mathbf{k} \in G_{g,l}[\sigma]} \sigma \frac{\Lambda_l/\gamma}{k_l} f(\sigma-1, \mathbf{k}-\mathbf{1}_l) \sum_{s_l^* \in \mathcal{S}_l} p_{s_l^*, s_l}^{(l)} \Gamma_{\mathbf{s}^*(l)}[\sigma-1, \mathbf{k}-\mathbf{1}_l] \\ &= \sigma \frac{\Lambda_l/\gamma}{g} \sum_{s_l^* \in \mathcal{S}_l} p_{s_l^*, s_l}^{(l)} \sum_{\mathbf{k} \in G_{g,l}[\sigma]} f(\sigma-1, \mathbf{k}-\mathbf{1}_l) \Gamma_{\mathbf{s}^*(l)}[\sigma-1, \mathbf{k}-\mathbf{1}_l] \\ &= \sigma \frac{\Lambda_l/\gamma}{g} \sum_{s_l^* \in \mathcal{S}_l} p_{s_l^*, s_l}^{(l)} \sum_{\mathbf{k} \in G_{g-1,l}[\sigma-1]} f(\sigma-1, \mathbf{k}) \Gamma_{\mathbf{s}^*(l)}[\sigma-1, \mathbf{k}] \end{aligned} \tag{21}$$

where the last equality holds since $\{\mathbf{k}-\mathbf{1}_l : \mathbf{k} \in G_{g,l}[\sigma]\} = \{\mathbf{k} : \mathbf{k} \in G_{g-1,l}[\sigma-1]\}$ from (15). Applying Definition 2 to the second sum in the last equality of (21), we obtain the first part of the lemma.

For $g = 1$ and $0 \leq l \leq M$

$$\begin{aligned} \Omega_{\mathbf{s}}[1, l, \sigma] &= \sum_{\mathbf{k} \in G_{1,l}[\sigma]} \sigma(\Lambda_l/\gamma) f(\sigma-1, \mathbf{k}-\mathbf{1}_l) \sum_{s_l^* \in \mathcal{S}_l} p_{s_l^*, s_l}^{(l)} \Gamma_{\mathbf{s}^*(l)}[\sigma-1, \mathbf{k}-\mathbf{1}_l] \\ &= \sigma(\Lambda_l/\gamma) \sum_{s_l^* \in \mathcal{S}_l} p_{s_l^*, s_l}^{(l)} \sum_{\mathbf{k} \in G_{1,l}[\sigma]} f(\sigma-1, \mathbf{k}-\mathbf{1}_l) \Gamma_{\mathbf{s}^*(l)}[\sigma-1, \mathbf{k}-\mathbf{1}_l]. \end{aligned}$$

Thus

$$\begin{aligned} \Omega_{\mathbf{s}}[1, l, \sigma] &= \sigma(\Lambda_l/\gamma) \sum_{s_l^* \in \mathcal{S}_l} p_{s_l^*, s_l}^{(l)} \sum_{h=l+1}^{M-1} \sum_{u=1}^{\sigma-1} \sum_{\mathbf{k} \in G_{u,h}[\sigma-1]} f(\sigma-1, \mathbf{k}) \Gamma_{\mathbf{s}^*(l)}[\sigma-1, \mathbf{k}] \\ &\quad + \sum_{\mathbf{k} \in G_{\sigma-1,M}[\sigma-1]} f(\sigma-1, \mathbf{k}) \Gamma_{\mathbf{s}^*(l)}[\sigma-1, \mathbf{k}], \end{aligned} \tag{22}$$

where this last equality holds since

$$\{\mathbf{k}-\mathbf{1}_l : \mathbf{k} \in G_{1,l}[\sigma]\} = \bigcup_{h=l+1}^{M-1} \bigcup_{u=1}^{\sigma-1} \{\mathbf{k} : \mathbf{k} \in G_{u,h}[\sigma-1]\} \bigcup \{\mathbf{k} : \mathbf{k} \in G_{\sigma-1,M}[\sigma-1]\}$$

from (16). For $\mathbf{k} \in G_{u,h}[\sigma-1]$, $\Gamma_{\mathbf{s}^*(l)}[\sigma-1,\mathbf{k}] = 0$ if $s_l^* \neq s_l'$ and $h > l$. This is a consequence of the observation made previously after Definition 2. Applying this observation and that definition, we obtain

$$\Omega_{\mathbf{s}}[1,l,\sigma] = \sum_{h=l+1}^{M-1} \sum_{u=1}^{\sigma-1} \Omega_{\mathbf{s}'(l)}[u,h,\sigma-1] + \Omega_{\mathbf{s}'(l)}[\sigma-1,M,\sigma-1]. \tag{23}$$

We now define

$$\Upsilon_{\mathbf{s}}[g,l,\sigma] = \sum_{h=l}^{M-1} \sum_{u=1}^{g} \Omega_{\mathbf{s}}[u,h,\sigma-1] + \Omega_{\mathbf{s}}[\sigma-1,M,\sigma-1]. \tag{24}$$

Substituting (24) into (23), we obtain the second part of the lemma. The recursion for $\Upsilon_{\mathbf{s}}[g,l,\sigma]$ is easily obtained from the definition above. The initial conditions also follow immediately. $\square$

When $t = T_j$, the calculations are much simpler, since we can treat each independent chain individually as if in isolation. We then simply multiply the resulting probabilities together. Alternatively, we can obtain similar recursions as above by following the same steps as before.

We now calculate the entries of the jth block row of $\mathbf{H}$ (the entries corresponding to mini-interval Δ_j). In some models the final state of a transition from $\mathbf{s}'$ to $\mathbf{s}$ depends on whether or not the length δ_j of the mini-interval Δ_j is less than or equal to T_j, i.e., whether or not φ_j was disabled ($j \neq 0$). When the state at the end of Δ_j is the same independent of the value of δ_j, then we need only evaluate the probability of reaching an absorbing state of chain $\mathcal{W}^{(j)}$ by time T_j, which is easily obtained from uniformization by conditioning on the number of transitions by time T_j.

For $\delta_j < T_j$ we have, from (8), (9), (11) and Definition 2

$$d_{s',s}^{(j)} = \sum_{n=1}^{\infty} e^{-\gamma T_j} \frac{(\gamma T_j)^n}{n!} \left(\frac{\Lambda_0}{\gamma}\right) \sum_{\sigma=0}^{n-1} \Upsilon_{\mathbf{s}}[\sigma,0,\sigma] \tag{25}$$

Given a specified error tolerance ϵ, each term in the infinite series above is bounded by the corresponding Poisson term. Thus an upper limit $N = N(\epsilon)$ can be determined in advance to ensure that the calculated values will be within ϵ of the actual value.

It now remains to obtain the entries of the transition probability matrix $\mathbf{H}$ corresponding to mini-interval Δ_0. We recall that no deterministic event is enabled during Δ_0, and the interval ends when any event φ_j is enabled. Therefore, we can construct a chain $\mathcal{U}$ which contains all states with no deterministic event enabled plus a set of absorbing states for each event φ_j. The entry $d^{(0)}_{s',s}$ is the probability, starting from s', that the system reaches an absorbing state of $\mathcal{U}$ corresponding to a deterministic transition being enabled. Since these probabilities may be calculated directly using standard techniques, we omit the development here.

Recall that $\mathcal{S}'^{(j)}$, $j = 1, \ldots, E$, is the subset of states for which event φ_j is enabled, and $\mathcal{S}'^{(0)}$ is the set of states for which a deterministic event is disabled but no deterministic event is enabled. Then $\mathbf{D}^{(j,k)}$ consists of entries $d^{(j)}_{s',s}$ of the jth block row of $\mathbf{H}$ for which $s \in \mathcal{S}'^{(k)}$. The diagonal elements of $\mathbf{H}$ for $j \neq 0$ represent the probability of a transition from $s' \in \mathcal{S}'^{(j)}$ to $s \in \mathcal{S}'^{(j)}$, and they indicate that the deterministic transition φ_j was immediately re-enabled after the jump. Note that the matrix $\mathbf{H}$ has the same dimension as the rate matrix of the original model if all events are exponential. Furthermore the special structures of $\mathbf{H}$ can be exploited, as we discuss later. Solving $\boldsymbol{\beta} = \boldsymbol{\beta}\mathbf{H}$ gives the stationary probability vector for the embedded Markov chain $\mathcal{Y}$.

4 Calculating Measures of Interest

In the previous section we obtained the steady state vector $\boldsymbol{\beta}$ for the embedded Markov chain $\mathcal{Y}$. In this section we consider computation of measures of interest based on the solution for the embedded Markov chain.

Let $P_{\mathcal{R}}$ be the limiting probability as $t \to \infty$ that the process $\mathcal{X}$ is in a particular subset $\mathcal{R} \subset \mathcal{S}$ of states, i.e., $P_{\mathcal{R}} = \lim_{t\to\infty} P[X(t) \in \mathcal{R}]$. For example, if the process $\mathcal{X}$ models a system which consists of one or more queues, $\mathcal{R}$ might be the set of states where a certain queue contains a particular number of customers. If an availability model is analyzed, then $\mathcal{R}$ may be the set of states that represent a given number of failed components. Define R_s to be a reward equal to the cumulative time that the process $X(t)$ spends in the set of states $\mathcal{R}$ during the time between two embedded points, say, η_{k-1} and η_k, given an initial state of s. Next, let $R_{s'}(k)$ be the total reward (total time in $\mathcal{R}$) during $(0, \eta_k)$ given that $X(0) = s'$. From Theorem 7.14 of [13], we have for all $s' \in \mathcal{S}'$, the state space of $\mathcal{Y}$

$$\lim_{k\to\infty} \frac{R_{s'}(k)}{k} \overset{\text{w.p.1}}{=} \sum_{s\in\mathcal{S}'} E[R_s]\beta_s. \tag{26}$$

We also define $R(t)$ to be the amount of time $X(t)$ spends in $\mathcal{R}$ during $(0, t)$. We have

$$P_{\mathcal{R}} \overset{\text{w.p.1}}{=} \lim_{t\to\infty} \frac{R(t)}{t}. \tag{27}$$

From (26) and (27) it can be shown that

$$P_{\mathcal{R}} = \frac{\sum_{s\in\mathcal{S}'} E[R_s]\beta_s}{\sum_{s\in\mathcal{S}'} E[L_s]\beta_s}, \tag{28}$$

where L_s is the length of time between embedded points given that the initial state is s (see [4, 5, 7]).

We may write this expression in terms of expected values related to the mini-intervals Δ_j, $j = 0, \ldots, E$. Let $U_s^{(j)}$ be the time spent in $\mathcal{R}$ during a Δ_j mini-interval which starts in state s, and let $\delta_s^{(j)}$ be the length of this mini-interval. Since the $\mathcal{S}'^{(j)}$ are a partition of the state space $\mathcal{S}'$, we can express (27) as

$$P_{\mathcal{R}} = \frac{\sum_{j=0}^{E} \sum_{s\in\mathcal{S}'^{(j)}} E[U_s^{(j)}]\beta_s}{\sum_{j=0}^{E} \sum_{s\in\mathcal{S}'^{(j)}} E[\delta_s^{(j)}]\beta_s}. \tag{29}$$

It remains to find the expected values in (29). We first note that the $E[\delta_s^{(j)}]$ are independent of $\mathcal{R}$, and they are easier to obtain than the $E[U_s^{(j)}]$. If we condition on the initial state $s_0(j)$ of the mini-interval Δ_j for the preemptive chain $\mathcal{W}^{(j)}$ (see equation (6)), we obtain, for $j = 1, \ldots, E$,

$$E[\delta^{(j)} \mid s_0(j)] = T_j - T_j \sum_{n=0}^{\infty} e^{-\Lambda_j T_j} \frac{(\Lambda_j T_j)^n}{n!} \left\{ \frac{\sum_{m=0}^{n} \pi_{\text{abs}}(m, s_0(j))}{n+1} \right\}. \tag{30}$$

where $\boldsymbol{\pi}(m, s_0(j)) = \boldsymbol{\pi}(m-1, s_0(j))\mathbf{W}^{(j)}$ with $\mathbf{W}^{(j)}$ the transition matrix of the uniformized chain corresponding to $\mathcal{W}^{(j)}$, $\boldsymbol{\pi}(0, s_0(j)) = \mathbf{1}_{s_0(j)}$. We note that it is not necessary to calculate equation (30) for all possible initial states $s_0(j)$. Instead, it is easy to show that the computations need only to be carried out using the uniformized chain with an initial distribution $\mathbf{b}^{(j)}$, where this vector is given by

$$\mathbf{b}_{s^*}^{(j)} = \sum_{\substack{s \in \mathcal{S}'^{(j)} \\ s_0 = s^*}} \beta_s.$$

From this observation, the denominator in equation (29) can be obtained as

$$\sum_{j=0}^{E} \sum_{s\in\mathcal{S}'^{(j)}} E[\delta_s^{(j)}]\beta_s = \sum_{j=0}^{E} T_j - \sum_{j=0}^{E} T_j \sum_{n=0}^{\infty} e^{-\Lambda_j T_j} \frac{(\Lambda_j T_j)^n}{n!} f(n) \tag{31}$$

where

$$f(n) = \frac{\sum_{m=0}^{n} \pi_{\text{abs}}(m, \mathbf{b}^{(j)})}{n+1}$$

can be calculated recursively [6] by

$$f(n+1) = \frac{n+1}{n+2} f(n) + \frac{\pi_{\text{abs}}(n+1, \mathbf{b}^{(j)})}{n+2}.$$

Note that $f(n) \leq 1$ for all n.

When $j = 0$, no deterministic event can be enabled during the interval Δ_0. We recall that, in this case, mini-interval Δ_0 ends when any deterministic event becomes enabled. Therefore, the expected length of $\delta_s^{(0)}$ is the mean time to reach the absorbing states of the chain $\mathcal{W}^{(0)}$ from the initial state s, which can be easily calculated by standard techniques. For instance, one can replace the chain with absorbing states by another chain such that, whenever the original chain reaches an absorbing state, it is restarted from the initial state. The expected value of $\delta_s^{(0)}$ is then given by

$$E[\delta_s^{(0)}] = \frac{1}{\pi_f} - 1,$$

where π_f is the steady state probability that the modified chain is in any state corresponding to an absorbing state of the original chain.

We now consider the calculation of $E[U_s^{(j)}]$ in (29), for $j = 1, \ldots, E$, which corresponds to the expected time during mini-interval Δ_j that the system spends in the subset $\mathcal{R}$. We recall that the behavior of the system during Δ_j is given by $M_j + 1$ independent subchains with preemptive chain $\mathcal{W}^{(j)}$. In general, the expected time in $\mathcal{R}$ may depend on the behavior of all these chains. Recall that $\mathcal{V}^{(j)}$ is the chain formed by all states in which φ_j is enabled including absorbing states. We wish to find the expected time during mini-interval Δ_j that chain $\mathcal{V}^{(j)}$ spends in $\mathcal{R}$. Using the uniformized chain corresponding to $\mathcal{V}^{(j)}$, the expected time can be found by performing transient analysis over an interval of length T_j. Conditioning on the number of transitions during $(0, T_j)$, we have (see [6, 10])

$$\sum_{s \in \mathcal{S}'^{(j)}} E[U_s^{(s)}]\beta_s = T_j \sum_{n=0}^{\infty} e^{-\Lambda_j T_j} \frac{(\Lambda_j T_j)^n}{n!} \left\{ \frac{\sum_{m=0}^{n} \pi_{\mathcal{V}^{(j)}}(m, \mathbf{b}^{(j)})}{n+1} \right\}, \tag{32}$$

where $\boldsymbol{\pi}(m, \mathbf{b}^{(j)}) = \boldsymbol{\pi}(m-1, \mathbf{b}^{(j)})\mathbf{V}^{(j)}$ (with $\mathbf{V}^{(j)}$ the transition matrix of the uniformized chain corresponding to $\mathcal{V}^{(j)}$), $\boldsymbol{\pi}(0, \mathbf{b}^{(j)}) = \mathbf{b}^{(j)}$ and $\mathbf{b}^{(j)}$ is the initial distribution vector for $\mathcal{V}^{(j)}$.

When the measure of interest depends on only one of the independent subchains, then computational savings are possible, since the cardinality of the chains involved in the calculations is potentially much less than the cardinality of chain $\mathcal{V}^{(j)}$ which involves the combined states of all chains. Let $\mathcal{W}_c^{(j)}$ be the cth independent subchain during Δ_j. We first assume that $\mathcal{R}$ is a subset of the states of independent subchain $\mathcal{W}_c^{(j)}$. In our notation, chain $\mathcal{W}_0^{(j)}$

is chain $\mathcal{W}^{(j)}$, which controls the length of mini-interval Δ_j. This case arises, for example, if we are interested in the marginal queue length distribution of the number of customers of a queue in a model with several interacting queues (see [7]). When $c \neq 0$, the amount of time during mini-interval Δ_j that the system spends in $\mathcal{R}$ depends on the initial distribution of chain $\mathcal{W}_c^{(j)}$ as well as on δ_j, the length of mini-interval Δ_j (recall that δ_j depends on chain $\mathcal{W}^{(j)}$.) Uniformizing $\mathcal{W}_c^{(j)}$ and conditioning on δ_j we obtain

$$E[U^{(j)}|s'_c, s'_0] = \int_0^{T_j} \sum_{n=0}^{\infty} \frac{t \sum_{m=0}^{n} \pi_c(m, s'_c)}{n+1} \left[\sum_{l=1}^{\infty} e^{-\Lambda_0 t} \Lambda_0 \frac{(\Lambda_0 t)^{l-1}}{(l-1)!} \phi_{\text{abs}}(l, s'_0) \right] dt \qquad (33)$$

where $\phi_{\text{abs}}(l, s'_0)$ is the probability that the uniformized chain $\mathcal{W}_0^{(j)}$ is absorbed at exactly step l starting from initial state s'_0. Performing the integration and following similar steps used to obtain (8), we have

$$E[U^{(j)}|s'_c, s'_0] = \frac{1}{\gamma_c} \sum_{n=1}^{\infty} e^{-\gamma T_j} \frac{(\gamma_c T_j)^n}{n!}$$
$$\sum_{m=1}^{n} \sum_{\sigma=0}^{m-1} \left(\frac{\Lambda_c}{\gamma_c}\right)^{\sigma} \left(\frac{\Lambda_0}{\gamma_c}\right)^{m-\sigma-1} \frac{m!}{(m-1-\sigma)!(\sigma+1)!} \pi_c(m-\sigma, s'_c) \phi_{\text{abs}}(m-\sigma, s'_0) \quad (34)$$

where $\gamma_c = \Lambda_c + \Lambda_0$.

Note that we do not need to calculate the measure above for all possible pairs of states with components s'_c and s'_0. Specifically, we can aggregate states with the same value of s'_c and s'_0, i.e., let

$$b_{s'_c}^{(c,0)}(s'_0) = \sum_{s \in \mathcal{S}_{s'_c, s'_0}} \beta_s^{(j)}$$

where $\mathcal{S}_{s'_c, s'_0} = \{\langle s_0, s_1, \ldots, s_M \rangle \in \mathcal{S} : s_0 = s'_0, s_c = s'_c\}$. Then

$$\sum_{s \in \mathcal{S}} E[U_s^{(j)}] \beta_s^{(j)} = \frac{1}{\gamma_c} \sum_{s'_c \in \mathcal{W}_c^j} \sum_{n=1}^{\infty} e^{-\gamma_c T_j} \frac{(\gamma_c T_j)^n}{n!} \qquad (35)$$
$$\sum_{m=1}^{n} \sum_{\sigma=0}^{m-1} \left(\frac{\Lambda_c}{\gamma_c}\right)^{\sigma} \left(\frac{\Lambda_0}{\gamma_c}\right)^{m-1-\sigma} \frac{m!}{(m-1-\sigma)!(\sigma+1)!} \pi_c(m-\sigma, \mathbf{b}^{(c,0)}(s'_c)) \phi_{\text{abs}}(m-\sigma, \mathbf{b}^{(c,0)}(s'_c))$$

where $\pi_c(m, \mathbf{b}^{(c,0)}(s'_c)) = \pi_c(m-1, \mathbf{b}^{(c,0)}(s'_c)) \mathbf{W}^{(j)}$ and $\pi_c(0, \mathbf{b}^{(c,0)}(s'_c)) = \mathbf{b}_{s'_c}^{(c,0)}$. Recursions similar to those developed for equation (8) can be found.

5 Computational Complexity

In this section we briefly discuss the main elements of the computational cost to calculate the measures of interest. We first discuss the calculation of the entries of $\mathbf{D}^{(j,k)}$.

The entries $d_{s',s}$ of $\mathbf{D}^{(j,k)}$ are obtained from equation (25). The main cost of (25) is the cost to obtain $\Upsilon_{\mathbf{s}}[\sigma, 0, \sigma]$ from (19) which in turn requires a recursion to obtain $\Omega_s[g, l, \sigma]$ given by (18). Since $1 \leq g \leq n$ and $1 \leq l \leq M$ it is easy to see from (18) that the number of operations necessary to obtain $\Omega_s[\,]$ is $O(\sum_i d_i S_i N^2)$ where S_i and d_i is the cardinality and number of non-zero entries of independent chain i, respectively. Note that these calculations are organized so to take advantage of the decomposition of the model into independent subchains and involve chains that have state space cardinalities usually much smaller than the entire chain if decomposition is not performed.

The storage requirements are such that only N vectors of $\Omega_s[\,]$ values for s varying over the state space of an independent chain.

It is worth noting that the transition $d_{s',s}$ is affected by the value T_j only through the Poisson term in (25). Therefore, $d_{s',s}$ can be obtained for different values of T_j with little extra effort.

It is also interesting to observe that, in many problems, matrix $\mathbf{H}$ have special structures that can be exploited in the final solution. For instance, assume that, once a deterministic event is disabled it cannot be re-enabled immediately, but only after an exponential event or another deterministic event is executed. Then the diagonal blocks of $\mathbf{H}$ are all zero. If, in addition, other special structure exist as in the example we presented, significant savings can be obtained.

Another important observation is concerned with the number of states of the embedded Markov chain $\mathbf{H}$. It is possible, for instance, that the number of states in $\mathbf{H}$ is significantly less than the number of states of the entire model. As an example, consider a model with one deterministic transition and assume that, when this transition is disabled, the model always goes to one of two possible states. Furthermore, assume that the deterministic transition is always enabled from a single state. Then $\mathbf{H}$ has 3 states, independently of the process during the time the deterministic transition is enabled. An application of this kind of behavior can be found in [5].

6 Extensions

In this section we discuss several extensions to the basic model which can also be handled by simple variations of the approach developed in previous sections. In the first extension we assume that, when a deterministic event expires, a given exponential event is allowed to complete before the system "fires" the deterministic event (i.e. before the system executes the transition associated with the firing of the deterministic event). To illustrate how this can be useful, assume that the system model is composed of M queues which are served alternately (a polling model). Each queue is served until it empties or until a given timeout expires. (The timeout is restarted whenever the server starts serving a queue.) If, when the timeout expires, the customer in queue is allowed to complete service before the server switches to another queue, i.e., a "service overrun" occurs, then to include this in the model requires the extension mentioned. We note that, in the case of exponential service times, the remaining service time is again exponential. Therefore, during the overrun, we can use the same approach developed previously for embedded points during intervals with no deterministic transitions enabled. In other words, this case can be handled by allowing an exponential transition to be enabled, conditioned on the event that the timeout expired.

Now consider the case where, during a mini-interval in which a deterministic event is enabled, certain exponential events are not allowed to be activated if the deterministic event is close to firing. We give two simple examples where this extension is used in a model. First, consider the polling model described above. If a customer starts service near the end of the timeout interval, it is unlikely that it can finish service before the timeout expires. Thus it may be advantageous to switch the service to the other queue near the end of the timeout period instead of accepting another customer into service. We consider the following extension. Assume that there is a constant ω_j associated with timeout T_j for queue j such that, if the timeout has exceeded $T_j - \omega_j$, then no other customer is allowed in service. The server then switches to another queue as soon the timeout expires or the customer being served finishes.

As a second example, we consider an availability model, specifically a system which undergoes scheduled maintenance at fixed time intervals, say each T units of time. Further assume that, if the system stops working, an immediate maintenance call is requested and the next regular maintenance call will be rescheduled for T units hence. However, if the system fails close enough to the next regularly scheduled maintenance, then no special request in made and the maintenance schedule is not altered.

In order to facilitate the description of the changes necessary in our solution to handle the extension, consider the polling example above. Suppose that there is a customer in service at time $T_j - \omega_j$. Then, either the customer finishes service before the timeout expires and the system immediately switches to another queue, or the customer does not finish service and the timeout fires and preempts the customer. This behavior is similar to the case where overruns can occur, for a timeout value of $T_j - \omega_j$. Furthermore, unlike the "overrun case", the length of the remaining interval is an exponential random variable truncated at ω_j. In summary the modifications necessary are: (a) assume that the timeout has value $T_j - \omega_j$ and; (b) consider a new deterministic event φ'_j which is enabled concurrently with an exponential event.

We conclude this section by mentioning that the approach we described is not limited to deterministic transitions. In [9] Grassmann showed that the uniformization technique can be used not only during an interval of fixed length, but when the interval is a random variable with density equal to piecewise uniform or piecewise polynomial, etc. Grassman's results can be applied directly with our results to extend teh computational algorithms to general event time distributions.

7 Example

In this section we present a simple example to illustrate the approach discussed in previous sections. We choose a polling system with three queues that are served cyclically by a single server. Each queue has a finite waiting room capacity. Once a server arrives at queue j it serves the customers queued until it empties or until a timeout of length T_j (which is set at the beginning of the service) expires. We also assume that there is a switchover time of constant length. In summary, we consider an exhaustive polling model with server timeouts. There are six deterministic events in this example. Three of them are the timeouts for each queue and three correspond to the switchover times. The matrix $\mathbf{H}$ for this polling system has a special structure as indicated below.

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{D}^{(1,2)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}^{(2,3)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}^{(3,4)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}^{(4,5)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}^{(5,6)} \\ \mathbf{D}^{(6,1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Furthermore, for each mini-interval Δ_j, there are 3 independent chains that can be easily identified. For instance assume that queue j is being served. Then, the chain which gives the number of customers at that queue corresponds to chain $\mathcal{W}_0^{(j)}$ since, when this queue empties, the server switches and the timeout for that queue is disabled. Chain $\mathcal{W}_1^{(j)}$ and $\mathcal{W}_2^{(j)}$ are birth processes that correspond to the arrivals of the other queues when queue j is being served. These three queues are clearly independent. Several special structures exist that can be exploited to obtain important savings (see [7] for details).

In our polling system model we can study, for example, the effect of varying the timeout values for each queue. To facilitate the comparison we assume that the sources are identical. Figure 1 plots the average delay of customers from each queue versus the total offered load. Three sets of timeout values are used: 6-3-2, 7-2-2 and 4-4-3. The offered load varies from 30% to 90% of the capacity of the server, the queues have a finite buffer of size 10, the expected service time is equal to 0.5 msec and the switchover times are equal to 0.01 msec. From the figure it is evident the tradeoffs involved when choosing the timeout values. For

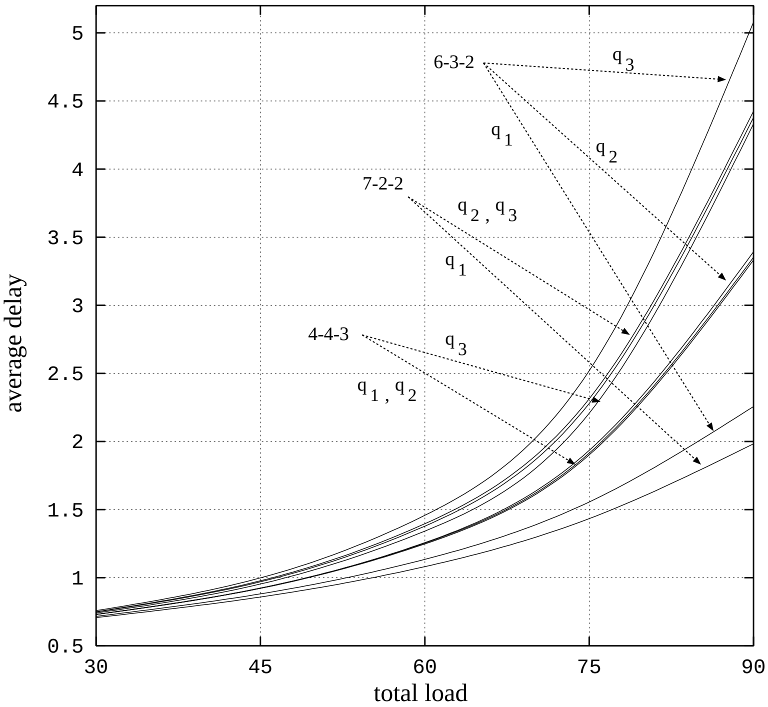

Figure 1: The effect of varying the timeout values.

instance, the second queue is practically not affected when we change the timeout values from 4-4-3 to 6-3-2, while the first queue is given a higher fraction of the server at the expense of the third queue.

8 Conclusions

We have presented a solution proceedure and efficient computational algorithms for solving a class of non-Markovian processes and computing performance measures of interest. This class of models was also considered in [4]. Our approach is different in that uniformization is used as the basis for the computations while the computation methods discussed in [4] involve numerical solution of differential equations. Furthermore we presented simple recursions which take advantage of "independent subchains" which are common in many models.

Uniformization has several potential benefits as an alternative computational approach; these benefits often result from the relatively simple probabilistic interpretation of the expressions. For example, in this paper we were led to efficient recursive expressions for the computations, extensions to the basic model that were easy to incorporate into the computation, and to computational efficiency from exploiting special structure in the models.

References

[1] M. Ajmone Marsan and G. Chiola. *Lecture Notes in Computer Science: Advances in Petri Nets*, chapter On Petri Nets with Deterministic and Exponentially Distributed Firing Times, pages 132–145. Springer, 1987.

[2] M. Ajmone Marsan, G. Chiola, and A. Fumagalli. *Lecture Notes in Computer Science: Advances in Petri Nets*, chapter Improving the Efficiency of the Analysis of DSPN Models, pages 30–50. Springer, 1987.

[3] S. Berson, E. de Souza e Silva, and R.R. Muntz. *Numerical Solution of Markov Chains*, chapter An Object Oriented Methodology for the Specification of Markov Models, pages 11–36. Marcel Dekker, Inc., 1991.

[4] H. Choi, V. Kulkarni, and K. Trivedi. Markov regenerative stochastic petri nets. In *Proc. Performance '93*, pages 339–356, 1993.

[5] E. de Souza e Silva and H.R. Gail. Analyzing scheduled maintenance policies for repairable computer systems. *IEEE Trans. on Computers*, 39(11):1309–1324, 1990.

[6] E. de Souza e Silva and H.R. Gail. Performability analysis of computer systems: from model specification to solution. *Performance Evaluation*, 14:157–196, 1992.

[7] E. de Souza e Silva, H.R. Gail, and R.R. Muntz. Polling systems with server timeouts. Technical report, UCLA CS Department, CSD-930026, Los Angeles, CA, 1993.

[8] R. German and C. Lindemann. Analysis of stochastic petri nets by the method of supplementary variables. In *Proc. Performance '93*, pages 320–338, 1993.

[9] W.K. Grassmann. The GI/PH/1 queue: a method to find the transition matrix. *Infor*, 20(2):144–156, 1982.

[10] W.K. Grassmann. Means and variances of time averages in Markovian environments. *European Journal of Operational Research*, 31:132–139, 1987.

[11] W.K. Grassmann. Finding transient solutions in Markovian event systems through randomization. In W.J. Stewart, editor, *Numerical Solution of Markov Chains*, pages 357–371. Marcel Dekker, Inc., 1991.

[12] D. Gross and D.R. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32(2):343–361, 1984.

[13] D.P. Heyman and M.J. Sobel. *Stochastic Models in Operations Research, Volume I.* McGraw-Hall, 1982.

[14] A. Jensen. Markoff chains as an aid in the study of Markoff processes. *Skandinavsk Aktuarietidskrift*, 36:87–91, 1953.

[15] C. Lindemann. An improved numerical algorithm for calculating steady-state solutions of deterministic and stochastic Petri net models. In *Proceedings of the 4th International Workshop on Petri-nets and Performance Models*, pages 176–184, 1991.