A Numerical Method for the Evaluation of the Distribution of Cumulative Reward Till Exit of a Subset of Transient States of a Markov Reward Model

Juan A. Carrasco and V. Suñé
Departament d'Enginyeria Electrònica
Universitat Politècnica de Catalunya
Diagonal 647, plta. 9
08028 Barcelona, Spain

Except for formatting details, this version matches exactly the version published with the same title and authors in *IEEE Trans. on Dependable and Secure Computing*, vol. 8, no. 6, 2011, pp. 798–809

Abstract

Markov reward models have interesting modeling applications, particularly those addressing fault-tolerant hardware/software systems. In this paper, we consider a Markov reward model with a reward structure including only reward rates associated with states, in which both positive and negative reward rates are present and null reward rates are allowed, and develop a numerical method to compute the distribution function of the cumulative reward till exit of a subset of transient states of the model. The method combines a model transformation step with the solution of the transformed model using a randomization construction with two randomization rates. The method introduces a truncation error, but that error is strictly bounded from above by a user-specified error control parameter. Further, the method is numerically stable and takes advantage of the sparsity of the infinitesimal generator of the transformed model. Using a Markov reward model of a fault-tolerant hardware/software system, we illustrate the application of the method and analyze its computational cost. Also, we compare the computational cost of the method with that of the (only) previously available method for the problem. Our numerical experiments seem to indicate that the new method can be efficient and that for medium-size and large models can be substantially faster than the previously available method.

Keywords: Fault tolerance, modeling techniques, Markov reward models, numerical algorithms.

1 Introduction

Markov reward models (MRMs) have interesting modeling applications, particularly those addressing fault-tolerant hardware/software systems. A MRM is a (time homogeneous) continuous-time Markov chain (CTMC) with a reward structure imposed over it. The reward structure may reflect several properties of the system behavior: performance, power consumption, cost, etc. The reward structure may include in general reward rates associated with states and impulse rewards associated with transitions. In this paper, we consider MRMs with a reward structure including only reward rates associated with states. The reward rate r_i associated with state i has the meaning of rate at which the reward is earned while the CTMC is in state i.

In this paper we are concerned with the computation of the distribution function of the cumulative reward till exit of a subset of transient states of a MRM. More specifically, we consider a MRM with an underlying finite CTMC $X = \{X(t); t \geq 0\}$ with infinitesimal generator and state space $\Omega = S \cup \{a\}$, where $S \neq \emptyset$ includes transient states and a is an absorbing state that is eventually entered with probability one, and a reward rate structure r_i, $i \in S$, with $|r_i| < \infty$. Let T denote the time to absorption of X, i.e. $T = \min\{t \geq 0 \, : \, X(t) = a\}$. The measure considered in the paper is

$$F(s) = \Pr\left[R(T) \leq s\right] ,$$

where $R(T) = \int_0^T r_{X(\tau)} \, \mathrm{d}\tau$ is the cumulative reward up to absorption of X.

Let

$$\begin{aligned} S^0 &= \{i \in S \, : \, r_i = 0\} , \\ S^+ &= \{i \in S \, : \, r_i > 0\} , \\ S^- &= \{i \in S \, : \, r_i < 0\} . \end{aligned}$$

A transformation of the CTMC X making easy the computation of $F(s)$ for the case $S^0 = S^- = \emptyset$, $s \geq 0$ was developed in [1]. The modified CTMC Y has same state space and initial probability distribution as X and has an infinitesimal generator with elements $b_{i,j}$, $i \in S$, obtained by dividing by r_i the elements $a_{i,j}$ of the infinitesimal generator of X, and verifies $\Pr[R(T) \leq s] = \Pr[Y(s) = a]$. It follows that $F(s) = \Pr[Y(s) = a]$, which can be computed using standard methods, e.g. ODE solvers or randomization (also called uniformization) [2], [3]. That model transformation was generalized to cover the case $S^- = \emptyset$, $s \geq 0$ in [4], where it was shown how to build a modified CTMC Y with state space $S^+ \cup \{a\}$ such that $R(T)$ has the same probability distribution as $\min\{t \geq 0 \, : \, Y(t) = a\}$, implying $F(s) = \Pr[Y(s) = a]$. The same case was analyzed in [5] for semi-Markov reward models with only reward rates associated with states. There, a model transformation similar in spirit to the one described in [4] was developed and several alternatives were proposed to solve the transformed semi-Markov reward process. The approach in [4] can be easily extended to cover the case $S^+ = \emptyset$, $s \leq 0$ by noting that: 1) inverting the sign of the reward rates maps the case $S^+ = \emptyset$ into the case $S^- = \emptyset$ and makes the cumulative reward up to absorption equal to $-R(T)$, 2) $\Pr[R(T) \leq s] = \Pr[-R(T) \geq -s] = 1 - \Pr[-R(T) < -s]$, 3) for $s = 0$, $\Pr[-R(T) < -s] = 0$, and 4) for $s < 0$, $\Pr[-R(T) < -s] = \Pr[-R(T) \leq -s]$, since, being

$-R(T)$ the minimum time at which the modified CTMC corresponding to the mapped MRM is in state a, $-R(T)$ has a continuous distribution function for values > 0.

To the best of our knowledge, the computation of $F(s)$ for MRMs with a reward structure in which both positive and negative reward rates are present has been considered only in [6]. There, closed-form expressions for the conditional probabilities $\Pr[R(T) \leq s \mid R(T) > 0]$, $s > 0$, $\Pr[-R(T) \leq s \mid R(T) < 0]$, $s > 0$, and closed-form expressions for the probabilities $\Pr[R(T) > 0]$, $\Pr[R(T) < 0]$ were derived for the case $S^0 = \emptyset$, $S^+ \neq \emptyset$, $S^- \neq \emptyset$. These expressions depend on two matrices that are independent of s and that can be computed using an iterative algorithm described in [6]. Using a model transformation similar to the one developed in [4] to "eliminate" the states in S^0 and obtaining the above probabilities on the transformed model, $F(s)$ can be computed easily for the case $S^+ \neq \emptyset$, $S^- \neq \emptyset$.

The reward rate associated with the absorbing state a, r_a, does not have any effect on $F(s)$. We can then assume $r_a = 0$. Doing so, we clearly have $F(s) = \Pr[\int_0^\infty r_{X(\tau)}\,\mathrm{d}\tau \leq s]$ and it can be shown that $\Pr[\int_0^\infty r_{X(\tau)}\,\mathrm{d}\tau \leq s] = \lim_{t\to\infty} \Pr[\int_0^t r_{X(\tau)}\,\mathrm{d}\tau \leq s]$, so, in principle one could estimate $F(s)$ by computing $F_t(s) = \Pr[\int_0^t r_{X(\tau)}\,\mathrm{d}\tau \leq s]$ for t large enough. The distribution function $F_t(s)$, often referred to as performability, a more general concept introduced in [7], has received much attention in the last years and there exist currently several methods for the computation of $F_t(s)$ [8, 9, 10, 11, 12], a method for the computation of $\Pr[\int_0^t r_{X(\tau)}\,\mathrm{d}\tau > s] = 1 - F_t(s)$ [13] (see also [14]), two methods for the computation of the distribution function of the time-averaged cumulative reward, $\Pr[(1/t)\int_0^t r_{X(\tau)}\,\mathrm{d}\tau \leq s] = F_t(ts)$, [15, 16], a method for the computation of $\Pr[(1/t)\int_0^t r_{X(\tau)}\,\mathrm{d}\tau > s] = 1 - F_t(ts)$ [16], and a method for the computation of bounds for $F_t(s)$ [17]. All these methods require $r_i \geq 0$, $i \in \Omega$, and, thus, to apply them in the case $S^+ \neq \emptyset$, $S^- \neq \emptyset$ one has to replace r_i, $i \in \Omega$, by $r'_i = r_i - \min_{j\in S^-} r_j$ and use $F_t(s) = \Pr[\int_0^t r_{X(\tau)}\,\mathrm{d}\tau \leq s] = \Pr[\int_0^t r'_{X(\tau)}\,\mathrm{d}\tau \leq s - t\min_{j\in S^-} r_j]$. We also mention the methods described in [18], which for MRMs satisfying certain conditions allow to compute efficiently bounds for $\Pr[\int_0^t r_{X(\tau)}\,\mathrm{d}\tau > s] = 1 - F_t(s)$ for the case $S^+ \neq \emptyset$, $S^- \neq \emptyset$. However, there does not seem to exist any criterion for selecting t large enough so that $F_t(s)$ is close to $F(s)$. This leaves the method based on the results given in [6] as the only alternative for the computation of $F(s)$ for the case $S^+ \neq \emptyset$, $S^- \neq \emptyset$. As we shall illustrate, the computational cost of that method can be quite high for medium-size and large MRMs. With that motivation, we will develop a new numerical method for the computation of $F(s)$ for the case $S^+ \neq \emptyset$, $S^- \neq \emptyset$. The method combines a model transformation step with the solution of the transformed CTMC model using a randomization construction with two randomization rates. As we shall illustrate, the use of two randomization rates instead of a single one may reduce significantly the computational cost of the method. The method introduces a truncation error, but that error is strictly bounded from above by a user-specified error control parameter. An important feature of the method is that it is numerically stable. Another important feature is that it takes advantage of the sparsity of the infinitesimal generator of the transformed CTMC model. The rest of the paper is organized as follows. The method is developed in Section 2 and its subsections. Using a MRM of a fault-tolerant hardware/software system, in Section 3, we illustrate the application of the method and analyze its computational cost. Also, we compare the computational cost of the method with that of the method based on the results given in [6]. Finally, we present our con-

clusions in Section 4. The Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2010.39, includes two proofs, the theorem establishing the correctness of a generalized randomization construction in which different randomization rates are allowed in different states, and the development and description of numerically stable procedures for the computation of some quantities required by the method.

2 The Method

To develop the method we will proceed by steps. In Section 2.1, we will construct a CTMC with state space $S^+ \cup S^- \cup \{a\}$ such that the difference between the times spent by that CTMC in S^+ and S^- until absorption in state a is a random variable with the same probability distribution as $R(T)$. In Section 2.2, we will analyze the probability distribution of that variable using a randomization construction with two randomization rates. The result will be a closed-form expression for $F(s)$ in the form of a sum with infinite domain whose terms are the products of some probabilities. In Section 2.3, we will show how the infinite sum can be truncated with strictly bounded from above error. In Section 2.4, we will develop computable expressions for the probabilities. In Section 2.5, we will summarize the computational steps of the method and describe some implementation details. Finally, in Section 2.6 we will discuss the numerical stability and computational cost of the method.

Before entering into the details of the method, we introduce some notation. In general, vectors will be understood as column vectors, using the superscript $^{\mathrm{T}}$ to denote the transpose operator. As usual, the identity (resp., all zeroes) matrix will be denoted by $\mathbf{I}$ (resp., $\mathbf{O}$), and an all ones (resp., zeroes) vector by $\mathbf{1}$ (resp., $\mathbf{0}$), in all cases with appropriate dimensions as given by the context. By $\mathbf{1}_c$ we will denote the indicator function returning the value 1 if condition c is satisfied and the value 0 otherwise. We will use $|\cdot|$ to denote the cardinality of a set. For notational convenience, we define an additional subset: $S^{\pm} = S^+ \cup S^-$. The (i,j)th element of a matrix $\mathbf{M}$ will be denoted by $m_{i,j}$, whereas the symbol $\mathbf{M}_{y,z}$ with $y, z \in \{0, +, -, \pm\}$ will denote the submatrix of $\mathbf{M}$ formed by the elements $m_{i,j}$ with $i \in S^y$ and $j \in S^z$. In a slight abuse of notation, $\mathbf{M}_{y,a}$ with $y \in \{0, +, -, \pm\}$ will denote the subvector of $\mathbf{M}$ formed by the elements $m_{i,a}$ with $i \in S^y$. The ith element of a vector $\boldsymbol{v}$ will be denoted by v_i, using $\boldsymbol{v}_y$ with $y \in \{0, +, -, \pm\}$ to denote the subvector of $\boldsymbol{v}$ formed by the elements v_i with $i \in S^y$. Finally, $\mathbf{diag}[v_i]_{i \in G}$ will denote a diagonal matrix with diagonal elements v_i, $i \in G$. Throughout the paper we will use the convention $0^0 = 1$.

2.1 Model Transformation

In this section, we will construct a CTMC $Y = \{Y(t); t \geq 0\}$ with state space $S^{\pm} \cup \{a\}$ such that

$$F(s) = \Pr\left[T^+ - T^- \leq s\right] ,$$

where T^+ (resp., T^-) is the random variable "time spent by Y in S^+ (resp., S^-) until absorption in state a."

The construction of Y is conceptually simple and can be justified following the line of reasoning in [4]. First, we "eliminate" the states in S^0 so that the CTMC spends zero time in them, keeping the sequence of visited states in $S^{\pm}$. Second, we divide by $|r_i|$ the elements $c_{i,j}$, $i \in S^{\pm}$, of the infinitesimal generator $\mathbf{C}$ of the CTMC with states in S^0 eliminated so that holding times in the new CTMC Y have identical probability distributions as the rewards earned in those holding times in the CTMC with states in S^0 eliminated and unscaled infinitesimal generator elements. Let $\mathbf{A}$ and $\boldsymbol{\alpha}$ denote the infinitesimal generator and initial probability distribution vector of X. We assume both partitioned according to $\Omega = S^0 \cup S^{\pm} \cup \{a\}$ and write

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{0,0} & \mathbf{A}_{0,\pm} & \mathbf{A}_{0,a} \\ \mathbf{A}_{\pm,0} & \mathbf{A}_{\pm,\pm} & \mathbf{A}_{\pm,a} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & 0 \end{pmatrix}, \quad \boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\alpha}_{\pm} \\ \alpha_a \end{pmatrix}.$$

Let $\mathbf{R} = \mathbf{diag}[|r_i| + \mathbf{1}_{i=a}]_{i \in S^{\pm} \cup \{a\}}$ with $r_a = 0$. Then, assuming $S^0 \neq \emptyset$, the infinitesimal generator $\mathbf{B}$ and the initial probability distribution vector $\boldsymbol{\beta}$ of Y are

$$\mathbf{B} = \mathbf{R}^{-1}\mathbf{C} = \mathbf{R}^{-1} \begin{pmatrix} \mathbf{C}_{\pm,\pm} & \mathbf{C}_{\pm,a} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_{\pm} \\ \beta_a \end{pmatrix},$$

with

$$\begin{aligned} \mathbf{C}_{\pm,\pm} &= \mathbf{A}_{\pm,\pm} - \mathbf{A}_{\pm,0}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,\pm}\,, \\ \mathbf{C}_{\pm,a} &= \mathbf{A}_{\pm,a} - \mathbf{A}_{\pm,0}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,a}\,, \\ \boldsymbol{\beta}_{\pm}{}^{\mathrm{T}} &= \boldsymbol{\alpha}_{\pm}{}^{\mathrm{T}} - \boldsymbol{\alpha}_0{}^{\mathrm{T}}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,\pm}\,, \\ \beta_a &= \alpha_a - \boldsymbol{\alpha}_0{}^{\mathrm{T}}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,a}\,. \end{aligned}$$

Note that $-\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,\pm}$ gives the probabilities of X exiting subset S^0 through particular states in $S^{\pm}$ given entry in S^0 through particular states and $-\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,a}$ gives the probabilities of exiting subset S^0 through state a given entry in S^0 through particular states. Thus, the terms $-\mathbf{A}_{\pm,0}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,\pm}$ and $-\mathbf{A}_{\pm,0}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,a}$ effectively "eliminate" the states in S^0 with regard to the transition rates, and the terms $-\boldsymbol{\alpha}_0{}^{\mathrm{T}}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,\pm}$ and $-\boldsymbol{\alpha}_0{}^{\mathrm{T}}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,a}$ effectively "eliminate" the states in S^0 with regard to the initial probabilities. Finally, the multiplication by $\mathbf{R}^{-1}$ performs the scaling of the elements of the infinitesimal generator. The existence of $\mathbf{A}_{0,0}^{-1}$ is ensured by the fact that all states $i \in S^0$ are transient in X.

For $S^0 = \emptyset$, the infinitesimal generator and initial probability distribution vector of Y would be $\mathbf{B} = \mathbf{R}^{-1}\mathbf{A}$ and $\boldsymbol{\beta} = \boldsymbol{\alpha}$.

Obtaining $\mathbf{C}$ and $\boldsymbol{\beta}$ in the case $S^0 \neq \emptyset$ is similar to the elimination of states with zero reward rate in the model transformation step developed in [5], and the alternatives discussed there could be used. However, here, to avoid subtractions we will obtain $\mathbf{C}$ and $\boldsymbol{\beta}$ by performing the Gaussian elimination of the block $\mathbf{A}_{0,0}$ in the matrix

$$\mathbf{D} = \begin{pmatrix} \mathbf{A}_{0,0} & \mathbf{A}_{0,\pm} & \mathbf{A}_{0,a} \\ \mathbf{A}_{\pm,0} & \mathbf{A}_{\pm,\pm} & \mathbf{A}_{\pm,a} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & 0 \\ \boldsymbol{\alpha}_0{}^{\mathrm{T}} & \boldsymbol{\alpha}_{\pm}{}^{\mathrm{T}} & \alpha_a \end{pmatrix},$$

yielding a matrix

$$\begin{pmatrix} \mathbf{U}_{0,0} & \times & \times \\ \mathbf{O} & \mathbf{A}_{\pm,\pm} - \mathbf{A}_{\pm,0}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,\pm} & \mathbf{A}_{\pm,a} - \mathbf{A}_{\pm,0}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,a} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & 0 \\ \mathbf{0}^{\mathrm{T}} & \boldsymbol{\alpha}_{\pm}{}^{\mathrm{T}} - \boldsymbol{\alpha}_0{}^{\mathrm{T}}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,\pm} & \alpha_a - \boldsymbol{\alpha}_0{}^{\mathrm{T}}\mathbf{A}_{0,0}^{-1}\mathbf{A}_{0,a} \end{pmatrix},$$

where $\mathbf{U}_{0,0}$ is upper triangular. Calling $\mathbf{D}^{(m)}$, $m = 1, \ldots, |S^0|$, the matrix obtained by performing Gaussian elimination up to the mth diagonal element of $\mathbf{A}_{0,0}$ in $\mathbf{D}$, it can be checked that, in the procedure, no subtractions occur except in the computation of the (negative) diagonal elements of $\mathbf{D}^{(m)}$. Those subtractions can be easily avoided by computing those diagonal elements as the sum, with the sign reversed, of the corresponding off-diagonal elements.

2.2 Closed-Form Expression for $F(s)$

In this section, we will obtain a closed-form expression for $F(s)$ based on a randomization construction for Y with two randomization rates. Let us start by reviewing the (standard) randomization construction [19] applied on Y. Consider any $\Lambda \geq \max_{i \in S^{\pm} \cup \{a\}} -b_{i,i}$ and let the (time homogeneous) discrete-time Markov chain (DTMC) $\widehat{Y} = \{\widehat{Y}_n; n = 0, 1, 2, \ldots\}$ with same state space and initial probability distribution as Y and transition matrix $\mathbf{I} + (1/\Lambda)\mathbf{B}$. Consider now the stochastic process $Z = \{Z(t); t \geq 0\}$ obtained by associating with the state visiting process $\widehat{Y}$ independent exponential visit durations with parameter Λ (when referring to Z, we will say that $\widehat{Y}$ has been randomized with rate Λ). Then (see, e.g., [20, Theorem 4.19]), Z is probabilistically identical to Y, and anything depending on the probabilistic path behavior of Y can be computed using Z instead. Theorem **??** in the Appendix, which can be found on the Computer Society Digital Library at `http://doi.ieeecomputersociety.org/10.1109/TDSC.2010.49`, extends that result to a randomization construction in which $\widehat{Y}$ has transition matrix $\mathbf{I} + \boldsymbol{\Lambda}^{-1}\mathbf{B}$, $\boldsymbol{\Lambda} = \mathbf{diag}[\Lambda_i]_{i \in S^{\pm} \cup \{a\}}$, $\Lambda_i \geq -b_{i,i}$, $\Lambda_i > 0$, and is randomized with rate Λ_i in each state $i \in S^{\pm} \cup \{a\}$. In this paper, for efficiency reasons, we will consider a randomization construction with two randomization rates: one, $\Lambda^+ \geq \max_{i \in S^+} -b_{i,i}$, for the states in $S^+ \cup \{a\}$, and another, $\Lambda^- \geq \max_{i \in S^-} -b_{i,i}$, for the states in S^-. Then, $\widehat{Y}$ will have transition matrix

$$\mathbf{P} = \mathbf{I} + \boldsymbol{\Lambda}_{\pm}^{-1}\mathbf{B}\,, \tag{1}$$

where $\boldsymbol{\Lambda}_{\pm} = \mathbf{diag}[\mathbf{1}_{i \in S^+ \cup \{a\}}\Lambda^+ + \mathbf{1}_{i \in S^-}\Lambda^-]_{i \in S^{\pm} \cup \{a\}}$, and initial probability distribution vector $\boldsymbol{\beta}$. Invoking Theorem **??** in the Appendix, which can be found on the Computer Society Digital Library at `http://doi.ieeecomputersociety.org/10.1109/TDSC.2010.49`, we can state the following result.

Proposition 1. *Let $Z = \{Z(t); t \geq 0\}$ be the stochastic process obtained by randomizing the DTMC $\widehat{Y}$ with rate Λ^+ in the states $i \in S^+ \cup \{a\}$ and rate Λ^- in the states $i \in S^-$. Then, Z is probabilistically identical to Y.*

We can now obtain the sought formalization for $F(s) = \Pr[T^+ - T^- \leq s]$. By considering the events $Y(0) \neq a \wedge T^+ - T^- \leq s$ and $Y(0) = a \wedge T^+ - T^- \leq s$ and noting that $Y(0) = a$ implies $T^+ = T^- = 0$, we clearly have

$$\begin{aligned} F(s) &= \Pr\left[T^+ - T^- \leq s\right] \\ &= \Pr\left[Y(0) \neq a \wedge T^+ - T^- \leq s\right] + \Pr\left[Y(0) = a \wedge T^+ - T^- \leq s\right] \\ &= \Pr\left[Y(0) \neq a \wedge T^+ - T^- \leq s\right] + \mathbf{1}_{s \geq 0}\beta_a \,. \end{aligned}$$

Let Q_{k^+,k^-} denote the probability that, starting in $S^{\pm}$, the DTMC $\widehat{Y}$ will enter state a after exactly k^+ visits to states in S^+ and k^- visits to states in S^-, $k^+ \geq 0$, $k^- \geq 0$, $k^+ + k^- > 0$. Let T^k_Λ denote a k-Erlang random variable with parameter Λ and let $P_{k^+,k^-}(s) = \Pr[T^{k^+}_{\Lambda^+} - T^{k^-}_{\Lambda^-} \leq s]$. By construction, Z spends an independent exponentially distributed time with parameter Λ^+ in each visit of $\widehat{Y}$ to S^+ and an independent exponentially distributed time with parameter Λ^- in each visit of $\widehat{Y}$ to S^-. Then, using the probabilistic identity of Y and Z asserted by Proposition 1 and the theorem of total probability,

$$\Pr\left[Y(0) \neq a \wedge T^+ - T^- \leq s\right] = \sum_{\substack{k^+ \geq 0,\, k^- \geq 0 \\ k^+ + k^- > 0}} Q_{k^+,k^-} P_{k^+,k^-}(s) \,,$$

implying

$$F(s) = \sum_{\substack{k^+ \geq 0,\, k^- \geq 0 \\ k^+ + k^- > 0}} Q_{k^+,k^-} P_{k^+,k^-}(s) + \mathbf{1}_{s \geq 0}\beta_a \,. \tag{2}$$

The formulation for $F(s)$ given by (2) is the basis of the method. It remains to truncate the infinite sum and to derive computable expressions for the probabilities Q_{k^+,k^-} and $P_{k^+,k^-}(s)$. We will address these issues in the following sections.

2.3 Truncation of the Infinite Sum

We will perform three truncations to the infinite sum of (2). The first truncation will delete the pairs (k^+, k^-) with $k^+ > N^+ \geq 0$, the second truncation will delete the pairs (k^+, k^-) with $k^- > N^- \geq 0$, and the third truncation will delete the pairs (k^+, k^-) with $k^+ + k^- > N \geq 0$. The domain, $D(N^+, N^-, N)$, of pairs (k^+, k^-) over which the sum will have to be computed is

$$D(N^+, N^-, N) = \{(k^+, k^-) \,:\, 0 \leq k^+ \leq N^+ \wedge 0 \leq k^- \leq N^- \wedge 0 < k^+ + k^- \leq N\}$$

and is illustrated in Figure 1 for the case $N^+ > 1$, $N^- > 1$ and $\max\{N^+, N^-\} < N < N^+ + N^-$ (we will argue at the end of this section that $N^+ \leq N$ and $N^- \leq N$). Let $\widehat{F}(s)$ be the estimate for $F(s)$ obtained by restricting the sum to the domain $D(N^+, N^-, N)$, i.e.

$$\widehat{F}(s) = \sum_{(k^+,k^-) \in D(N^+,N^-,N)} Q_{k^+,k^-} P_{k^+,k^-}(s) + \mathbf{1}_{s \geq 0}\beta_a \,. \tag{3}$$

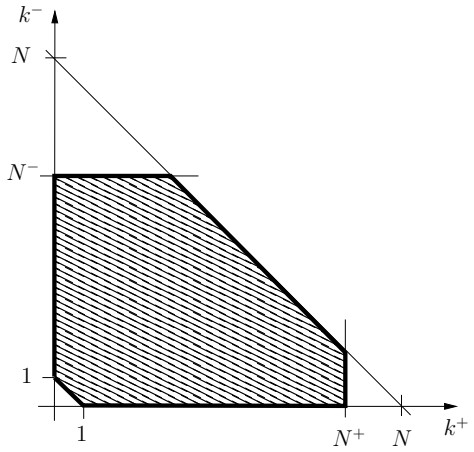

Figure 1: Illustration of domain $D(N^+, N^-, N)$ (the domain includes the points in the frontier).

Then, letting $D^{\mathrm{c}}(N^+, N^-, N) = \{(k^+, k^-) \,:\, k^+ \geq 0 \wedge k^- \geq 0 \wedge k^+ + k^- > 0 \wedge (k^+, k^-) \notin D(N^+, N^-, N)\}$, we have

$$\begin{aligned} F(s) - \widehat{F}(s) &= \sum_{(k^+,k^-)\in D^{\mathrm{c}}(N^+,N^-,N)} Q_{k^+,k^-} P_{k^+,k^-}(s) \\ &\leq \sum_{(k^+,k^-)\in D^{\mathrm{c}}(N^+,N^-,N)} Q_{k^+,k^-} \\ &\leq \sum_{\substack{k^+>N^+ \\ k^-\geq 0}} Q_{k^+,k^-} + \sum_{\substack{k^->N^- \\ k^+\geq 0}} Q_{k^+,k^-} + \sum_{\substack{k^++k^->N \\ k^+\geq 0,\, k^-\geq 0}} Q_{k^+,k^-} \\ &= \delta^+(N^+) + \delta^-(N^-) + \delta(N)\,. \end{aligned}$$

Let $\varepsilon > 0$ be an error control parameter. Then, to compute $F(s)$ using $\widehat{F}(s)$ with truncation error bounded from above by ε, it is enough to select N^+, N^- and N as

$$\begin{aligned} N^+ &= \min\left\{n \geq 0 \,:\, \delta^+(n) \leq \frac{\varepsilon}{3}\right\}, \\ N^- &= \min\left\{n \geq 0 \,:\, \delta^-(n) \leq \frac{\varepsilon}{3}\right\}, \\ N &= \min\left\{n \geq 0 \,:\, \delta(n) \leq \frac{\varepsilon}{3}\right\}. \end{aligned}$$

Obtaining the truncation parameters N^+, N^-, N requires computing $\delta^+(n)$, $\delta^-(n)$, and $\delta(n)$ for increasing n starting at $n = 0$. We start by discussing the computation of $\delta(n) = \sum_{k^++k^->n,\, k^+\geq 0,\, k^-\geq 0} Q_{k^+,k^-}$. Let $\boldsymbol{\pi}(n)$ denote the probability distribution vector of $\widehat{Y}$ at step n. From the definition of Q_{k^+,k^-} (see Section 2.2), it follows that $\delta(n)$ is the probability that, starting in $S^{\pm}$, the DTMC $\widehat{Y}$ will enter the absorbing state a after more than n visits to $S^{\pm}$, or, equivalently, that $\widehat{Y}$ will be in $S^{\pm}$ at step n. Then, $\delta(n) = \boldsymbol{\pi}(n)_{\pm}{}^{\mathrm{T}}\mathbf{1}$, where vectors $\boldsymbol{\pi}(n)_{\pm}$ can be computed for increasing n starting at $n = 0$ using $\boldsymbol{\pi}(0)_{\pm} = \boldsymbol{\beta}_{\pm}$ and $\boldsymbol{\pi}(n)_{\pm}{}^{\mathrm{T}} = \boldsymbol{\pi}(n-1)_{\pm}{}^{\mathrm{T}}\mathbf{P}_{\pm,\pm}$, $n > 0$.

The computation of $\delta^+(n) = \sum_{k^+>n,\, k^-\geq 0} Q_{k^+,k^-}$ for increasing n starting at $n = 0$ can be performed by analyzing a DTMC $\widehat{Y}^+ = \{\widehat{Y}^+_n; n = 0, 1, 2, \ldots\}$ with state space $S^+ \cup \{a\}$ defined

from $\widehat{Y}$ by "jumping over" the states in S^-. Formally, $\widehat{Y}^+_n = \widehat{Y}_{M_n}$, where $M_0 = \min\{n \geq 0 : \widehat{Y}_n \in S^+ \cup \{a\}\}$ and $M_n = \min\{n > M_{n-1} : \widehat{Y}_n \in S^+ \cup \{a\}\}$, $n > 0$. The transition matrix $\mathbf{P}^+$ and initial probability distribution vector $\boldsymbol{\beta}^+$ of $\widehat{Y}^+$ are

$$\mathbf{P}^+ = \begin{pmatrix} \mathbf{P}^+_{+,+} & \mathbf{P}^+_{+,a} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{pmatrix}, \quad \boldsymbol{\beta}^+ = \begin{pmatrix} \boldsymbol{\beta}^+_+ \\ \beta^+_a \end{pmatrix},$$

with

$$\begin{aligned}
\mathbf{P}^+_{+,+} &= \mathbf{P}_{+,+} + \mathbf{P}_{+,-}(\mathbf{I} - \mathbf{P}_{-,-})^{-1}\mathbf{P}_{-,+}\,, \\
\mathbf{P}^+_{+,a} &= \mathbf{P}_{+,a} + \mathbf{P}_{+,-}(\mathbf{I} - \mathbf{P}_{-,-})^{-1}\mathbf{P}_{-,a}\,, \\
\boldsymbol{\beta}^{+\mathrm{T}}_+ &= \boldsymbol{\beta}_+{}^{\mathrm{T}} + \boldsymbol{\beta}_-{}^{\mathrm{T}}(\mathbf{I} - \mathbf{P}_{-,-})^{-1}\mathbf{P}_{-,+}\,, \\
\beta^+_a &= \beta_a + \boldsymbol{\beta}_-{}^{\mathrm{T}}(\mathbf{I} - \mathbf{P}_{-,-})^{-1}\mathbf{P}_{-,a}\,.
\end{aligned}$$

We note that the existence of $(\mathbf{I} - \mathbf{P}_{-,-})^{-1}$ is guaranteed by the fact that all states $i \in S^-$ are transient in $\widehat{Y}$. Let $\boldsymbol{\pi}^+(n)$ denote the probability distribution vector of $\widehat{Y}^+$ at step n. From the definition of Q_{k^+,k^-}, it follows that $\delta^+(n)$ is the probability that, starting in $S^{\pm}$, the DTMC $\widehat{Y}$ will enter the absorbing state a after more than n visits to S^+, or, equivalently, that $\widehat{Y}^+$ will be in S^+ at step n. Then, we have $\delta^+(n) = \boldsymbol{\pi}^+(n)_+{}^{\mathrm{T}}\mathbf{1}$, where vectors $\boldsymbol{\pi}^+(n)_+$ can be computed for increasing n starting at $n = 0$ using $\boldsymbol{\pi}^+(0)_+ = \boldsymbol{\beta}^+_+$ and $\boldsymbol{\pi}^+(n)_+{}^{\mathrm{T}} = \boldsymbol{\pi}^+(n-1)_+{}^{\mathrm{T}}\mathbf{P}^+_{+,+}$, $n > 0$.

The computation of $\delta^-(n) = \sum_{k^- > n,\, k^+ \geq 0} Q_{k^+,k^-}$ for increasing n starting at $n = 0$ can be performed by analyzing a DTMC $\widehat{Y}^- = \{\widehat{Y}^-_n; n = 0, 1, 2, \ldots\}$ with state space $S^- \cup \{a\}$ defined from $\widehat{Y}$ by "jumping over" the states in S^+. The transition matrix $\mathbf{P}^-$ and initial probability distribution vector $\boldsymbol{\beta}^-$ of $\widehat{Y}^-$ are

$$\mathbf{P}^- = \begin{pmatrix} \mathbf{P}^-_{-,-} & \mathbf{P}^-_{-,a} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{pmatrix}, \quad \boldsymbol{\beta}^- = \begin{pmatrix} \boldsymbol{\beta}^-_- \\ \beta^-_a \end{pmatrix},$$

with

$$\begin{aligned}
\mathbf{P}^-_{-,-} &= \mathbf{P}_{-,-} + \mathbf{P}_{-,+}(\mathbf{I} - \mathbf{P}_{+,+})^{-1}\mathbf{P}_{+,-}\,, \\
\mathbf{P}^-_{-,a} &= \mathbf{P}_{-,a} + \mathbf{P}_{-,+}(\mathbf{I} - \mathbf{P}_{+,+})^{-1}\mathbf{P}_{+,a}\,, \\
\boldsymbol{\beta}^{-\mathrm{T}}_- &= \boldsymbol{\beta}_-{}^{\mathrm{T}} + \boldsymbol{\beta}_+{}^{\mathrm{T}}(\mathbf{I} - \mathbf{P}_{+,+})^{-1}\mathbf{P}_{+,-}\,, \\
\beta^-_a &= \beta_a + \beta_+{}^{\mathrm{T}}(\mathbf{I} - \mathbf{P}_{+,+})^{-1}\mathbf{P}_{+,a}\,.
\end{aligned}$$

We note that the existence of $(\mathbf{I} - \mathbf{P}_{+,+})^{-1}$ is guaranteed by the fact that all states $i \in S^+$ are transient in $\widehat{Y}$. Let $\boldsymbol{\pi}^-(n)$ denote the probability distribution vector of $\widehat{Y}^-$ at step n. From the definition of Q_{k^+,k^-}, it follows that $\delta^-(n)$ is the probability that, starting in $S^{\pm}$, the DTMC $\widehat{Y}$ will enter the absorbing state a after more than n visits to S^-, or, equivalently, that $\widehat{Y}^-$ will be in S^- at step n. Then, $\delta^-(n) = \boldsymbol{\pi}^-(n)_-{}^{\mathrm{T}}\mathbf{1}$, where vectors $\boldsymbol{\pi}^-(n)_-$ can be computed for increasing n starting at $n = 0$ using $\boldsymbol{\pi}^-(0)_- = \boldsymbol{\beta}^-_-$ and $\boldsymbol{\pi}^-(n)_-{}^{\mathrm{T}} = \boldsymbol{\pi}^-(n-1)_-{}^{\mathrm{T}}\mathbf{P}^-_{-,-}$, $n > 0$.

The matrix $\mathbf{P}^+$ and the vector $\boldsymbol{\beta}^+$ can be obtained by performing the Gaussian elimination of

the block $-(\mathbf{I}-\mathbf{P}_{-,-})$ in the matrix

$$\mathbf{Q}=\begin{pmatrix} -(\mathbf{I}-\mathbf{P}_{-,-}) & \mathbf{P}_{-,+} & \mathbf{P}_{-,a} \\ \mathbf{P}_{+,-} & \mathbf{P}_{+,+} & \mathbf{P}_{+,a} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & 1 \\ \boldsymbol{\beta}_{-}{}^{\mathrm{T}} & \boldsymbol{\beta}_{+}{}^{\mathrm{T}} & \beta_a \end{pmatrix}.$$

Calling $\mathbf{Q}^{(m)}$, $m=1,\ldots,|S^{-}|$, the matrix that is obtained by performing Gaussian elimination up to the mth diagonal element of $-(\mathbf{I}-\mathbf{P}_{-,-})$ in $\mathbf{Q}$, it can be checked that no subtractions occur in the procedure provided that the (negative) diagonal elements of $\mathbf{Q}^{(m)}_{-,-}$ are computed as the sum, with the sign reversed, of the corresponding off-diagonal elements. The matrix $\mathbf{P}^{-}$ and the vector $\boldsymbol{\beta}^{-}$ can be obtained without subtractions similarly.

To conclude this section we note that, given the previously commented interpretations of $\delta(n)$, $\delta^{+}(n)$, and $\delta^{-}(n)$ in terms of the probabilistic behavior of $\widehat{Y}$, we clearly have $\delta^{+}(n)\leq\delta(n)$ and $\delta^{-}(n)\leq\delta(n)$, implying $N^{+}\leq N$ and $N^{-}\leq N$.

2.4 Computable Expressions for the Probabilities Q_{k^+,k^-} and $P_{k^+,k^-}(s)$

We start by deriving computable expressions for Q_{k^+,k^-}, $(k^+,k^-)\in D(N,N^+,N^-)$, which, we recall, is the probability that, starting in $S^{\pm}$, the DTMC $\widehat{Y}$ will enter state a after exactly k^+ visits to states in S^{+} and k^- visits to states in S^{-}. Let $\pi_i(k^+,k^-)$, $i\in S^{\pm}\cup\{a\}$, denote the probability that in its first k^++k^- visits, $\widehat{Y}$ has made k^+ visits to S^{+} and k^- visits to S^{-} and the (k^++k^-)th visited state is state i. Let the vector $\boldsymbol{\pi}(k^+,k^-)=(\pi_i(k^+,k^-))_{i\in S^{\pm}\cup\{a\}}$. Then, clearly,

$$Q_{k^+,k^-}=\boldsymbol{\pi}(k^+,k^-)_{\pm}{}^{\mathrm{T}}\mathbf{P}_{\pm,a}\,, \tag{4}$$

reducing the problem of computing Q_{k^+,k^-}, $(k^+,k^-)\in D(N,N^+,N^-)$, to that of computing the vectors $\boldsymbol{\pi}(k^+,k^-)_{\pm}$, $(k^+,k^-)\in D(N,N^+,N^-)$. These vectors are related by the following set of recurrences, which follow from applying backward renewal equations on $\widehat{Y}$:

$$\boldsymbol{\pi}(0,k^-)_{+}=\mathbf{0}\,,\quad k^->0\,, \tag{5}$$

$$\boldsymbol{\pi}(1,0)_{+}=\boldsymbol{\beta}_{+}\,, \tag{6}$$

$$\boldsymbol{\pi}(k^+,0)_{+}{}^{\mathrm{T}}=\boldsymbol{\pi}(k^+-1,0)_{+}{}^{\mathrm{T}}\mathbf{P}_{+,+}\,,\quad k^+>1\,, \tag{7}$$

$$\boldsymbol{\pi}(k^+,k^-)_{+}{}^{\mathrm{T}}=\boldsymbol{\pi}(k^+-1,k^-)_{\pm}{}^{\mathrm{T}}\mathbf{P}_{\pm,+}\,,\quad k^+>0,\ k^->0\,, \tag{8}$$

$$\boldsymbol{\pi}(k^+,0)_{-}=\mathbf{0}\,,\quad k^+>0\,, \tag{9}$$

$$\boldsymbol{\pi}(0,1)_{-}=\boldsymbol{\beta}_{-}\,, \tag{10}$$

$$\boldsymbol{\pi}(0,k^-)_{-}{}^{\mathrm{T}}=\boldsymbol{\pi}(0,k^--1)_{-}{}^{\mathrm{T}}\mathbf{P}_{-,-}\,,\quad k^->1\,, \tag{11}$$

$$\boldsymbol{\pi}(k^+,k^-)_{-}{}^{\mathrm{T}}=\boldsymbol{\pi}(k^+,k^--1)_{\pm}{}^{\mathrm{T}}\mathbf{P}_{\pm,-}\,,\quad k^+>0,\ k^->0\,. \tag{12}$$

We deal next with the probabilities $P_{k^+,k^-}(s)$, $(k^+,k^-)\in D(N,N^+,N^-)$. These probabilities, for which there does not seem to exist efficient closed-form expressions, are related by the recurrences given by the following theorem.

Theorem 1. *Assume $s > 0$. Then, for $k^+ > 0$,*

$$P_{k^+,0}(s) = \sum_{n=k^+}^{\infty} \frac{(\Lambda^+ s)^n}{n!} \, \mathrm{e}^{-\Lambda^+ s} \,,$$

for $k^- > 0$,

$$P_{0,k^-}(s) = 1 \,,$$

and, for $k^+ > 0$ and $k^- > 0$,

$$P_{k^+,k^-}(s) = \frac{\Lambda^+}{\Lambda^+ + \Lambda^-} \, P_{k^+-1,k^-}(s) + \frac{\Lambda^-}{\Lambda^+ + \Lambda^-} \, P_{k^+,k^--1}(s) \,.$$

Assume $s \leq 0$. Then, for $k^- > 0$, under the convention $0^0 = 1$,

$$P_{0,k^-}(s) = \sum_{n=0}^{k^--1} \frac{(\Lambda^- |s|)^n}{n!} \, \mathrm{e}^{-\Lambda^- |s|} \,,$$

for $k^+ > 0$,

$$P_{k^+,0}(s) = 0 \,,$$

and, for $k^+ > 0$ and $k^- > 0$,

$$P_{k^+,k^-}(s) = \frac{\Lambda^+}{\Lambda^+ + \Lambda^-} \, P_{k^+-1,k^-}(s) + \frac{\Lambda^-}{\Lambda^+ + \Lambda^-} \, P_{k^+,k^--1}(s) \,.$$

Proof. See the Appendix, which can be found on the Computer Society Digital Library at `http://doi.ieeecomputersociety.org/10.1109/TDSC.2010.49`. □

In the following section, we will summarize the computational steps of the method and will describe how (4)–(12) and the recurrences given by Theorem 1 are used in the numerical evaluation $\widehat{F}(s)$ in (3).

2.5 Computational Steps

According to our previous developments, the proposed method to compute $F(s_i)$, $i = 1, 2, \ldots$, with truncation error bounded from above by ε can be summarized in the following computational steps:

1. Obtain the matrix $\mathbf{B}$ and the vector $\boldsymbol{\beta}$ from the matrix $\mathbf{A}$ and the vector $\boldsymbol{\alpha}$ as described in Section 2.1.

2. Choose the randomization rates $\Lambda^+ \geq \max_{i \in S^+} -b_{i,i}$ and $\Lambda^- \geq \max_{i \in S^-} -b_{i,i}$ and compute the matrix $\mathbf{P}$ using (1). (For reasons of numerical stability, strict inequalities should hold —see Section 2.6.)

3. Compute the truncation parameters N^+, N^-, and N as described in Section 2.3.

4. Compute the probabilities Q_{k^+,k^-}, $P_{k^+,k^-}(s_i)$, $(k^+,k^-) \in D(N^+,N^-,N)$, $i = 1,2,\ldots$.

5. Compute $F(s_i)$ as $\widehat{F}(s_i) = \sum_{(k^+,k^-)\in D(N^+,N^-,N)} Q_{k^+,k^-} P_{k^+,k^-}(s_i) + \mathbf{1}_{s_i \geq 0}\beta_a$, $i = 1,2,\ldots$.

The probabilities Q_{k^+,k^-}, $(k^+,k^-) \in D(N^+,N^-,N)$, are computed once using (4)–(12) and stored. The computation is performed traversing the domain $D(N^+,N^-,N)$ in a diagonal by diagonal manner, starting at the diagonal $k^+ + k^- = 1$ and with increasing k^+ within each diagonal. The probabilities $P_{k^+,k^-}(s_i)$, $(k^+,k^-) \in D(N^+,N^-,N)$, $i = 1,2,\ldots$, are computed along with $\widehat{F}(s_i)$ as follows. Let $P_k(\lambda) = (\lambda^k/k!)\mathrm{e}^{-\lambda}$, $\lambda \geq 0$, $k \geq 0$, denote Poisson probabilities. For each $s_i > 0$, we obtain, if $N^+ \geq 1$, $P_{k^+,0}(s_i) = TR_{k^+}(\Lambda^+ s_i) = \sum_{n=k^+}^{\infty} P_n(\Lambda^+ s_i)$, $1 \leq k^+ \leq N^+$, using the procedure described in the Appendix, which can be found on the Computer Society Digital Library at `http://doi.ieeecomputersociety.org/10.1109/TDSC.2010`, and store them. Then, we obtain $P_{k^+,k^-}(s_i)$, $(k^+,k^-) \in D(N^+,N^-,N)$, using Theorem 1, traversing the domain $D(N^+,N^-,N)$ by rows (increasing k^-, starting at $k^- = 0$ and, for each k^-, increasing k^+), and accumulate the sum of $\widehat{F}(s_i)$ using the so-called Kahan's summation algorithm (see, e.g., [21]), which yields excellent relative accuracy to compute the sum of a set of floating-point values with the same sign [21]. Finally, for each $s_i \leq 0$, we obtain, if $N^- \geq 1$, $P_{0,k^-}(s_i) = TL_{k^-}(\Lambda^-|s_i|) = \sum_{n=0}^{k^--1} P_n(\Lambda^-|s_i|)$, $1 \leq k^- \leq N^-$, using the procedure described in the Appendix, which can be found on the Computer Society Digital Library at `http://doi.ieeecomputersociety.org/10.1109/TDSC.2010`, and store them. Then, we obtain $P_{k^+,k^-}(s_i)$, $(k^+,k^-) \in D(N^+,N^-,N)$, using Theorem 1, traversing the domain $D(N^+,N^-,N)$ by columns (increasing k^+, starting at $k^+ = 0$ and, for each k^+, increasing k^-), and accumulate the sum of $\widehat{F}(s_i)$ using Kahan's summation algorithm.

2.6 Numerical Stability and Computational Cost

The procedures for the computation of $P_{k^+,0}(s) = TR_{k^+}(\Lambda^+ s)$, $s > 0$, and $P_{0,k^-}(s) = TL_{k^-}(\Lambda^-|s|)$, $s \leq 0$, described in the Appendix, which can be found on the Computer Society Digital Library at `http://doi.ieeecomputersociety.org/10.1109/TDSC.2010`, are, ignoring underflows in Poisson probabilities, numerically stable and, ignoring those underflows, compute $P_{k^+,0}(s)$ and $P_{0,k^-}(s)$ with very small relative error. Those procedures depend on a parameter η, $0 < \eta < 1$, to be defined below. No subtractions occur in the rest of the method except in the computation of the diagonal elements of the matrix $\mathbf{P}$ using (1), namely $p_{i,i} = 1 - |b_{i,i}|/\Lambda^+$, $i \in S^+$, and $p_{i,i} = 1 - |b_{i,i}|/\Lambda^-$, $i \in S^-$. In order to avoid those subtractions to have an important impact on the relative error with which the probabilities Q_{k^+,k^-} are computed, we take $\Lambda^+ = (1+\psi)\max_{i\in S^+} -b_{i,i}$ and $\Lambda^- = (1+\psi)\max_{i\in S^-} -b_{i,i}$, with $\psi = 10^{-3}$. As a conclusion, the method is numerically stable and, ignoring underflows and overflows, round-off errors should have a small impact on the relative error with which $F(s)$ is computed. The truncation error is strictly bounded from above by the user-specified error control parameter ε. Overflows and underflows are very unlikely to happen except for underflows in Poisson probabilities (consider, for instance, the expression for $P_{k^+,0}(s)$, $k^+ > 0$, for the case $s > 0$ and the expression for $P_{0,k^-}(s)$,

$k^- > 0$, for the case $s < 0$ when $k^- \ll \Lambda^-|s|$), underflows in the computation of the probabilities $P_{k^+,k^-}(s)$, $k^+ > 0$, $k^- > 0$, using the recurrences of Theorem 1, and underflows in the products $Q_{k^+,k^-}P_{k^+,k^-}(s)$ involved in the computation of $\widehat{F}(s)$ using (3). The problem with underflows is that they may introduce a large relative error at the point they occur, raising the issue of how large the resulting error in $\widehat{F}(s)$ can be. However, it can be justified that the absolute error introduced in $\widehat{F}(s)$ by those underflows will be very small in realistic scenarios if enough precision is used. The justification is based on a rough upper bound for the absolute error. To derive the bound, we will assume that the underlying arithmetic is exact and will analyze the error introduced in $\widehat{F}(s)$ by discarding Poisson probabilities, probabilities $P_{k^+,k^-}(s)$, $k^+ > 0$, $k^- > 0$, and products $Q_{k^+,k^-}P_{k^+,k^-}(s)$ whose value is $< \eta$, where η, $0 < \eta < 1$, is such that underflows in the actual arithmetic can only happen when the computed value is $< \eta$ in absolute value.

Theorem 2. *In exact arithmetic, the absolute error introduced in $\widehat{F}(s)$ by discarding the Poisson probabilities whose value is $< \eta$, the probabilities $P_{k^+,k^-}(s)$, $(k^+,k^-) \in D(N^+,N^-,N)$, $k^+ > 0$, $k^- > 0$ whose value is $< \eta$, and the products $Q_{k^+,k^-}P_{k^+,k^-}(s)$, $(k^+,k^-) \in D(N^+,N^-,N)$ whose value is $< \eta$ is bounded from above by*

$$\Big(\max\Big\{\mathbf{1}_{s>0}2(1+\Lambda^+s)N^+, \mathbf{1}_{s<0}\,\frac{N^-(N^-+1)}{2}\Big\} + 2|D(N^+,N^-,N)| + 1\Big)|D(N^+,N^-,N)|\eta\,.$$

Proof. See the Appendix, which can be found on the Computer Society Dogital Library at `http://doi.ieeecomputersociety.org/10.1109/TDSC.2010.49`. □

The upper bound given by Theorem 2 is small enough to guarantee for practical purposes that the absolute error in $\widehat{F}(s)$ caused by underflows will be negligible if enough precision is used. Consider, for instance, that all computations are carried out using the 64-bit binary format defined in the IEEE 754 floating-point standard [22]. In that case, we can take $\eta = 2^{-1022} = 2.225 \times 10^{-308}$. Let $M = \max\{N^+, N^-, \mathbf{1}_{s>0}\Lambda^+ s\}$. Then, using $|D(N^+,N^-,N)| \leq (N^+ + 1)(N^- + 1) \leq (M+1)^2$, the upper bound given by the theorem is bounded by $(2(M+1)^3(2M+1)+(M+1)^2)\eta = (2(M+1)^3(2M+1)+(M+1)^2) \times 2.225 \times 10^{-308}$, and for this upper bound to be, say, $\geq 10^{-40}$, M would have to be $\geq 5.790 \times 10^{66}$ and either $|D(N^+,N^-,N)|$ would have a size such that the method would be unfeasible due to both CPU time and memory consumption or, if $s > 0$, $\Lambda^+ s$ would have an unrealistically large value.

The truncation parameters N^+, N^- and N depend on the characteristics of $\widehat{Y}$ and it is difficult, in general, to anticipate their values. It is also difficult to anticipate the computational cost of the Gaussian eliminations, since it depends on the amount of fill that occurs during them. That fill can be reduced heuristically by using an appropriate ordering of the states associated with the rows of the blocks that have to be eliminated. This is equivalent to performing symmetrical permutations of rows and columns intersecting the block that has to be eliminated and several criteria are available for choosing those permutations [23]. In our implementation we did not follow any criterion and used the ordering of the states induced by the order in which the states were generated. For MRMs

with a number of states neither too small nor too large, the cost of the computation of the probabilities Q_{k^+,k^-}, $(k^+,k^-) \in D(N^+,N^-,N)$, will often dominate the computational cost of the method. Let K denote the number of non-null elements of $\mathbf{P}$. From (4)–(12), it follows that computing one probability Q_{k^+,k^-} has a cost $O(K)$ time. In addition, given the recurrences (5)-(12), the computation of the probabilities Q_{k^+,k^-} traversing the domain $D(N^+,N^-,N)$ by diagonals and with increasing k^+ within each diagonal (see Section 2.5) requires a set of $3+\min\{N^+,N^-\}$ vectors of size $|S^{\pm}|$ each to store the vectors $\boldsymbol{\pi}(k^+,k^-)_{\pm}$ (the maximum number of (k^+,k^-) pairs in any diagonal is $1+\min\{N^+,N^-\}$). Therefore, the cost of the computation of the probabilities Q_{k^+,k^-}, $(k^+,k^-) \in D(N^+,N^-,N)$, is $O(K|D(N^+,N^-,N)|) = O(KN^+N^-)$ time and, taking into account the storage of $\mathbf{P}$ and the Q_{k^+,k^-} themselves, $O(K+|D(N^+,N^-,N)|+(3+\min\{N^+,N^-\})|S^{\pm}|) = O(K+N^+N^-+(3+\min\{N^+,N^-\})|S^{\pm}|)$ memory. For MRMs having a very small number of states, the computational cost of the method would be dominated by the computation of the probabilities $P_{k^+,k^-}(s)$, specially if $F(s)$ has to be computed for several values of s. For MRMs with a very large number of states, the computational cost of the Gaussian eliminations could be relatively important due to an important fill during these eliminations.

3 Analysis

In this section, we will illustrate, using a scalable example, the application of the method. We will also analyze the computational cost of the method and compare it with that of the alternative method based on the results given in [6].

The scalable example is a hardware/software fault-tolerant system controlling a manufacturing process. The system has n_{s} control sites numbered 1, 2, ..., n_{s}. Each site includes two redundant computers and a fault-tolerant software using the recovery block technique (each computer holds its own copy of the primary and secondary versions of the software). In that technique [24], an acceptance test checks the outputs of the primary version of the software and the secondary version is activated if the check is unsuccessful. In that case, the outputs of the secondary version are also checked by the acceptance test. We assume that, initially, each site is running the primary software version in one of the computers while the second computer is in standby (but not running any version of the software). For control site i, $1 \leq i \leq n_{\mathrm{s}}$, the failure rate of the primary software version is $\lambda_{\mathrm{PS}i}$. Upon occurrence of a failure in the primary software version, with probability $C_{\mathrm{S}i}$, the control task can be resumed using the secondary software version held in the same computer and the primary software version is restarted at rate $\mu_{\mathrm{PSR}i}$. When the restart finishes, the primary software version becomes the one active again. With probability $1-C_{\mathrm{S}i}$, the control task cannot be resumed using the secondary software version and a global software restart of both versions of the software has to be carried out on the computer that was running them. That restart has rate $\mu_{\mathrm{GSR}i}$. After that software restart operation the primary software version becomes active again. The secondary software version of site i, $1 \leq i \leq n_{\mathrm{s}}$, has failure rate $\lambda_{\mathrm{SS}i}$, and any failure in that secondary software version requires executing the global software restart with rate $\mu_{\mathrm{GSR}i}$ on the computer running the software, after which the primary software version becomes the one active. For control site i, $1 \leq i \leq n_{\mathrm{s}}$, the

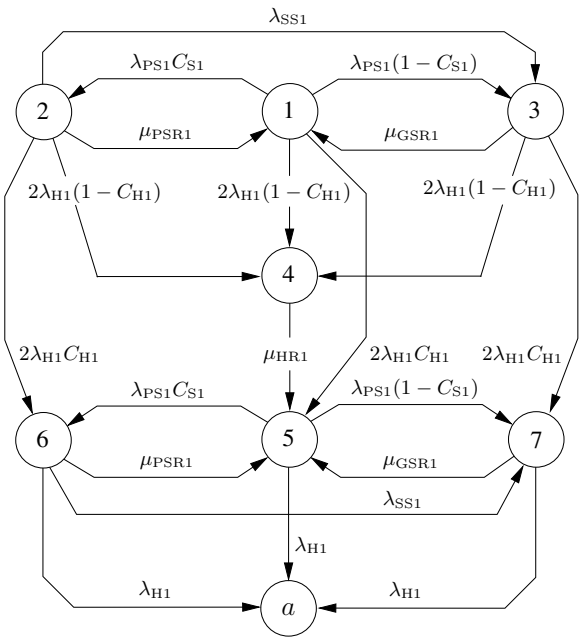

Figure 2: State transition diagram of the CTMC underlying the MRM of the fault-tolerant system with one control site and $n_{\mathrm{HR1}} = 1$.

failure rate of each computer is $\lambda_{\mathrm{H}i}$. When one computer is running the primary software version or running the secondary software version and restarting the primary software version or performing a global restart of both software versions, and the second computer is in standby, with (coverage) probability $C_{\mathrm{H}i}$, the site tolerates the failure of any computer without stopping the running software and the software restart operation, and with probability $1 - C_{\mathrm{H}i}$, the failure of a computer requires performing a hardware restart, after which the site is left with a single computer running the primary software version. The time taken by that restart follows an Erlang distribution with $n_{\mathrm{HR}i}$ stages and mean value $\mu_{\mathrm{HR}i}^{-1}$. When only one computer is left, the failure of the computer makes the control site and the system fail. We assume that there is an unlimited number of repairmen to perform software and hardware restarts. Fig. 2 shows the state transition diagram of the CTMC underlying the MRM of the system with one control site and $n_{\mathrm{HR1}} = 1$. Entry into the absorbing state a captures the failure of the system due to the failure of both computers of the site. For the system with more than one control site, entry into the absorbing state a would capture the failure of the system due to the failure of both computers in any of the control sites.

We will consider instances of the scalable example with two, three, four, and five control sites. For $n_{\mathrm{s}} = 2$ we will use the following values: $\lambda_{\mathrm{PS1}} = 10^{-3}\,\mathrm{h}^{-1}$, $\lambda_{\mathrm{PS2}} = 1.5 \times 10^{-3}\,\mathrm{h}^{-1}$, $\lambda_{\mathrm{SS1}} = 10^{-4}\,\mathrm{h}^{-1}$, $\lambda_{\mathrm{SS2}} = 1.5 \times 10^{-4}\,\mathrm{h}^{-1}$, $\lambda_{\mathrm{H1}} = 3 \times 10^{-5}\,\mathrm{h}^{-1}$, and $\lambda_{\mathrm{H2}} = 4 \times 10^{-5}\,\mathrm{h}^{-1}$. For $n_{\mathrm{s}} = 3$ we will use the same values and $\lambda_{\mathrm{PS3}} = 2 \times 10^{-3}\,\mathrm{h}^{-1}$, $\lambda_{\mathrm{SS3}} = 2 \times 10^{-4}\,\mathrm{h}^{-1}$, and $\lambda_{\mathrm{H3}} = 5 \times 10^{-5}\,\mathrm{h}^{-1}$. For $n_{\mathrm{s}} = 4$ we will use the same values as for $n_{\mathrm{s}} = 3$ and $\lambda_{\mathrm{PS4}} = 2.5 \times 10^{-3}\,\mathrm{h}^{-1}$, $\lambda_{\mathrm{SS4}} = 2.5 \times 10^{-4}\,\mathrm{h}^{-1}$, and $\lambda_{\mathrm{H4}} = 6 \times 10^{-5}\,\mathrm{h}^{-1}$. For $n_{\mathrm{s}} = 5$ we will use the same values as for $n_{\mathrm{s}} = 4$ and $\lambda_{\mathrm{PS5}} = 3 \times 10^{-3}\,\mathrm{h}^{-1}$, $\lambda_{\mathrm{SS5}} = 3 \times 10^{-4}\,\mathrm{h}^{-1}$, and $\lambda_{\mathrm{H5}} = 7 \times 10^{-5}\,\mathrm{h}^{-1}$. In all cases we will take $C_{\mathrm{S}i} = 0.9$, $\mu_{\mathrm{PSR}i} = 0.5\,\mathrm{h}^{-1}$, $\mu_{\mathrm{GSR}i} = 0.15\,\mathrm{h}^{-1}$, $C_{\mathrm{H}i} = 0.98$, and $\mu_{\mathrm{HR}i} = 0.12\,\mathrm{h}^{-1}$,

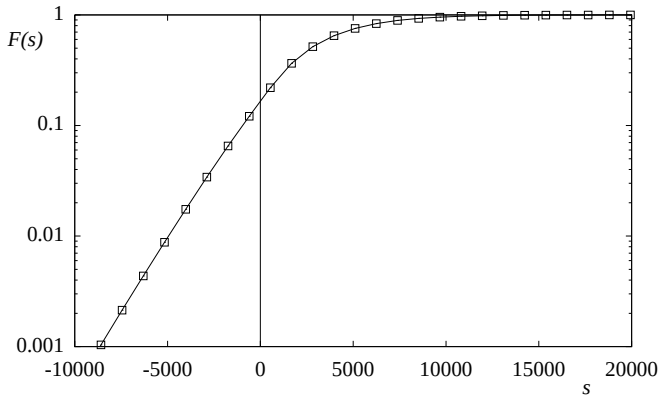

Figure 3: $F(s)$ for the MRM of the fault-tolerant system with five control sites and $n_{\mathrm{HR}i} = 1$, $1 \leq i \leq 5$.

$1 \leq i \leq n_{\mathrm{s}}$, leaving $n_{\mathrm{HR}i}$, $1 \leq i \leq n_{\mathrm{s}}$ as parameters. We associate a reward rate $1\ \mathrm{h}^{-1}$ with the states in which all control sites are working with the primary software version (see states 1 and 5 of Fig. 2), a reward rate 0 with the states in which all control sites are working, but some site is using the secondary software version (see states 2 and 6 of Fig. 2), and a reward rate $-100\ \mathrm{h}^{-1}$ with the states in which some control site is temporarily not working because a global software restart or a hardware restart is under way (see states 3, 4, and 7 of Fig. 2). Reward rates $1\ \mathrm{h}^{-1}$ are interpreted as rate of economic benefit resulting from the use of the system with all control sites using the primary software versions, reward rates 0 model the fact that no economic benefit (but also no economic loss) results from the use of the system with some control site using the secondary software version, and reward rates $-100\ \mathrm{h}^{-1}$ model the rate of economic loss resulting from some control site being temporarily not working. With that interpretation, $F(s)$ is the distribution function of the economic benefit resulting from the use of the system until it fails because of the failure of both computers in any of the control sites. That economic benefit could be balanced against the acquisition cost of the system.

All results were obtained on a workstation equipped with four Quad-Core Intel Xeon X7350 2.93 GHz processor chips with 64 kB of level 1 cache per core and 8 MB of level 2 cache per core pair, and 64 GB of main memory, using only one core and using the IEEE 754 64-bit binary format. For the parameter η on which the procedures described in the Appendix, which can be found on the Computer Society Digital Library at `http://doi.ieeecomputersociety.org/10.1109/TDSC.2010.39`, depend, we took $\eta = 2^{-1022} = 2.225 \times 10^{-308}$.

We start by illustrating the application of the method. Fig. 3 plots $F(s)$ as a function of s for the MRM of the fault-tolerant system with five control sites and $n_{\mathrm{HR}i} = 1$, $1 \leq i \leq 5$. An interesting observation is that the economical benefit resulting from the use of the system is negative with a non-negligible probability. That probability is equal to 0.1657.

Now, we turn to the analysis of the computational cost of the method. We start by analyzing that cost when the method is run with a single s and increasingly stringent error requirements for the

Table 1: Results for the new method for the MRM of the fault-tolerant system with five control sites and $n_{\mathrm{HR}i} = 1$, $1 \leq i \leq 5$.

s	ε	N^+	N^-	N	$\lvert D(N^+, N^-, N)\rvert$	CPU time (s)	% Q_{k^+,k^-}
$-10{,}000$	10^{-4}	75	937	1,004	71,251	1,322	96.4
$-10{,}000$	10^{-6}	100	1,276	1,366	128,921	2,486	97.4
$-10{,}000$	10^{-8}	126	1,608	1,721	204,251	3,917	98.0
0	10^{-4}	75	937	1,004	71,251	1,381	96.5
0	10^{-6}	100	1,276	1,366	128,921	2,461	97.4
0	10^{-8}	126	1,608	1,721	204,251	3,877	98.0
10,000	10^{-4}	75	937	1,004	71,251	1,358	96.5
10,000	10^{-6}	100	1,276	1,366	128,921	2,424	97.4
10,000	10^{-8}	126	1,608	1,721	204,251	3,841	98.0

MRM of the fault-tolerant system with five control sites and $n_{\mathrm{HR}i} = 1$, $1 \leq i \leq 5$. The MRM has $|S| = 16{,}807$, $|S^0| = 992$, $|S^+| = 32$, $|S^-| = 15{,}783$, and 219,868 transitions. In Table 1 we give the values of the truncation parameters N^+, N^-, N, the cardinality $|D(N^+, N^-, N)|$ of the domain $D(N^+, N^-, N)$, the CPU times in seconds, and the percent of CPU time spent in computing the probabilities Q_{k^+,k^-}. The randomization rates were $\Lambda^+ = (1+10^{-3}) \max_{i \in S^+} -b_{i,i} = 1.528634 \times 10^{-3}\,\mathrm{h}^{-1}$ and $\Lambda^- = (1+10^{-3}) \max_{i \in S^-} -b_{i,i} = 2.153544 \times 10^{-2}\,\mathrm{h}^{-1}$, and the number of non-null elements of the matrix $\mathbf{P}$ was $K = 2{,}806{,}745$. We can note that the CPU times are reasonable and increase moderately as the error requirement becomes more stringent. In addition, as expected, most of the CPU time is spent computing the probabilities Q_{k^+,k^-}. The maximum memory consumption was about 369 MB.

In order to get some insight into how the cost of the method scales with the size of the MRM, we ran the method for the MRM of the fault-tolerant system with two, three, four, and five control sites and increasing numbers of Erlang stages. In Table 2, we give $|S|$, $|S^0|$, $|S^+|$, $|S^-|$, and the number K of non-null elements of $\mathbf{P}$ for each of the MRMs, and N^+, N^-, N, $|D(N^+, N^-, N)|$, and the CPU times in seconds for $s = -10{,}000$ and $\varepsilon = 10^{-8}$. As we can see, the CPU times range from a fraction of a second to $\approx 1.8\,\mathrm{h}$. The maximum memory consumption was about 540 MB. Although it is not possible to draw general conclusions from the results of a few MRMs, it seems that the method described in the paper can solve in reasonable CPU times MRMs with $\approx$ 100,000 states and $K \approx 3{,}750{,}000$.

The use of two randomization rates may reduce significantly the computational cost of the method. To illustrate this point, we ran the method for the MRM of the fault-tolerant system with five control sites and $n_{\mathrm{HR}i} = 1$, $1 \leq i \leq 5$, using a single randomization rate $\max\{\Lambda^+, \Lambda^-\}$.

Table 2: Results for the new method for the MRM of the fault-tolerant system with two, three, four, and five control sites, increasing numbers of Erlang stages, $s = -10{,}000$, and $\varepsilon = 10^{-8}$.

n_{s}	$n_{\mathrm{HR}i}$, $1 \leq i \leq n_{\mathrm{s}}$	$\lvert S\rvert$	$\lvert S^0\rvert$	$\lvert S^+\rvert$	$\lvert S^-\rvert$	K	N^+	N^-	N	$\lvert D(N^+, N^-, N)\rvert$	CPU time (s)
2	1	49	12	4	33	370	106	399	496	42,754	0.07
	2	64	12	4	48	441	106	456	552	48,843	0.11
	3	81	12	4	65	518	106	532	627	56,964	0.16
	4	100	12	4	84	601	106	607	703	65,000	0.22
3	1	343	56	8	279	7,843	113	767	869	87,485	1.25
	2	512	56	8	448	9,077	113	828	930	94,439	1.70
	3	729	56	8	665	10,581	113	909	1,010	103,661	2.34
	4	1,000	56	8	936	12,379	113	990	1,091	112,895	3.18
4	1	2,401	240	16	2,145	154,360	120	1,172	1,280	141,854	36.4
	2	4,096	240	16	3,840	170,921	120	1,236	1,344	149,598	59.5
	3	6,561	240	16	6,305	193,720	120	1,322	1,430	160,004	78.3
	4	10,000	240	16	9,744	224,089	120	1,408	1,515	170,397	117.8
5	1	16,807	992	32	15,783	2,806,745	126	1,608	1,721	204,251	3,917
	2	32,768	992	32	31,744	3,001,121	126	1,676	1,789	212,887	4,380
	3	59,049	992	32	58,025	3,303,687	126	1,766	1,879	224,317	5,312
	4	100,000	992	32	98,976	3,753,263	126	1,857	1,969	235,860	6,391

Table 3: Results for the new method for the MRM of the fault-tolerant system with five control sites, $n_{\mathrm{HR}i} = 1$, $1 \leq i \leq 5$, and $s = -10{,}000$ using a single randomization rate.

ε	N^+	N^-	N	$\lvert D(N^+, N^-, N)\rvert$	CPU time (s)
10^{-4}	1,133	937	1,980	1,059,596	20,020
10^{-6}	1,527	1,276	2,675	1,942,999	35,638
10^{-8}	1,916	1,608	3,359	3,070,757	57,000

That randomization rate was $\max\{1.528634 \times 10^{-3}\ \mathrm{h}^{-1}, 2.153544 \times 10^{-2}\ \mathrm{h}^{-1}\} = 2.153544 \times 10^{-2}\ \mathrm{h}^{-1}$. In Table 3 we give N^+, N^-, N, $|D(N^+, N^-, N)|$, and the CPU times in seconds for $s = -10{,}000$ and $\varepsilon = 10^{-4}, 10^{-6}, 10^{-8}$. Comparing with the results reported in Table 1, using a single randomization rate results in a large increase in N^+. The truncation parameter N also increases. Essentially, this is because the maximum output rate of Y in S^- is larger than in S^+ and using the same randomization rate in all states slows down significantly the pace at which $\widehat{Y}^+$ enters the absorbing state, resulting in a substantial increase in N^+, and the pace at which $\widehat{Y}$ enters the absorbing state, making N larger as well. As a consequence, $|D(N^+, N^-, N)|$ and, accordingly, the CPU times are much larger. Therefore, for this particular example, using different randomization rates in S^+ and S^- reduces significantly the computational cost of the method.

In view of the results of Table 3 and the ensuing discussion, it is clear that the smaller the randomization rates Λ^+ and Λ^- are, the smaller the domain $D(N^+, N^-, N)$ will be, making the method less costly. This raises the question of whether the use of randomization rates slightly larger than needed (see Section 2.6), which is important in order to ensure the numerical stability of the method, may have an important impact on its computational cost. To address this issue, we ran the method for the first three cases reported in Table 1 using $\Lambda^+ = (1 + \psi) \max_{i \in S^+} -b_{i,i}$ and $\Lambda^- = (1+\psi) \max_{i \in S^-} -b_{i,i}$ with $\psi = 0$ instead of $\psi = 10^{-3}$. The result was that taking $\psi = 10^{-3}$ yielded an average increase of about 0.7% in the CPU times. This seems to be a very reasonable price to pay for ensuring the numerical stability of the method.

To conclude this section, we will compare the new method proposed in this paper with the method based on the results given in [6]. We recall that these results are: 1) closed-form expressions for the probabilities $\Pr[R(T) \leq s \mid R(T) > 0]$, $s > 0$, $\Pr[-R(T) \leq s \mid R(T) < 0]$, $s > 0$, $\Pr[R(T) > 0]$, and $\Pr[R(T) < 0]$, and 2) an iterative algorithm to compute the two matrices, $\mathbf{\Psi}$ and $\mathbf{\Psi}_{\mathrm{r}}$, on which those closed-form expressions depend (the matrices themselves are independent of s). Both the matrices and the probabilities require $S^0 = \emptyset$ and thus have to be computed using the transformed CTMC Y (see Section 2.1). Once the probabilities have been obtained, $F(s)$ can be evaluated as

$$F(s) = \begin{cases} \Pr[R(T) < 0] + \beta_a + \Pr[R(T) \leq s \mid R(T) > 0] \times \Pr[R(T) > 0]\ , & s > 0 \\ \Pr[R(T) < 0] + \beta_a\ , & s = 0 \\ (1 - \Pr[-R(T) \leq -s \mid R(T) < 0]) \times \Pr[R(T) < 0]\ , & s < 0 \end{cases} . \quad (13)$$

Therefore, the alternative method consists of three steps: 1) construction of the transformed CTMC Y as described in Section 2.1, 2) computation of the matrices $\mathbf{\Psi}$ and $\mathbf{\Psi}_{\mathrm{r}}$ for the transformed CTMC Y using the iterative algorithm, and 3) evaluation of (13) on the transformed CTMC Y at the required s. We have found that the algorithm, which has to be applied once to obtain $\mathbf{\Psi}$ and a second time to obtain $\mathbf{\Psi}_{\mathrm{r}}$, breaks down in both cases because one of the involved matrices turns out to be singular. Using the same notation as in [6], that matrix is the matrix $\mathbf{I} - \mathbf{A}_1(0, \lambda)$ (see second and third lines of the listing given in [6, p. 254—255]), whose last row becomes $\mathbf{0}^{\mathrm{T}}$. To solve this problem, it is enough to set the bottom-most, right-most element of the matrix to 1. The algorithm has quadratic convergence and will typically require few steps. However, each step involves the inversion of a matrix of size $(|S^{\pm}| + 1) \times (|S^{\pm}| + 1)$. Therefore, the cost of the algorithm is $O(|S^{\pm}|^3)$ time and $O(|S^{\pm}|^2)$ memory. That cost will often dominate the computational cost of the alternative method. For MRMs with a number of states neither too small nor too large, the proposed method will often have a computational cost $O(KN^+N^-)$ time and $O(K + N^+N^- + (3 + \min\{N^+, N^-\})|S^{\pm}|)$ memory (see Section 2.6). Since it is difficult, in general, to anticipate the values of N^+, N^-, and K, it is not possible to conclude which method is the least costly in general. However, the fact that the time complexity of the second step of the alternative method is $O(|S^{\pm}|^3)$ suggests that for medium-size and large MRMs, the method proposed in this paper can be faster. To illustrate that indeed this can be the case, we implemented the alternative method using the well-known GNU Scientific Library [25] together with the high-performance basic linear algebra package ATLAS [26]. In our implementation, the matrices $\mathbf{\Psi}$ and $\mathbf{\Psi}_{\mathrm{r}}$ are computed using the iterative algorithm described in [6] with the previously commented modification. The required matrix inverses are computed by means of LU decomposition. The probabilities involved in (13) are evaluated as follows. Let $\mathbf{K} = \mathbf{B}_{+,+} + \mathbf{\Psi}\mathbf{B}_{-,+}$, $\mathbf{K}_{\mathrm{r}} = \mathbf{B}_{-,-} + \mathbf{\Psi}_{\mathrm{r}}\mathbf{B}_{+,-}$, $\boldsymbol{\gamma}^{\mathrm{T}} = (\boldsymbol{\beta}_+{}^{\mathrm{T}} + \boldsymbol{\beta}_-{}^{\mathrm{T}}\mathbf{\Psi}_{\mathrm{r}})(\mathbf{I} - \mathbf{\Psi}\mathbf{\Psi}_{\mathrm{r}})^{-1}$, and $\boldsymbol{\gamma}_{\mathrm{r}}{}^{\mathrm{T}} = (\boldsymbol{\beta}_+{}^{\mathrm{T}}\mathbf{\Psi} + \boldsymbol{\beta}_-{}^{\mathrm{T}})(\mathbf{I} - \mathbf{\Psi}_{\mathrm{r}}\mathbf{\Psi})^{-1}$, let $\mathbf{\Delta}$ be the diagonal matrix with diagonal elements equal to one minus the row sums of $\mathbf{\Psi}$, and let $\mathbf{\Delta}_{\mathrm{r}}$ be the diagonal matrix with diagonal elements equal to one minus the row sums of $\mathbf{\Psi}_{\mathrm{r}}$. Then, interpreting the transformed CTMC Y as an MRM in which the states in S^+ (resp., S^-) have reward rate equal to one (resp., minus one) and using Theorem 5.1 in [6], $\theta_r = \Pr[R(T) < 0] = \boldsymbol{\gamma}_{\mathrm{r}}{}^{\mathrm{T}}\mathbf{\Delta}_{\mathrm{r}}\mathbf{1}$, $\theta = \Pr[R(T) > 0] = \boldsymbol{\gamma}^{\mathrm{T}}\mathbf{\Delta}\mathbf{1}$, $\Pr[R(T) \leq s \mid R(T) > 0] = 1 - \boldsymbol{\zeta}^{\mathrm{T}}\mathrm{e}^{\mathbf{G}s}\mathbf{1}$, $s > 0$, where $\boldsymbol{\zeta}^{\mathrm{T}} = \theta^{-1}\boldsymbol{\gamma}^{\mathrm{T}}\mathbf{\Delta}$ and $\mathbf{G} = \mathbf{\Delta}^{-1}\mathbf{K}\mathbf{\Delta}$, and $1 - \Pr[-R(T) \leq -s \mid R(T) < 0] = \boldsymbol{\zeta}_{\mathrm{r}}{}^{\mathrm{T}}\mathrm{e}^{\mathbf{G}_{\mathrm{r}}(-s)}\mathbf{1}$, $s < 0$, where $\boldsymbol{\zeta}_{\mathrm{r}}{}^{\mathrm{T}} = \theta_{\mathrm{r}}^{-1}\boldsymbol{\gamma}_{\mathrm{r}}{}^{\mathrm{T}}\mathbf{\Delta}_{\mathrm{r}}$ and $\mathbf{G}_{\mathrm{r}} = \mathbf{\Delta}_{\mathrm{r}}^{-1}\mathbf{K}_{\mathrm{r}}\mathbf{\Delta}_{\mathrm{r}}$. Since $\boldsymbol{\zeta}$ and $\mathbf{G}$ can be interpreted as the restriction of the initial probability distribution vector and infinitesimal generator of some absorbing CTMC to its transient states, and $\boldsymbol{\zeta}_{\mathrm{r}}$ and $\mathbf{G}_{\mathrm{r}}$ can be interpreted similarly, the terms $\boldsymbol{\zeta}^{\mathrm{T}}\mathrm{e}^{\mathbf{G}s}\mathbf{1}$ and $\boldsymbol{\zeta}_{\mathrm{r}}{}^{\mathrm{T}}\mathrm{e}^{\mathbf{G}_{\mathrm{r}}(-s)}\mathbf{1}$ are computed with well-controlled error using the randomization technique. There are, then, two error control parameters in our implementation of the alternative method: one, ε_1, for the computation of the matrices $\mathbf{\Psi}$ and $\mathbf{\Psi}_{\mathrm{r}}$ using the iterative algorithm, and another, ε_2, for the computation of $\boldsymbol{\zeta}^{\mathrm{T}}\mathrm{e}^{\mathbf{G}s}\mathbf{1}$ and $\boldsymbol{\zeta}_{\mathrm{r}}{}^{\mathrm{T}}\mathrm{e}^{\mathbf{G}_{\mathrm{r}}(-s)}\mathbf{1}$ using randomization. If $\varepsilon_1 \ll \varepsilon_2$, we can expect the actual error in $F(s)$ to be $\leq \varepsilon_2$. Therefore, we set $\varepsilon_1 = 10^{-10}$ and ran the alternative method for the MRM of the fault-tolerant system with two, three, and four control sites and increasing number of Erlang stages for $s = -10{,}000$ and $\varepsilon_2 = 10^{-8}$. In Table 4, we give the CPU times in seconds for the alternative method and the percentage of those CPU times invested in the the computation of $\mathbf{\Psi}$ and $\mathbf{\Psi}_{\mathrm{r}}$. Comparing with the results reported in Table 2, for the smallest MRMs ($|S^{\pm}| \leq 88$) the proposed method is slower than

Table 4: Results for the alternative method for the MRM of the fault-tolerant system with two, three, and four control sites and increasing numbers of Erlang stages, $s = -10{,}000$, $\varepsilon_1 = 10^{-10}$, and $\varepsilon_2 = 10^{-8}$.

n_{s}	$n_{\mathrm{HR}i}$, $1 \leq i \leq n_{\mathrm{s}}$	$\lvert S^{\pm}\rvert$	CPU time (s)	% $\mathbf{\Psi}$, $\mathbf{\Psi}_{\mathrm{r}}$
2	1	37	0.02	100
	2	52	0.04	90.0
	3	69	0.07	94.1
	4	88	0.14	97.1
3	1	287	1.96	98.4
	2	456	7.52	98.9
	3	673	29.3	99.1
	4	944	131.9	99.0
4	1	2,161	1,879	98.8
	2	3,856	11,646	98.9
	3	6,321	53,305	98.9

the alternative method. However, since, as expected, the CPU time of the latter increases sharply with $|S^{\pm}|$, the proposed method is faster for the remaining, larger MRMs, with a speedup ranging from 1.6 ($|S^{\pm}| = 287$) to 681 ($|S^{\pm}| = 6{,}321$). These results confirm that for medium-size and large MRMs the proposed method can be substantially faster than the alternative method, allowing the analysis with affordable computational resources of MRMs for which the alternative method is very expensive. Another advantage of the proposed method is that it has well-controlled error and is numerically stable, whereas in the alternative method there is no direct relationship between the error control parameter ε_1 and the actual error in $F(s)$. Besides, the method could be unstable because the iterative algorithm for the computation of the matrices $\mathbf{\Psi}$ and $\mathbf{\Psi}_{\mathrm{r}}$ involves subtractions. In spite of this fact, we assessed the actual precision of the alternative method by comparing the value of $F(s)$ yielded by the proposed method with $\varepsilon = 10^{-10}$ and the one obtained with the alternative method with $\varepsilon_1 = 10^{-10}$ and $\varepsilon_2 = 10^{-8}$ for the MRMs for which results are reported in Table 4 for a set of 500 equally spaced abscissas ranging from $-10{,}000$ to 100,000. The result has been a maximum difference, in absolute value, between the two values of $F(s)$ equal to 5.3×10^{-9}. Therefore, it seems that, at least for the MRMs we have tried, the alternative method is accurate.

4 Conclusions

We have developed a new method to compute the distribution function of the reward accumulated till exit of a subset of transient states of an MRM with a reward structure including only reward rates associated with states, in which both positive and negative reward rates are present and null reward rates are allowed. The new method combines a model transformation step with the solution of the transformed model using a randomization construction with two randomization rates. The method introduces a truncation error, but that error is strictly bounded from above by a user-specified error control parameter. In addition, the method is numerically stable and takes advantage of the sparsity of the infinitesimal generator of the transformed model. We have illustrated the application of the method using an MRM of a hardware/software fault-tolerant system. We have also analyzed the computational cost of the method and have compared it with that of the (only) alternative method for the problem. Our numerical experiments seem to indicate that the new method can be efficient and that for medium-size and large MRMs can be substantially faster than the alternative method.

Acknowledgments

This research work has been supported by the Ministry of Science and Technology of Spain and Fondo Europeo de Desarrollo Regional (FEDER) under the research grant DPI2004-05077 and by the Ministry of Science and Innovation of Spain and FEDER under the research grant TIN2008-06735-C02-02. The authors are also grateful to the anonymous referees, whose comments have helped to improve the quality and readability of the paper.

References

[1] M. D. Beaudry, "Performance-related reliability measures for computing systems," *IEEE Trans. Comput.*, vol. 27, no. 6, pp. 540–547, 1978.

[2] M. Malhotra, "A computationally efficient technique for transient analysis of repairable Markovian systems," *Performance Evaluation*, vol. 24, no. 4, pp. 311–331, 1996.

[3] A. Reibman and K. Trivedi, "Numerical transient analysis of Markov models," *Comput. & Opns Res.*, vol. 15, no. 1, pp. 19–36, 1988.

[4] V. G. Kulkarni, "A new class of multivariate phase type distributions," *Operations Research*, vol. 37, no. 1, pp. 151–158, 1989.

[5] G. Ciardo, R. A. Marie, B. Sericola, and K. S. Trivedi, "Performability analysis using semi-Markov reward processes," *IEEE Trans. Comput.*, vol. 39, no. 10, pp. 1251–1264, 1990.

[6] S. Ahn and V. Ramaswami, "Bilateral phase type distributions," *Stochastic Models*, vol. 21, no. 2, pp. 239–259, 2005.

[7] J. F. Meyer, "On evaluating the performability of degradable computing systems," *IEEE Trans. Comput.*, vol. C-29, no. 8, pp. 720–731, 1980.

[8] R. M. Smith, K. S. Trivedi, and A. V. Ramesh, "Performability analysis: Measures, an algorithm, and a case study," *IEEE Trans. Comput.*, vol. 37, no. 4, pp. 406–417, 1988.

[9] E. de Souza e Silva and H. R. Gail, "Calculating availability and performability measures of repairable computer systems using randomization," *J. of the ACM*, vol. 36, no. 1, pp. 171–193, 1989.

[10] S. M. R. Islam and H. H. Ammar, "Performability of the hypercube," *IEEE Trans. Rel.*, vol. 38, no. 5, pp. 518–526, 1989.

[11] L. Donatiello and V. Grassi, "On evaluating the cumulative performance distribution of fault-tolerant computer systems," *IEEE Trans. Comput.*, vol. 40, no. 11, pp. 1301–1307, 1991.

[12] K. R. Pattipati, Y. Li, and H. A. P. Blom, "A unified framework for the performability evaluation of fault-tolerant computer systems," *IEEE Trans. Comput.*, vol. 42, no. 3, pp. 312–326, 1993.

[13] H. Nabli and B. Sericola, "Performability analysis: A new algorithm," *IEEE Trans. Comput.*, vol. 45, no. 4, pp. 491–494, 1996.

[14] V. Suñé, J. A. Carrasco, H. Nabli, and B. Sericola, "Comment on 'Performability analysis: A new algorithm'," *IEEE Trans. Comput.*, vol. 59, no. 1, pp. 137–138, 2010.

[15] M. A. Qureshi and W. H. Sanders, "A new methodology for calculating distributions of reward accumulated during a finite interval," in *Proc. 26th Int. Symp. on Fault-Tolerant Computing*. IEEE, June 1996, pp. 116–125.

[16] E. de Souza e Silva and H. R. Gail, "An algorithm to calculate transient distributions of cumulative rate and impulse based reward," *Stochastic Models*, vol. 14, no. 3, pp. 509–536, 1998.

[17] S. Rácz, Á. Tari, and M. Telek, *Computer Performance Evaluation: Modelling Techniques and Tools*, ser. Lecture Notes in Computer Science. Springer Verlag, 2002, vol. 2324, ch. MRMSolve: Distribution Estimation of Large Markov Reward Models, pp. 141–158.

[18] J. A. Carrasco, "Two methods for computing bounds for the distribution of cumulative reward for large Markov models," *Performance Evaluation*, vol. 63, no. 12, pp. 1165–1195, 2006.

[19] W. K. Grassmann, "Transient solutions in Markovian queueing systems," *Comput. & Opns Res.*, vol. 4, no. 1, pp. 47–53, 1977.

[20] M. Kijima, *Markov Processes for Stochastic Modeling*. Chapman & Hall, 1997.

[21] N. J. Higham, "The accuracy of floating point summation," *SIAM J. Sci. Comput.*, vol. 14, no. 4, pp. 783–799, 1993.

[22] *IEEE Std 754-2008 Standard for Floating-Point Arithmetic (Revision of IEEE Std 754-1985)*, IEEE Computer Society, 2008.

[23] T. A. Davis, *Direct Methods for Sparse Linear Systems*, ser. Fundamentals of Algorithms. SIAM, 2006.

[24] I. Koren and C. M. Krishna, *Fault-Tolerant Systems*. Morgan Kaufmann Publishers, 2007.

[25] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, and F. Rossi, *GNU Scientific Library Reference Manual. Third edition, for version 1.12*. Network Theory Ltd, 2009.

[26] R. C. Whaley, A. Petitet, and J. J. Dongarra, "Automated empirical optimizations of software and the ATLAS project," *Parallel Computing*, vol. 27, no. 1-2, pp. 3–35, 2001.