Computers & Operations Research 38 (2011) 651–659

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Numerical analysis of continuous time Markov decision processes over finite horizons

Peter Buchholz *, Ingo Schulz

Department of Computer Science, TU Dortmund, D-44221 Dortmund, Germany

ARTICLE INFO

Available online 26 August 2010

Keywords:
Uniformization
Continuous time Markov decision processes
Finite horizon
Error bounds

ABSTRACT

Continuous time Markov decision processes (CTMDPs) with a finite state and action space have been considered for a long time. It is known that under fairly general conditions the reward gained over a finite horizon can be maximized by a so-called piecewise constant policy which changes only finitely often in a finite interval. Although this result is available for more than 30 years, numerical analysis approaches to compute the optimal policy and reward are restricted to discretization methods which are known to converge to the true solution if the discretization step goes to zero. In this paper, we present a new method that is based on uniformization of the CTMDP and allows one to compute an ε−optimal policy up to a predefined precision in a numerically stable way using adaptive time steps.

1. Introduction

Continuous time Markov decision processes (CTMDPs) are a widely used model type with applications in the control of queuing systems, the computation of maintenance policies and related optimization and control problems. The general model is described in several textbooks [1–4]. CTMDPs are mainly analyzed over infinite horizons such that a policy has to be found which maximizes the average or discounted reward. It can be shown that under fairly general conditions the optimal policy is stationary, i.e., it depends only on the state and can be computed from a discrete time Markov decision process (DTMDP) which results from the original CTMDP using uniformization [5,6]. For the DTMDP standard algorithms like value or policy iteration [2] can be applied to find the optimal policy. The correspondence between continuous and discrete time MDPs for infinite horizons has been established some time ago in [7,8].

Much less work about CTMDPs on finite horizons is available. In [9] it is shown that under general conditions the optimal policy for a CTMDP with finite state space and finite action set is piecewise constant. This result has been extended in [10] to Borel state spaces and compact action sets. In [11] the convergence of the policy for the horizon going to infinity is considered. In contrast to the infinite horizon case, for finite horizons a direct correspondence between the CTMDP and the DTMDP resulting from uniformization does not hold. Thus, the resulting DTMDP cannot be immediately applied for the computation of the optimal policy. The computation of the optimal policy in CTMDPs over finite horizons is rarely considered. In [9,12] time discretization is used and the optimal policy and reward are computed from the DTMDP resulting from discretization. It is shown that for the discretization step going to zero the reward and policy converge towards the exact values. Lippman [13,14] applies uniformization to generate a DTMDP for the computation of the optimal policy which allows one to extend the previous result to countable state spaces. However, he considers uniformization with a fixed number of transitions in the DTMDP such that the resulting reward and policy are only approximations which converge towards the true values if the uniformization rate and therefore the number of steps tend to infinity. Additionally, simulation can also be applied to determine the optimal policy and reward [15]. All the mentioned approaches compute approximations and it is not clear how good the approximations are in a concrete situation with a given CTMDP and a fixed discretization step since error bounds are hard to compute for general models. Furthermore, discretization is usually applied in a non-adaptive way such that many iterations are necessary to obtain a good approximation even if the policy remains constant.

For the transient analysis of continuous time Markov chains (CTMCs) uniformization is also used and has several times shown to be the best method for a wide variety of models [5,16]. In this case, the transient distribution is computed from a Poisson process and the transitions of the discrete time Markov chain (DTMC) resulting from uniformization. Nice properties of the method are its numerical stability and the possibility to compute a priori the required number of iterations to reach a predefined accuracy. Uniformization as a transient solution method can, of course, be applied to compute the reward for a known piecewise constant policy which is nothing else than computing the

* Corresponding author.
E-mail addresses: peter.buchholz@tu-dortmund.de (P. Buchholz), ingo.schulz@tu-dortmund.de (I. Schulz).

0305-0548/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2010.08.011

transient solution of a CTMC where the generator matrix changes at discrete time points [17]. Furthermore, uniformization has recently also been applied for the transient analysis of inhomogeneous CTMCs [18].

In this paper, we present a new approach to apply the idea of uniformization to compute an ε-optimal piecewise constant policy (i.e., a policy which results in a reward that is at most ε smaller than the maximal reward that is reached under an optimal policy) and the corresponding reward value. In contrast to the known discretization techniques, the new approach computes an error bound and uses an adaptive step length control that chooses large time steps when the policy remains constant.

The paper is organized as follows. In the next section, we present some basic definitions and notations and introduce known results about optimal policies. Section 3 present a discretization approach to approximate the optimal policy. Afterwards, in Section 4, we present our new algorithm based on uniformization. Then some examples are introduced to compare both techniques for policy optimization. The paper ends with the conclusions and an overview of possible extensions and open problems.

2. Basic definitions and known results

We consider CTMDPs with a finite state space $\mathcal{S}=\{1,\ldots,n\}$. In each state $i\in\mathcal{S}$ a decision u can be chosen from m_i different decisions collected in the set $\mathcal{D}_i$. For the ease of notation we number decisions consecutively. If decision u $(1\le u\le m_i)$ is chosen in state i, then transition rates out of state i are defined in a vector $\mathbf{q}_i^u\in\mathbb{R}^{1,n}$ such that $0\le\mathbf{q}_i^u(j)<\infty$ $(i\neq j)$ is the rate from state i to j under decision u. Furthermore, let $\mathbf{q}_i^u(i)=-\sum_{j\neq i}\mathbf{q}_i^u(j)$. If $\mathbf{d}$ is an n-dimensional vector with $1\le\mathbf{d}(i)\le m_i$ the decision taken in state i, then $\mathbf{Q}^{\mathbf{d}}\in\mathbb{R}^{n,n}$ with $\mathbf{Q}^{\mathbf{d}}(i,j)=\mathbf{q}_i^{\mathbf{d}(i)}(j)$ is the generator matrix of a CTMC. Let $\mathcal{D}=\times_{i=1}^{n}\mathcal{D}_i$ be the finite decision space of the CTMDP. We use the notation $\mathbf{d}\in\mathcal{D}$ for n-dimensional vectors $\mathbf{d}$ with $\mathbf{d}(i)\in\mathcal{D}_i$ and denote $\mathbf{d}$ as a decision vector. The reward of the system depends only on the current state. Thus, let $\mathbf{r}\in\mathbb{R}_+^{n,1}$ be a reward vector such that $\mathbf{r}(i)$ is the reward of staying for one time unit in state i. The model will be analyzed in the interval $[0,T]$. A policy π is a mapping from $[0,T]$ into $\mathcal{D}$ and $\mathbf{d}_t$ is the decision vector at time $t\in[0,T]$, i.e., $\mathbf{d}_t(i)$ is the decision taken if the system is in state i at time t. We require that π is measurable function where measurable means Lebesgue measurable. This implies that the following equations result in unique and continuous solutions for every policy π of the mentioned type [11,9]. Let $\mathcal{M}$ be the set of all measurable policies on $[0,T]$.

A policy is piecewise constant if there exist some $m<\infty$ and $0=t_0<t_1<t_2<\cdots<t_{m-1}<t_m=T<\infty$ such that for $t,t'\in(t_i,t_{i+1}]$ $(0\le i<m)$ $\mathbf{d}_t=\mathbf{d}_{t'}$. The policy is stationary if $m=1$.

For a given policy π let $\mathbf{V}_{s,t}^{\pi}$ with $0\le s\le t\le T$ be a matrix that contains in position (i,j) the probability that the process is at time t in state j under the condition that it was at time s in state i. According to [9] $\mathbf{V}_{s,t}^{\pi}$ is defined by the following differential equations:

$$\frac{d}{dt}\mathbf{V}_{s,t}^{\pi}=\mathbf{V}_{s,t}^{\pi}\mathbf{Q}^{\mathbf{d}_t} \tag{1}$$

with the initial condition $\mathbf{V}_{s,s}^{\pi}=\mathbf{I}$. $\mathbf{V}_{s,t}^{\pi}$ is an absolutely continuous transition matrix defined almost everywhere in $[0,T]$. We use the notation $\mathbf{V}_t^{\pi}$ for $\mathbf{V}_{0,t}^{\pi}$. For some initial distribution $\mathbf{p}_0$,

$$\mathbf{p}_t=\mathbf{p}_0\mathbf{V}_t^{\pi} \tag{2}$$

is the distribution at time t under policy π. Define

$$\mathbf{g}^{\pi}=\int_0^T\mathbf{V}_t^{\pi}\mathbf{r}\,dt \tag{3}$$

as the gain per state of policy π and

$$G^{\pi}=\mathbf{p}_0\mathbf{g}^{\pi} \tag{4}$$

as the gain of policy π with initial distribution $\mathbf{p}_0$. Vector $\mathbf{g}^{\pi}$ will be denoted as the gain vector. The goal is to define a policy $\pi\in\mathcal{M}$ that maximizes (3) in every component.

The following theorem summarizes the main results of [9, Theorems 1 and 6].

Theorem 1. *A policy is optimal if it maximizes for almost all* $t\in[0,T]$

$$\max_{\pi\in\mathcal{M}}(\mathbf{Q}^{\mathbf{d}_t}\mathbf{g}_t+\mathbf{r})\quad\text{where}\quad-\frac{d}{dt}\mathbf{g}_t=\mathbf{Q}^{\mathbf{d}_t}\mathbf{g}_t+\mathbf{r}\quad\text{and}\quad\mathbf{g}_T=\mathbf{0}. \tag{5}$$

There exists a piecewise constant policy $\pi\in\mathcal{M}$ *that maximizes the equations.*

In [9] a constructive proof of Theorem 1 is presented that can be used as a base for algorithms to approximate the optimal policy. For some measurable policy π with gain vector $\mathbf{g}_t$ at time t define the following sets:

$$\mathcal{F}_1(\mathbf{g}_t)=\{\mathbf{d}\in\mathcal{D}|\mathbf{d}\text{ maximizes }\mathbf{q}_{\mathbf{d}}^{(1)}\},$$
$$\mathcal{F}_2(\mathbf{g}_t)=\{\mathbf{d}\in\mathcal{F}_1(\mathbf{g}_t)|\mathbf{d}\text{ maximizes }-\mathbf{q}_{\mathbf{d}}^{(2)}\},$$
$$\vdots$$
$$\mathcal{F}_j(\mathbf{g}_t)=\{\mathbf{d}\in\mathcal{F}_{j-1}(\mathbf{g}_t)|\mathbf{d}\text{ maximizes }(-1)^{j-1}\mathbf{q}_{\mathbf{d}}^{(j)}\},$$

where

$$\mathbf{q}_{\mathbf{d}}^{(1)}=\mathbf{Q}^{\mathbf{d}}\mathbf{g}_t+\mathbf{r},\ \mathbf{q}_{\mathbf{d}}^{(j)}=\mathbf{Q}^{\mathbf{d}}\mathbf{q}^{(j-1)}\quad\text{and}$$
$$\mathbf{q}^{(j-1)}=\mathbf{q}_{\mathbf{d}}^{(j-1)}\quad\text{for any }\mathbf{d}\in\mathcal{F}_{j-1}(j=2,3,\ldots).$$

Vector $\mathbf{q}_{\mathbf{d}}^{(i)}$ contains the values of the ith derivative of $\mathbf{g}_t$ multiplied with $(-1)^{i-1}$ when policy $\mathbf{d}$ is chosen. Knowing $\mathbf{g}_t$ and the derivatives at t, the vector $\mathbf{g}_{t'}$ for $t'<t$ under decision vector $\mathbf{d}$ can be expressed by a Taylor series expansion:

$$\mathbf{g}_{t'}=\mathbf{g}_t+\sum_{i=1}^{\infty}\frac{(t'-t)^i}{i!}\mathbf{q}_{\mathbf{d}}^{(i)}. \tag{6}$$

The following theorem follows from Lemmas 3 and 4 of [9].

Theorem 2. *If* $\mathbf{d}\in\mathcal{F}_{n+1}(\mathbf{g}_t)\Rightarrow\mathbf{d}\in\mathcal{F}_{n+k}(\mathbf{g}_t)$ *for all* $k>1$.

Let π *be a measurable policy in* $(t',T]$ *and assume that* $\mathbf{d}\in\mathcal{F}_{n+1}(\mathbf{g}_t)$ *for* $t'<t<T$, *then exists some* ε $(0<\varepsilon\le t-t')$ *such that* $\mathbf{d}\in\mathcal{F}_{n+1}(\mathbf{g}_{t''})$ *for all* $t''\in[t-\varepsilon,t]$.

In order to select a unique policy, some ordering has to be defined on $\mathcal{D}$ such that we can select the policy with the smallest index from $\mathcal{F}_{n+1}(\mathbf{g}_t)$. Since our decision vectors are coded as integer vectors, it is natural to consider the canonical ordering. We can define a *selection procedure* that selects the lexicographically smallest vector $\mathbf{d}$ from $\mathcal{F}_{n+1}(\mathbf{g}_t)$. Then the following algorithm can be defined (see [9]):

1. Set $t'=T$ and $\mathbf{g}_T=\mathbf{0}$.
2. Select $\mathbf{d}_{t'}$ using $\mathbf{g}_{t'}$ from $\mathcal{F}_{n+1}(\mathbf{g}_{t'})$ as described.
3. Obtain $\mathbf{g}_t$ for $0\le t\le t'$ by solving

   $$-\frac{d}{dt}\mathbf{g}_t=\mathbf{r}+\mathbf{Q}^{\mathbf{d}_{t'}}\mathbf{g}_t$$

   with terminal condition $\mathbf{g}_{t'}$.
4. Set $t''=\inf\{t:\mathbf{d}_t\text{ satisfies the selection procedure in }(t'',t']\}$.
5. If $t''\le 0$ terminate, else go to 2. with $t'=t''$.

We denote by $\mathbf{g}^*{}_0$ the gain vector at $t=0$ and by π^* the piecewise constant policy resulting from the above algorithm. The optimal gain for a given initial vector $\mathbf{p}_0$ equals then $G^*=\mathbf{p}_0\mathbf{g}_0^*$. According

to the Bellman equations [1] the restriction of policy π^* to the interval $(t,T]$ $(0<t<T)$ results in an optimal policy with gain vector $\mathbf{g}_t{}^*$.

Although it is assured that the algorithm goes through the loop (2) through (5) only finitely many times, it is not practical since step (4) is not really implementable. In the following two sections we present two practical algorithms which can be implemented. However, the price is the introduction of some approximation error.

3. Discretization approaches for policy computation

As already mentioned, the optimal vector $\mathbf{g}^*$ and the optimal policy π^* cannot be effectively computed. We define a policy π as ε-optimal if $\|\mathbf{g}^*-\mathbf{g}^\pi\|_\infty \le \varepsilon$. The goal is to compute ε-optimal policies. In this section we present an approach known from literature before we introduce in the following section our own new approach.

A simple approach which is suggested in [9] and also used in [12] uses discretization. Define for $h>0$

$$\mathbf{P}_h^{\mathbf{d}} = \mathbf{I} + h\mathbf{Q}^{\mathbf{d}} \tag{7}$$

as the transition matrix of the discrete time process with decision vector $\mathbf{d}$. For h sufficiently small, $\mathbf{P}_h^{\mathbf{d}}$ is a stochastic matrix and for a known gain vector $\mathbf{g}_t$ at time t and decision vector $\mathbf{d}$ which is used in the interval $(t-h,t]$ we obtain the following approximation for the gain vector $\mathbf{g}_{t-h}$:

$$\mathbf{g}_{t-h} = \mathbf{P}_h^{\mathbf{d}}\mathbf{g}_t + h\mathbf{r} + o(h). \tag{8}$$

For some h which is small enough to result in stochastic matrices $\mathbf{P}_h^{\mathbf{d}}$ for every $\mathbf{d}\in\mathcal{D}$, the matrices $\mathbf{P}_h^{\mathbf{d}}$ define DTMDPs. Define $h=T/N$ for N large enough and the resulting DTMDP can be solved with the standard algorithm for DTMDPs [1] for a horizon of N steps. The algorithm computes an ε-optimal policy with ε going to 0 for $h\to 0$ [12]. However, for a fixed h, the value of ε can hardly be determined.

For an average number c of non-zero entries per row of the matrices $\mathbf{Q}^{\mathbf{d}}$, the overall effort of the approach is in $O(h^{-1}nc(\sum_{i=1}^n m_i))$ where nc equals the average number of non-zero elements in $\mathbf{Q}^{\mathbf{d}}$.

4. Uniformization of CTMDPs

Discretization has two disadvantages. First, the error bound ε cannot be determined or the other way round, it is not possible to choose h such that a predefined error bound ε will hold. Second, the discretization step h is constant and does not take into account that the discretization error may depend on differences in the gain resulting from choosing different decision vectors at time t.

We develop an approach which computes an ε-optimal policy for a predefined $\varepsilon>0$ in an adaptive way. The approach is based on uniformization [6,5,18] for the transient analysis of continuous time Markov chains (CTMCs) which allows one to compute the transient solution vector with a predefined error. Although uniformization has been applied for CTMDPs [13,14] as already explained, numerical algorithms for the computation of optimal policies and gains in the finite horizon case do not exist, to the best of our knowledge. In the context of model checking CTMCs, [19] developed an approach for uniform CTMDPs, i.e., CTMDPs where all diagonal elements of the generator matrix are identical. The policies that are considered do not explicitly depend on timing information which we do assume here. Furthermore, the approach of [19] cannot be easily extended to nonuniform CTMDPs.

Assume that the vector $\mathbf{g}_t$ is known and the decision vector $\mathbf{d}$ remains constant in the interval $(t-\delta,t]$, then

$$\mathbf{g}_{t-\delta}^{\mathbf{d}} = e^{\delta\mathbf{Q}^{\mathbf{d}}}\mathbf{g}_t + \int_{\tau=t-\delta}^{t} e^{(\tau-(t-\delta))\mathbf{Q}^{\mathbf{d}}}\mathbf{r}\,d\tau = e^{\delta\mathbf{Q}^{\mathbf{d}}}\mathbf{g}_t + \int_{\tau=0}^{\delta} e^{\tau\mathbf{Q}^{\mathbf{d}}}\mathbf{r}\,d\tau. \tag{9}$$

The first term describes the reward that has been accumulated in the interval $(t, T]$ and the second term describes the reward that is accumulated in the interval $(t-\delta,t]$. Under the assumption that the policy does not change in the interval $(t-\delta,t]$, the reward as well as the conditional distribution at the end of the interval can be computed from a homogeneous CTMC. The above equation can be solved by uniformization. Define $\alpha \ge \max_{\mathbf{d}\in\mathcal{D}}\max_{i\in\mathcal{S}}|\mathbf{Q}^{\mathbf{d}}(i,i)|$, by assumption $\alpha<\infty$ can always be found. Then

$$\mathbf{P}^{\mathbf{d}} = \mathbf{Q}^{\mathbf{d}}/\alpha + \mathbf{I} \tag{10}$$

is a stochastic matrix. Define $\beta(\alpha t,k) = e^{-\alpha t}(\alpha t)^k/k!$ as the probability that a Poisson process with parameter αt performs k jumps. Eq. (9) can then be transformed as follows:

$$\begin{aligned}
\mathbf{g}_{t-\delta}^{\mathbf{d}} &= e^{\delta\mathbf{Q}^{\mathbf{d}}}\mathbf{g}_t + \int_{\tau=0}^{\delta} e^{\tau\mathbf{Q}^{\mathbf{d}}}\mathbf{r}\,d\tau \\
&= \sum_{k=0}^{\infty}\beta(\alpha\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{g}_t + \sum_{k=0}^{\infty}(\mathbf{P}^{\mathbf{d}})^k\mathbf{r}\left(\sum_{l=k}^{\infty}\beta(\alpha\delta,l)\frac{\delta}{l+1}\right) \\
&= \sum_{k=0}^{\infty}\beta(\alpha\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{g}_t + \sum_{k=0}^{\infty}(\mathbf{P}^{\mathbf{d}})^k\mathbf{r}\left(\sum_{l=k}^{\infty}e^{-\alpha\delta}\frac{(\alpha\delta)^{l+1}}{\alpha((l+1)!)}\right) \\
&= \sum_{k=0}^{\infty}\beta(\alpha\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{g}_t + \frac{1}{\alpha}\sum_{k=0}^{\infty}\left(1-\sum_{l=0}^{k}\beta(\alpha\delta,l)\right)(\mathbf{P}^{\mathbf{d}})^k\mathbf{r}.
\end{aligned} \tag{11}$$

The representation of the integrals by the sums follows from [5, Eq. (3)] and [5, Theorem 1]. Observe that both sums in the last row of (11) contain only positive values such that any truncation of the infinite sums results in a lower bound for $\mathbf{g}_{t-\delta}^{\mathbf{d}}$. For notational convenience we define

$$\zeta(\alpha,\delta,K) = \frac{1}{\alpha}\left(1-\sum_{l=0}^{K}\beta(\alpha\delta,l)\right). \tag{12}$$

Now assume that bounds $\underline{\mathbf{g}}_t \le \mathbf{g}_t \le \overline{\mathbf{g}}_t$ are known. For $t=T$ we can trivially define those bounds since $\mathbf{g}_T=\mathbf{0}$ is assumed to be known. Then we define the following vectors:

$$\underline{\mathbf{v}}^{(k)} = \mathbf{P}^{\mathbf{d}}\underline{\mathbf{v}}^{(k-1)} \quad \text{with } \underline{\mathbf{v}}^{(0)} = \underline{\mathbf{g}}_t \quad \text{and} \quad \underline{\mathbf{w}}^{(k)} = \mathbf{P}^{\mathbf{d}}\underline{\mathbf{w}}^{(k-1)} \quad \text{with } \underline{\mathbf{w}}^{(0)} = \mathbf{r}. \tag{13}$$

If not stated otherwise, we choose the decision vector $\mathbf{d}$ that is the lexicographically smallest vector from $\mathcal{F}_{n+1}(\underline{\mathbf{g}}_t)$. Furthermore define

$$\begin{aligned}
\overline{\mathbf{v}}^{(k)} &= \max_{\mathbf{d}\in\mathcal{D}}(\mathbf{P}^{\mathbf{d}}\overline{\mathbf{v}}^{(k-1)}) \quad \text{with } \overline{\mathbf{v}}^{(0)} = \overline{\mathbf{g}}_t, \\
\overline{\mathbf{w}}^{(k)} &= \max_{\mathbf{d}\in\mathcal{D}}(\mathbf{P}^{\mathbf{d}}\overline{\mathbf{w}}^{(k-1)}) \quad \text{with } \overline{\mathbf{w}}^{(0)} = \mathbf{r}.
\end{aligned} \tag{14}$$

Both vector sequences can be easily computed since the problem is equivalent to the computation of an optimal policy in a DTMDP over a finite horizon [1]. The effort for computing vectors $\overline{\mathbf{v}}^{(0)},\ldots,\overline{\mathbf{v}}^{(K)}$ and $\overline{\mathbf{w}}^{(0)},\ldots,\overline{\mathbf{w}}^{(K)}$ is in $O(Knc(\sum_{i=1}^n m_i))$. Obviously, $\underline{\mathbf{v}}^{(k)} \le \overline{\mathbf{v}}^{(k)}$ and $\underline{\mathbf{w}}^{(k)} \le \overline{\mathbf{w}}^{(k)}$ for all $k=0,1,\ldots$.

The following theorem shows how to compute error bounds, if the summation is truncated, the decision vector equals $\mathbf{d}$ in $(t-\delta,t]$ and bounds for $\mathbf{g}_t$ are known.

Theorem 3. *The following elementwise bounds can be computed for vector* $\mathbf{g}^{\mathbf{d}}_{t-\delta}$:

$$\underline{\mathbf{g}}^{\mathbf{d},K}_{t-\delta} = \sum_{k=0}^{K} \beta(\alpha\delta,k)\underline{\mathbf{v}}^{(k)} + \sum_{k=0}^{K} \zeta(\alpha,\delta,k)\underline{\mathbf{w}}^{(k)} + (\underline{\gamma}^K_{t,\delta} + \underline{\eta}^K_{\delta})\mathbf{e}^T \le \mathbf{g}^{\mathbf{d}}_{t,\delta}$$
$$\le \sum_{k=0}^{K} \beta(\alpha\delta,k)\overline{\mathbf{v}}^{(k)} + \sum_{k=0}^{K} \zeta(\alpha,\delta,k)\overline{\mathbf{w}}^{(k)} + (\overline{\gamma}^K_{t-\delta} + \overline{\eta}^K_{\delta})\mathbf{e}^T = \overline{\mathbf{g}}^{\mathbf{d},K}_{t-\delta},$$

where decision vector $\mathbf{d}$ *is used for the computation of* $\underline{\mathbf{v}}^{(k)}, \underline{\mathbf{w}}^{(k)}$ and

$$\underline{\gamma}^K_{t,\delta} = \varepsilon_1(\alpha\delta,K)\min_{i\in\mathcal{S}}(\underline{\mathbf{v}}^{(K)}(i)),$$
$$\overline{\gamma}^K_{t,\delta} = \varepsilon_1(\alpha\delta,K)\max_{i\in\mathcal{S}}(\overline{\mathbf{v}}^{(K)}(i)),$$
$$\underline{\eta}^K_{\delta} = \varepsilon_2(\alpha\delta,K)\min_{i\in\mathcal{S}}(\underline{\mathbf{w}}^{(K)}(i)),$$
$$\overline{\eta}^K_{\delta} = \varepsilon_2(\alpha\delta,K)\max_{i\in\mathcal{S}}(\overline{\mathbf{w}}^{(K)}(i)),$$
$$\varepsilon_1(\alpha\delta,K) = 1 - \sum_{k=0}^{K} \beta(\alpha\delta,k),$$
$$\varepsilon_2(\alpha,\delta,K) = \left(\delta\left(1 - \sum_{k=0}^{K}\beta(\alpha\delta,k)\right) - \frac{K+1}{\alpha}\left(1 - \sum_{k=0}^{K+1}\beta(\alpha\delta,k)\right)\right).$$

Proof. Since $\mathbf{P}_{\mathbf{d}}$ is stochastic, multiplication with a vector from the right results in a vector with elements that are bounded from below and above by the elements of the vector that is multiplied. Additionally, we have $\sum_{k=K+1}^{\infty}\beta(\alpha\delta,k) = 1 - \sum_{k=0}^{K}\beta(\alpha\delta,k)$. Consequently, every element of $\sum_{k=K+1}^{\infty}(\mathbf{P}^{\mathbf{d}})^k\underline{\mathbf{g}}_t$ is bounded by the elements in $\underline{\mathbf{v}}^{(K)}$ and $\overline{\mathbf{v}}^{(K)}$. Furthermore, every element of $\sum_{k=K+1}^{\infty}\beta(\alpha\delta,k)(\mathbf{P}^{\mathbf{d}})^k\underline{\mathbf{g}}_t$ is bounded by $\underline{\gamma}^K_{t-\delta}$ and $\overline{\gamma}^K_{t-\delta}$, respectively.

It remains to compute bounds for the accumulated reward in the interval. The upper bound is derived from

$$\sum_{k=0}^{\infty}\zeta(\alpha,\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{r} = \sum_{k=0}^{K}\zeta(\alpha,\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{r} + \sum_{k=K+1}^{\infty}\zeta(\alpha,\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{r}$$
$$\le \sum_{k=0}^{K}\zeta(\alpha,\delta,k)\left(\max_{\mathbf{d}_k\in\mathcal{D}}\left(\mathbf{P}^{\mathbf{d}_k}\prod_{l=1}^{k-1}\mathbf{P}^{\mathbf{d}_l}\right)\right)\mathbf{r} + \sum_{k=K+1}^{\infty}\zeta(\alpha,\delta,k)\max_{i\in\mathcal{S}}\overline{\mathbf{w}}^{(K)}(i)\mathbf{e}^T$$

and the lower bound follows from

$$\sum_{k=0}^{\infty}\zeta(\alpha,\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{r} = \sum_{k=0}^{K}\zeta(\alpha,\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{r} + \sum_{k=K+1}^{\infty}\zeta(\alpha,\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{r}$$
$$\ge \sum_{k=0}^{K}\zeta(\alpha,\delta,k)(\mathbf{P}^{\mathbf{d}})^k\mathbf{r} + \sum_{k=K+1}^{\infty}\zeta(\alpha,\delta,k)\min_{i\in\mathcal{S}}\overline{\mathbf{w}}^{(K)}(i)\mathbf{e}^T.$$

It remains to show that the infinite sum in front of the maximum (or minimum) equals the finite representation in $\varepsilon_2(\alpha,\delta,K)$.

$$\sum_{k=K+1}^{\infty}\zeta(\alpha,\delta,k) = \sum_{k=K+1}^{\infty}\frac{1}{\alpha}\left(1 - \sum_{l=0}^{k}e^{-\alpha\delta}\frac{(\alpha\delta)^l}{l!}\right)$$
$$= \frac{1}{\alpha}\sum_{k=K+1}^{\infty}\sum_{l=k+1}^{\infty}e^{-\alpha\delta}\frac{(\alpha\delta)^l}{l!} = \frac{1}{\alpha}\sum_{k=K+2}^{\infty}e^{-\alpha\delta}\frac{(\alpha\delta)^k}{k!}(k-K-1)$$
$$= \frac{1}{\alpha}\sum_{k=K+2}^{\infty}e^{-\alpha\delta}\frac{(\alpha\delta)^k}{k!}k - \frac{K+1}{\alpha}\sum_{k=K+2}^{\infty}e^{-\alpha\delta}\frac{(\alpha\delta)^k}{k!}$$
$$= \delta\sum_{k=K+1}^{\infty}e^{-\alpha\delta}\frac{(\alpha\delta)^k}{k!} - \frac{K+1}{\alpha}\sum_{k=K+2}^{\infty}e^{-\alpha\delta}\frac{(\alpha\delta)^k}{k!}$$
$$= \delta\left(1 - \sum_{k=0}^{K}\beta(\alpha\delta,k)\right) - \frac{K+1}{\alpha}\left(1 - \sum_{k=0}^{K+1}\beta(\alpha\delta,k)\right).$$

In the sequel we use $\underline{\mathbf{g}}^K_{t-\delta}$ for the vector that is computed for the decision vector resulting from the selection procedure applied to vector $\underline{\mathbf{g}}_t$. Obviously, $\underline{\mathbf{g}}_t \le \mathbf{g}^*_t$ implies $\underline{\mathbf{g}}^{\mathbf{d},K}_{t-\delta} \le \mathbf{g}^*_{t-\delta}$ for any $\mathbf{d}\in\mathcal{D}$ and $\delta \ge 0$ such that also $\underline{\mathbf{g}}^K_{t-\delta} \le \mathbf{g}^*_{t-\delta}$ holds. □

The next step is to derive an upper bound and to assure that the spread of the bounds does not become too wide which assures that the difference between the bounds and $\mathbf{g}_t{}^*$ is also small enough. The previous theorem shows that the vectors $\overline{\mathbf{v}}^{(k)}$ and $\overline{\mathbf{w}}^{(k)}$ result in an upper bound for $\mathbf{g}^{\mathbf{d}}_{t-\delta}$ resulting from a fixed decision vector in the interval $(t-\delta,t]$. The next theorem shows that it is also an upper bound for $\mathbf{g}^*_{t-\delta}$ resulting from the optimal policy.

Theorem 4. *For all* $0 \le \delta \le t$ *we have*

$$\sum_{k=0}^{\infty}(\beta(\alpha\delta,k)\overline{\mathbf{v}}^{(k)} + \zeta(\alpha,\delta,k)\overline{\mathbf{w}}^{(k)}) \ge \mathbf{g}^*_{t-\delta}.$$

Proof. According to the results in Section 2 a piecewise constant policy π^* exists that results in the optimal gain vector. Let $t-\delta = t_0 < t_1 < \cdots < t_m = t$ be the times at which the decision vector changes under π^* and let $\mathbf{d}_1,\ldots,\mathbf{d}_m$ the decision vectors such that $\mathbf{d}_i$ is the decision vector in $(t_{i-1},t_i]$. Applying uniformization, the process performs transitions according to a Poisson process with rate α. If a transition occurs at time $\tau\in(t_{i-1},t_i]$, then the transition probabilities are defined by $\mathbf{P}^{\mathbf{d}_i}$. Now consider some sample path in $(t-\delta,t]$ with L transitions that occur at times $\tau_l\in(t-\delta,t]$ ($\tau_{l+1}<\tau_l$ for $l=1,\ldots,L-1$) and let $\mathbf{c}_l$ be the decision vector of the l-th transition, i.e., $\mathbf{c}_l = \mathbf{d}_i$ iff $\tau_l\in(t_{i-1},t_i]$. Thus, the gain of the sample path equals

$$\left(\prod_{l=1}^{L}\mathbf{P}^{\mathbf{c}_l}\right)\mathbf{g}^*_t + \sum_{l=1}^{L+1}(\tau_{l-1}-\tau_l)\left(\prod_{m=1}^{l-1}\mathbf{P}^{\mathbf{c}_{l-m}}\right)\mathbf{r} \le \overline{\mathbf{v}}^{(L)} + \sum_{l=1}^{L+1}(\tau_{l-1}-\tau_l)\overline{\mathbf{w}}^{(l-1)},$$

where $\tau_{L+1} = t-\delta$ and $\tau_0 = t$. Since the transitions and therefore the whole sample path are defined by a Poisson process with rate α that is independent of the discrete time process, the distributions of sample paths are the same in the original process and in the process using the vectors $\overline{\mathbf{v}}$ and $\overline{\mathbf{w}}$. □

The theorem includes bounds resulting from infinite sums. However, the sums can be truncated yielding the following bounds:

$$\sum_{k=0}^{K}\beta(\alpha\delta,k)\overline{\mathbf{v}}^{(k)} + \overline{\gamma}^K_{t,\delta} \ge \sum_{k=0}^{\infty}\beta(\alpha\delta,k)\overline{\mathbf{v}}^{(k)}$$
and
$$\sum_{k=0}^{K}\zeta(\alpha,\delta,k)\overline{\mathbf{w}}^{(k)} + \overline{\eta}^K_{\delta} \ge \sum_{k=0}^{\infty}\zeta(\alpha,\delta,k)\overline{\mathbf{w}}^{(k)}. \tag{15}$$

These bounds can be computed with a finite number of steps and hold for every $\delta\ge 0$. If the vectors $\underline{\mathbf{v}}^{(1)},\ldots,\underline{\mathbf{v}}^{(K)}$, $\overline{\mathbf{v}}^{(1)},\ldots,\overline{\mathbf{v}}^{(K)}, \underline{\mathbf{w}}^{(1)},\ldots,\underline{\mathbf{w}}^{(K)}$, $\overline{\mathbf{w}}^{(1)},\ldots,\overline{\mathbf{w}}^{(K)}$ are precomputed, then bounds for $\underline{\mathbf{g}}_{t-\delta}$ can be computed from known bounds $\underline{\mathbf{g}}_t$ and $\overline{\mathbf{g}}_t$ with an effort in $O(nK)$.

To reach a final error bound ε (i.e., $\|\overline{\mathbf{g}}_0 - \underline{\mathbf{g}}_0\|_\infty \le \varepsilon$), the error at time t should be in range of $\varepsilon(T-t)/T$. Assume that $\|\overline{\mathbf{g}}_t - \underline{\mathbf{g}}_t\|_\infty \le \varepsilon(T-t)/T$. Then δ should be chosen such that the $\|\overline{\mathbf{g}}_{t-\delta} - \underline{\mathbf{g}}_{t-\delta}\|_\infty \le \varepsilon(T-t+\delta)/T$. Let $\varepsilon_t = \|\overline{\mathbf{g}}_t - \underline{\mathbf{g}}_t\|_\infty$ and $\varepsilon_{t-\delta} = \|\overline{\mathbf{g}}_{t-\delta} - \underline{\mathbf{g}}_{t-\delta}\|_\infty$. An appropriate δ can be determined by a simple line search.

We show that an appropriate value for δ exists such that for the final error $\varepsilon_0 \le \varepsilon$ holds. We consider first the relation between the spread of the bounds $\varepsilon_{t-\delta}$ and δ. $\varepsilon_{t-\delta}$ consists of three components

$$\varepsilon_{t-\delta} = \varepsilon_{trunc}(t,\delta,K) + \varepsilon_{acc}(t,\delta,K) + \varepsilon_{succ}(t,\delta,K), \tag{16}$$

where ε_{trunc} describes the error due to the truncation of the Poisson probabilities, ε_{acc} is the error due to the reward accumulated in $(t-\delta,t]$ and ε_{succ} is the error due to the computation of the vector at

time t which determines the reward gained in $(t,T]$. All error components consider up to K transitions in the interval:

$$\varepsilon_{trunc}(t,\delta,K) = \overline{\gamma}^K_{t,\delta} - \underline{\gamma}^K_{t,\delta} + \overline{\eta}^K_\delta - \underline{\eta}^K_\delta \in O(\delta^{K+1}) \tag{17}$$

which implies that by choosing K large enough, ε_{trunc} can be made much smaller than the other two components. Now assume that $K \geq 1$:

$$\begin{aligned}\varepsilon_{acc}(t,\delta,K) &= \left\| \sum_{k=0}^{K} \left(\overline{\mathbf{w}}^{(k)} - \left(\prod_{l=1}^{k} \mathbf{P}^{\mathbf{d}_l^*} \right) \mathbf{r} \right) \sum_{l=k}^{K} e^{-\alpha\delta} \frac{(\alpha\delta)^l \delta}{(l+1)!} \right\|_\infty \\ &= \frac{e^{-\alpha\delta}}{\alpha} \left\| \sum_{k=1}^{K} \left(\left(\prod_{j=1}^{k} \mathbf{P}^{\mathbf{d}_j} \right) - \left(\prod_{l=1}^{k} \mathbf{P}^{\mathbf{d}_l^*} \right) \right) \mathbf{r} \sum_{l=k+1}^{K} \frac{(\alpha\delta)^l}{l!} \right\|_\infty \in O(\delta^2),\end{aligned} \tag{18}$$

where $\mathbf{d}_l^*$ is the decision vector chosen by the selection procedure at the time of the lth transition in the interval $[t-\delta,t]$ and $\mathbf{d}_k$ are the decision vectors used for the computation of $\overline{\mathbf{w}}^{(k)}$. The third part of the error results from the difference between the vectors $\overline{\mathbf{v}}^{(k)}$ and $(\prod_{l=1}^{k} \mathbf{P}^{\mathbf{d}_l^*})\underline{\mathbf{g}}_t$. Since $\|\overline{\mathbf{v}}^{(0)} - \mathbf{g}_t^*\|_\infty \in O(\varepsilon_t)$

$$\varepsilon_{succ}(t,\delta,K) = \left\| \sum_{k=0}^{K} \left(\overline{\mathbf{v}}^{(k)} - \left(\prod_{l=1}^{k} \mathbf{P}^{\mathbf{d}_l^*} \right) \overline{\mathbf{g}}_t \right) e^{-\alpha\delta} \frac{(\alpha\delta)^k}{k!} \right\|_\infty \in O((1+\delta)\varepsilon_t). \tag{19}$$

Observe that the order of ε_{acc} and ε_{succ} is independent of δ and $K\,(\geq 1)$ whereas the concrete values increase with an increasing value of K. On the other hand ε_{trunc} becomes smaller for a larger value of K. For a fixed δ the overall error is usually reduced (it cannot be increased) by increasing K but will, of course, usually not shrink to zero for some fixed $\delta > 0$.

Now consider the order of the error after L time steps of length δ each. We begin with $\overline{\mathbf{g}}_T = \underline{\mathbf{g}}_T = \mathbf{g}_T = \mathbf{0}$ such that $\varepsilon_T = 0$. After a time step of length δ, which is small enough such that only the highest order of the error term counts and $K > 1$, we obtain $\varepsilon_{T-\delta} \in O(\delta^2)$ and after the next time step $\varepsilon_{T-2\delta} \in O((1-\delta)\delta^2) + O(\delta^2) = O(\delta^2)$. Consequently, the local error is in $O(\delta^2)$. Since the number of steps is in $O(1/\delta)$, the global error is in $O(\delta)$. Consequently, for every $\varepsilon > 0$ appropriate values of $\delta > 0$ can be found such that $\varepsilon_0 \leq \varepsilon$. Of course, this a theoretical result which neglects the finite arithmetic of computers and runtime restrictions.

For an improvement of the upper bound for a fixed $\delta > 0$ define the following decision vectors:

$$\begin{aligned}&\mathbf{d}_k'' = \operatorname*{argmax}_{\mathbf{d} \in \mathcal{D}} (\mathbf{P}^{\mathbf{d}} (\beta(\alpha\delta,k)\overline{\mathbf{x}}^{(k-1)} + \zeta(\alpha,\delta,k)\overline{\mathbf{y}}^{(k-1)})) \\ &\text{where } \overline{\mathbf{x}}^{(0)} = \overline{\mathbf{g}}_t, \quad \overline{\mathbf{x}}^{(k)} = \mathbf{P}^{\mathbf{d}_k''} \overline{\mathbf{x}}^{(k-1)}, \\ &\overline{\mathbf{y}}^{(0)} = \mathbf{r} \quad \text{and} \quad \overline{\mathbf{y}}^{(k)} = \mathbf{P}^{\mathbf{d}_k''} \overline{\mathbf{y}}^{(k-1)} \quad (k \geq 1)\end{aligned} \tag{20}$$

Theorem 5. *The relation*

$$\sum_{k=0}^{\infty} (\beta(\alpha\delta,k)\overline{\mathbf{v}}^{(k)} + \zeta(\alpha,\delta,k)\overline{\mathbf{w}}^{(k)}) \geq \sum_{k=0}^{\infty} (\beta(\alpha\delta,k)\overline{\mathbf{x}}^{(k)} + \zeta(\alpha,\delta,k)\overline{\mathbf{y}}^{(k)}) \geq \mathbf{g}_{t-\delta}^*$$

holds for an arbitrary but fixed $\delta > 0$.

Proof. For the proof we can assume that $\overline{\mathbf{g}}_t \geq \mathbf{g}_t^*$ is known and is used to compute the vector $\overline{\mathbf{x}}^{(k)}$.

To prove the first inequality observe that $\overline{\mathbf{v}}^{(0)} = \overline{\mathbf{x}}^{(0)}$ and $\overline{\mathbf{w}}^{(0)} = \overline{\mathbf{y}}^{(0)}$. Now we can show by induction

$$\overline{\mathbf{v}}^{(k)} = \max_{\mathbf{d} \in \mathcal{D}} (\mathbf{P}^{\mathbf{d}} \overline{\mathbf{v}}^{(k-1)}) \geq \mathbf{P}^{\mathbf{d}_k''} \overline{\mathbf{x}}^{(k-1)} = \overline{\mathbf{x}}^{(k)}$$

and

$$\overline{\mathbf{w}}^{(k)} = \max_{\mathbf{d} \in \mathcal{D}} (\mathbf{P}^{\mathbf{d}} \overline{\mathbf{w}}^{(k-1)}) \geq \mathbf{P}^{\mathbf{d}_k''} \overline{\mathbf{y}}^{(k-1)} = \overline{\mathbf{y}}^{(k)}.$$

For the second inequality, we have

$$\begin{aligned}\mathbf{g}_{t-\delta}^* &= \sum_{k=0}^{\infty} \left(\prod_{l=1}^{k} \mathbf{P}^{\mathbf{d}_l^*} \right) (\beta(\alpha\delta,k)\mathbf{g}_t^* + \zeta(\alpha,\delta,k)\mathbf{r}) \\ &\leq \sum_{k=0}^{\infty} \left(\prod_{l=1}^{k} \mathbf{P}^{\mathbf{d}_l''} \right) (\beta(\alpha\delta,k)\overline{\mathbf{g}}_t + \zeta(\alpha,\delta,k)\mathbf{r}). \qquad \square\end{aligned}$$

A finite upper bound is given by

$$\begin{aligned}&\sum_{k=1}^{K} (\beta(\alpha\delta,k)\overline{\mathbf{x}}^{(k)} + \zeta(\alpha,\delta,k)\overline{\mathbf{y}}^{(k)}) + \overline{\gamma}^K_{t-\delta} + \overline{\eta}^K_{t-\delta} \\ &\quad \geq \sum_{k=1}^{\infty} (\beta(\alpha\delta,k)\overline{\mathbf{x}}^{(k)} + \zeta(\alpha,\delta,k)\overline{\mathbf{y}}^{(k)}).\end{aligned} \tag{21}$$

The effort for the computation of the improved upper bound is $O(Knc(\sum_{i=1}^{n} m_i))$ for each new δ.

With the previous steps we can define an algorithm for computing an $\varepsilon-$optimal policy. For a practical realization of the algorithm it is necessary to consider the finite precision of the arithmetic and runtime restrictions. Thus, we introduce a minimal step width $\delta_{\min}$. This modification assures that the number of steps is not larger than $T/\delta_{\min}$, the number of iterations is bounded by $KT/\delta_{\min}$ and the computed values that are added to the gain vector do not become too small which improves the stability and efficiency of the algorithm. On the other hand the minimal time step may hinder the algorithm to compute the solution with the required precision. However, our experience shows that a small value in the range of 10^{-8} for $\delta_{\min}$ has no influence on the final result, as long as ε is not extremely small.

1. Set $i=1$, $t = T$ and $\underline{\mathbf{g}}_T = \overline{\mathbf{g}}_T = \mathbf{0}$.
2. Select $\mathbf{d}_t$ from $\mathcal{F}_{n+1}(\mathbf{g}_t)$ as described.
   If $t = T$ then $\mathbf{c}_0 = \mathbf{d}_t$.
3. Compute $\underline{\mathbf{v}}^{(1)},\ldots,\underline{\mathbf{v}}^{(K)}$ and $\underline{\mathbf{w}}^{(1)},\ldots,\underline{\mathbf{w}}^{(K)}$ (using (13)).
4. Compute $\overline{\mathbf{v}}^{(1)},\ldots,\overline{\mathbf{v}}^{(K)}$ and $\overline{\mathbf{w}}^{(1)},\ldots,\overline{\mathbf{w}}^{(K)}$ (using (14)).
5. Compute $\delta \in [\delta_{\min},t]$, $\underline{\mathbf{g}}_{t-\delta}$ from $\underline{\mathbf{v}}^{(k)}$, $\mathbf{w}^{(k)}$ (using Theorem 3) and $\overline{\mathbf{g}}_{t-\delta}$ from $\overline{\mathbf{v}}^{(k)}$, $\overline{\mathbf{w}}^{(k)}$ (using Theorem 4) such that $\|\overline{\mathbf{g}}_{t-\delta} - \underline{\mathbf{g}}_{t-\delta}\|_\infty \leq \varepsilon(T-t+\delta)/T$ if this is not possible set $\delta = \delta_{\min}$.
6. Compute $\overline{\mathbf{x}}^{(1)},\ldots,\overline{\mathbf{x}}^{(K)},\overline{\mathbf{y}}^{(1)},\ldots,\overline{\mathbf{y}}^{(K)}$ (using (20)) and improve $\overline{\mathbf{g}}_{t-\delta}$ (using Theorem 5).
7. if $\mathbf{d}_t \neq \mathbf{c}_{i-1}$ then
   $\mathbf{c}_i = \mathbf{d}_t$, $t_i = t-\delta$ and $i = i + 1$.
8. If $t = \delta$ then terminate else go to 2. with $t = t-\delta$.

The algorithm stores the optimal decision vectors in $\mathbf{c}_i$ and the times when the policy changes in t_i where index i is used in reverse order. Let L be the number of intervals that have been computed in the algorithm, then define $t_0 = T$ and $t_{L+1} = 0$. The computed ε-optimal policy uses decision vector $\mathbf{c}_i$ in $(t_i,t_{i-1}]$. Step 6 of the algorithm is optional.

We briefly analyze the effort of the algorithm. Step 2 requires an effort in $O(nc(\sum_{i=1}^{n} m_i))$ and step 3 requires an effort in $O(Knc)$. Computation of the vectors in step 4 requires an effort in $O(Knc(\sum_{i=1}^{n} m_i))$. The finding of δ requires an effort in $O(MKn)$ where M equals the number of function evaluations to find δ, usually $M < n$ can be assumed. Computation of the improved

upper bound requires again an effort in $O(Knc(\sum_{i=1}^{n} m_i))$ such that the overall effort per time step is $O(Knc(\sum_{i=1}^{n} m_i))$. Compared to the discretization approach the algorithm is more efficient if for the average time step $\delta > Kh$ holds.

It is possible to choose K adaptively by defining the fraction of the error resulting from the truncation of the Poisson probabilities. Let $\omega \in (0,1)$ be this fraction. The optimal value for ω is model dependent, we use $\omega \in [10^{-6},10^{-2}]$ in our examples. Then the lines 3–5 in the algorithm are substituted by the following code.

1. Initialize $K=1$, *stop* = *false* and the vectors $\underline{\mathbf{v}}^{(0)}$, $\overline{\mathbf{v}}^{(0)}$, $\underline{\mathbf{w}}^{(0)}$, $\overline{\mathbf{w}}^{(0)}$;
2. repeat
3. compute $\underline{\mathbf{v}}^{(K)}$, $\overline{\mathbf{v}}^{(K)}$, $\underline{\mathbf{w}}^{(K)}$, $\overline{\mathbf{w}}^{(K)}$ (using (13) and (14));
4. find $\delta = \max_{\delta' \in [0,t]}(\varepsilon_{trunc}(t,\delta',K) \le \frac{\omega\delta'}{T}\varepsilon)$;
5. $\delta = \max(\delta, \delta_{\min})$;
6. compute $\beta(\alpha\delta,k)$ and $\zeta(\alpha,\delta,k)$ for $k=0,\ldots,K$;
7. compute $\varepsilon_{acc}(t,\delta,K)$ and $\varepsilon_{succ}(t,\delta,K)$;
8. if $\varepsilon_{trunc}(t,\delta,K)+\varepsilon_{acc}(t,\delta,K)+\varepsilon_{succ}(t,\delta,K) > \frac{T-t+\delta'}{T}\varepsilon$ then
9. reduce δ until
   $\varepsilon_{trunc}(t,\delta,K)+\varepsilon_{acc}(t,\delta,K)+\varepsilon_{succ}(t,\delta,K) \le \frac{T-t+\delta}{T}\varepsilon$
   or $\delta = \delta_{\min}$ and set $stop = true$;
10. else
11. $K = K + 1$;
12. until *stop* or $K = K_{\max}+1$;

As long as the truncation error is too large (i.e. more than $\omega\delta\varepsilon/T$), the number of iterations is increased which reduces the truncation error and increases the other two error components. Iterations are added until a maximum number of iterations has been performed or the relation between the error components is as required. The adaptive computation assures that small differences in the vectors $\underline{\mathbf{v}}^{(k)}$, $\overline{\mathbf{v}}^{(k)}$ and $\underline{\mathbf{w}}^{(k)}$, $\overline{\mathbf{w}}^{(k)}$ result in many iterations and larger time steps, whereas for large differences only a few iterations (often only one iteration) are performed and the time step becomes small. Large differences in the vectors $\underline{\mathbf{v}}^{(k)}$, $\overline{\mathbf{v}}^{(k)}$ and $\underline{\mathbf{w}}^{(k)}$, $\overline{\mathbf{w}}^{(k)}$ at iteration k indicate that after k transitions the policy changes and the time step has to be small enough to assure that the probability of k or more transitions in $(t-\delta,t]$ is small. The values of $K_{\max}$ determine the memory requirements to store intermediate vectors. Observe that the restriction of the number of iterations due to $K_{\max}$ becomes only relevant for large values of δ and does not restrict the approach to compute the result with a predefined accuracy.

5. Examples

The algorithm has been implemented in C++ as part of our numerical library for Markov models. We present here the analysis of three example models to show the efficiency and result quality of the uniformization approach compared to the discretization approach. In all example runs we choose $\delta_{\min} = 10^{-8}$ and $K_{\max}=20$. Furthermore, we compute K adaptively. The values for $\delta_{\min}$ and $K_{\max}$ are chosen such that they do not influence the behavior of the uniformization algorithm, i.e., the step length δ that is computed is always larger than $\delta_{\min}$ and the iteration stops almost always with $K < K_{\max}$.

5.1. A simple example

We first present a very simple example that describes the maintenance process of a system. The CTMDP has only five states and is shown in Fig. 1. It describes a simple system which can be in one of the three operational states numbered 1 through 3 that describe observable operational levels. In these states, the reward is 1 and the sojourn time is exponentially distributed with rate 1. After leaving the last operational state (state 3) through the transition described by the solid arc, the system goes into a failure state (state 5) where the reward is 0. A repair which brings the system back to the first operational state requires an exponentially distributed repair time with rate 0.01. In the second and third operational state, it may be decided to perform a maintenance operation which implies that the system is shut down which requires an exponentially distributed time with rate 10. Of course, during the system is shut down, the operational state may degrade or the system may fail. After a successful shut down, the system is in the maintenance state (state 4) with reward 0. Maintenance requires an exponentially distributed duration with rate 0.2 and brings the system back to the first operational state. Transitions to perform maintenance are described by dashed arcs. $\mathcal{D}_i$ contains one decision for the states 1, 4 and 5 and two decisions, namely maintenance or no maintenance, in the states 2 and 3.

The behavior of the optimal policy is as follows: For a long time horizon, the best decision is to initiate maintenance in the states 2 and 3. If the horizon becomes smaller, then first maintenance is no longer initiated in state 2 and finally, it is also switched off in state 3.

We analyze the example in the interval [0,100] and assume that it is initially in the first state. Table 1 contains the times when the decision vector changes. Initially maintenance is initiated in the states 2 and 3. After the first switch, maintenance is no longer initiated in state 2 and after the second switch, maintenance is no longer initiated in state 3. The table includes results for uniformization and discretization. The number of iterations in uniformization is set adaptively according to the value of ε. If uniformization performs i iterations, then h is set to $100/i$ such that discretization performs the same number of iterations. Observe that uniformization requires the number of iterations twice, namely to compute the lower and to compute the upper bound. For $\varepsilon = 1.0$ uniformization requires only 450 iterations which would result in $h=0.22$ which is a too large time step since matrix $\mathbf{P}_h^{\mathbf{d}}$ is not stochastic in this case. It can be seen that uniformization requires less iterations to obtain switching times up to some fixed accuracy, although the difference is fairly small.

Table 2 shows the bounds for the reward gained by an optimal policy computed with uniformization and the approximation computed with the discretization approach. It can be seen that the upper bound is fairly conservative whereas the lower bound contains the exact reward even for $\varepsilon = 1.0$. The result of the discretization approach is always between the two bounds.

5.2. A multiprocessor system

The second example describes a fault-tolerant multiprocessor. The system (see Fig. 2) consists of four processors, three memories

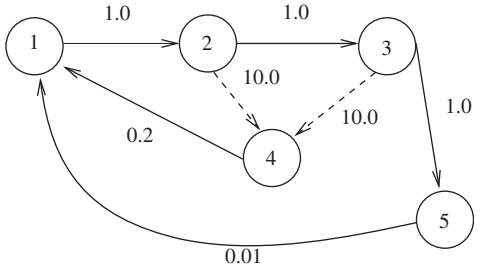

Fig. 1. CTMDP of the running example.

Table 1
Times when the decision vector changes for the small example.

Uniformization				Discretization	
ε	Iter.	Switching times		Switching times	
1.0e+0	450	29.2460	95.88094	–	–
5.5e−1	1120	29.4645	95.88114	29.5455	95.81818
1.0e−1	6524	29.4939	95.88292	29.5065	95.87677
1.0e−2	43,894	29.4941	95.88338	29.4961	95.88326
1.0e−3	390,563	29.4942	95.88344	29.4944	95.88338
1.0e−4	3,797,573	29.4942	95.88344	29.4943	95.88342
7.0e−5	5,414,805	29.4942	95.88344	29.4942	95.88344

Table 2
Reward bounds and approximate rewards for the small example.

Uniformization				Discretization
ε	Iter.	Bounds for G^*		Approximation for G^*
1.0e+0	450	20.9308	21.8906	–
5.5e−1	1120	20.9308	21.4714	20.9355
1.0e−1	6524	20.9308	21.0278	20.9344
1.0e−2	43,894	20.9308	20.9381	20.9313
1.0e−3	390,563	20.9308	20.9314	20.9309
1.0e−4	3,797,573	20.9308	20.9309	20.9308
7.0e−5	5,414,805	20.9308	20.9308	20.9308

and two buses. All components may fail and are repaired by a single repair unit with a preemptive repair priority for the different components. The times to failure and repair times are exponentially distributed with the rates shown in Table 3. We consider availability and performability analysis of the system and begin with availability analysis.

The system is available, if at least one component of each type is working. For fixed repair priorities the system can be modeled as a Markov process with 60 states. To find the state dependent repair priority that maximizes availability, we model the system as a CTMDP. In every state where components of more than one type are down, it is possible to choose the component to be repaired first. We have 26 states where two types of component are down and 24 states where components of all three types are down. This implies that size of $\mathcal{D}$ equals $2^{26}\cdot 3^{24} = 282{,}496{,}645{,}345$. However, $\sum m_i$ is only 134 which about twice the number of states.

We analyze the availability under an optimal policy in the interval [0,100] (see Fig. 3). Results are computed for $\varepsilon = 10^{-6}$ such that differences between upper and lower bounds are not visible. The availability under the optimal policy converges towards 0.9953 for $T\to\infty$ compared to 0.9943 which is the steady state availability under the best static repair priorities where repair of memories has the highest priority, buses have the second priority and processors are repaired last. For finite horizons, the optimal policy changes five times, namely at $T-5.478\text{e}-4$, at $T-1.558\text{e}-2$, at $T-0.6818$, at $T-1.9677$ and at $T-1.9878$. For the lack of space, we do not present the complete decision vector here. However, the tendency is that towards the end of the mission, repair of processors gets a higher priority. The reason is that processors have the smallest repair times such that it is more likely to repair a processor in a small time slot, than a memory or a bus.

Table 4 shows the number of iterations and the reward bounds for the multiprocessor example. Like for the previous example the discretization step is set in such a way that discretization performs the same number of iterations than uniformization to compare the results. For $\varepsilon = 0.001$ uniformization computes the result with six significant digits and requires only 293 iterations. For the discretization parameter $h < 0.2407$ is required to obtain a stochastic matrices $\mathbf{P}_h^{\mathbf{d}}$ such that at least $100\cdot h^{-1} = 416$ iterations have to be performed. However, even with 169,380 iterations discretization reaches only five significant digits of accuracy and the results lies outside of the bounds computed by uniformization with only 56 iterations. Observe that G describes the accumulated reward which is the availability multiplied with the length of the interval. The columns with the headline *Sw. time* contains the times when the stationary policy switches for the first time due to the finite horizon. Again it can be noticed that the results of uniformization are more accurate than those of the discretization approach.

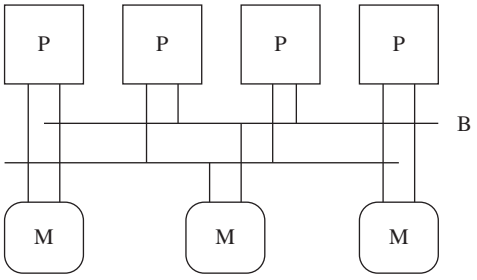

Fig. 2. Fault tolerant multiprocessor example.

Table 3
Failure and repair rates for the second example.

	Failure	Repair
Bus	0.01	0.2
Mem.	0.025	1.0
Proc.	0.02	4.0

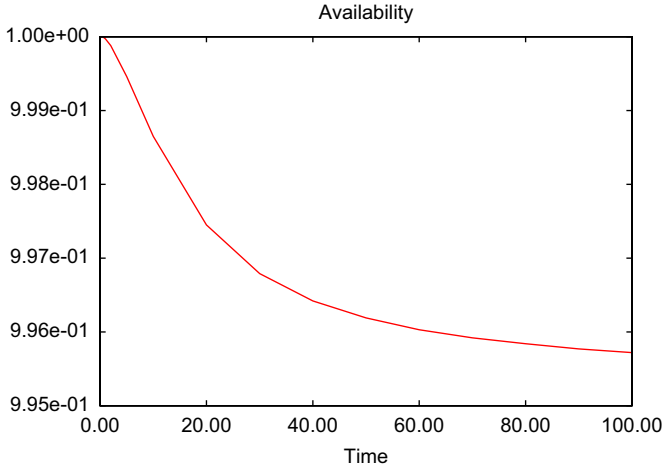

Fig. 3. Time dependent availability under the optimal policy.

We now consider the same model but with a different reward vector. Let $\#P$, $\#M$ and $\#B$ be the number of working processors, memories and buses, respectively. Then the reward in state ($\#P$, $\#M$, $\#B$) equals

$$Ind(\#P\cdot\#M\cdot\#B>0)\cdot 1.2^{\#P}\cdot 1.8^{\#M}\cdot 1.8^{\#B}$$

where $Ind(a>b)$ is the indicator function which is 1 for $a>b$ and 0 otherwise. The reward can now be interpreted as a performability measure, e.g., the throughput under failures and repairs.

Table 4
Reward bounds and approximate rewards for the availability of the multiprocessor.

Uniformization					Discretization	
ε	Iter.	Bounds for G^*		Sw. time	App. G^*	Sw. time
1.0e+0	46	99.4759	99.5730	96.6678	–	–
1.0e−1	56	99.5623	99.5723	96.1696	–	–
1.0e−2	92	99.5713	99.5722	97.9529	–	–
1.0e−3	293	99.5721	99.5721	97.9905	–	–
1.0e−4	851	99.5721	99.5721	98.0080	99.6816	98.1200
1.0e−5	2946	99.5721	99.5721	98.0113	99.6049	98.0133
1.0e−6	8235	99.5721	99.5721	98.0122	99.5818	98.0207
1.0e−7	32,874	99.5721	99.5721	98.0122	99.5751	98.0136
1.0e−8	169,380	99.5721	99.5721	98.0122	99.5727	98.0128

Table 5
Reward bounds and approximate rewards for the performability of the multiprocessor.

Uniformization					Discretization	
ε	Iter.	Bounds for G^*		Sw. time	Approx. G^*	Sw. time
1.0e+1	54	3596.02	3597.01	90.9710	–	–
1.0e+0	64	3596.89	3596.99	90.9652	–	–
1.0e−1	79	3596.98	3596.99	90.1525	–	–
1.0e−2	149	3596.99	3596.99	90.8298	–	–
1.0e−3	388	3596.99	3596.99	91.0748	–	–
1.0e−4	1044	3596.99	3596.99	91.0927	3600.42	91.0920
1.0e−5	3549	3596.99	3596.99	91.0948	3598.00	91.0961
1.0e−6	18,666	3596.99	3596.99	91.0950	3597.18	91.0961
1.0e−7	96,065	3596.99	3596.99	91.0950	3597.02	91.0956

Again a CTMDP can be used to find a repair strategy that optimizes the performability. The optimal policy switches at $T-0.0012$, $T-6.009$ and $T-8.905$. The policy for performability optimization differs from the optimal policy for availability. For performability there is a tendency to do the repair of memories or buses before the repair of processors since processors have the smallest coefficient in the formula to compute the performability. However, if the time approaches T, then again processors are prioritized because of their smaller repair times. Table 5 contains the performability results. Again, uniformization computes very tight bounds with very few iterations whereas discretization requires much more iterations to reach a similar accuracy.

5.3. A queueing network

The next example is a simple queueing network consisting of two finite capacity queues with capacity 10 each and an exponentially distributed service time with mean 1.0 and 2.0. The queues are fed by a Poisson process with rate 1.0. For each arrival it has to be decided whether to put it into the buffer of the first or the second queue. After a packet has been put into a buffer it has to remain there until it is served. The goal is to find a policy maximizes the throughput over a finite interval $[0,T]$ when the system is started with empty buffers at time 0. The resulting Markov chain contains 121 states and in each state with buffer space in both queues, it can be decided where to put the arriving customer.

The policy changes several times at the end of the interval, at time $T-31.69$ the last change takes place and the stationary policy has been reached. Fig. 4 shows the average throughput in the interval $[0,T]$ for $0 \le T \le 100$. Bounds are computed with a spread of 0.0001 such that the difference between upper and lower bounds is almost not visible. The effort of the uniformization and the discretization approach for different numbers of iterations is shown in Table 6. For this example uniformization and discretization produce similar results for a fixed number of iterations. Like for the simple example the upper bound computed with uniformization is rather conservative such that the true result is near to the lower bound. As in both other examples the switching times are better approximated by uniformization than with the discretization approach after a fixed number of iterations.

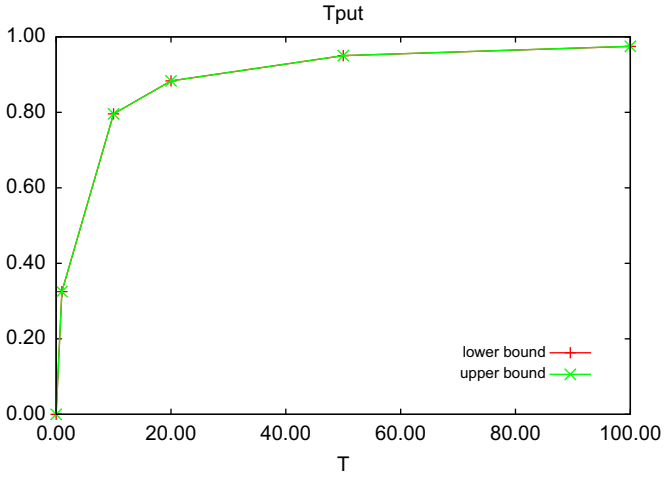

Fig. 4. Average throughput of the queueing system in the interval $[0,T]$ under an optimal policy.

Table 6
Reward bounds and approximate rewards for the queueing example.

Uniformization					Discretization	
ε	Iter.	Bounds for G^*		Sw. time	Approx. G^*	Sw. time
1.0e+0	1615	97.4880	98.4822	68.2561	97.5501	68.5449
1.0e−1	15,184	97.4881	97.5877	68.3037	97.5073	68.3642
1.0e−2	146,937	97.4881	97.4981	68.3099	97.4888	68.3123
1.0e−3	1,467,520	97.4881	97.4891	68.3102	97.4881	68.3104
1.0e−4	14,677,467	97.4881	97.4882	68.3102	97.4881	68.3101

6. Conclusions

In this paper we present a new approach to compute an optimal policy and the resulting reward for continuous time Markov decision processes over finite time horizons. In contrast to the available discretization approach, uniformization allows one to compute error bounds on the reward that can be gained by an optimal policy and it determines the time steps adaptively according to the spread of the bounds. Results of some examples show that tight reward bounds can be computed and after a fixed number of iterations the lower bound computed with uniformization is usually a better approximation of the reward than the result of the discretization approach computed with the same number of iterations.

References

[1] Bertsekas DP. Dynamic programming and optimal control, vol. I. Athena Scientific; 2005.

[2] Bertsekas DP. Dynamic programming and optimal control, vol. II. Athena Scientific; 2007.
[3] Hu Q, Yue W. Markov decision processes with their applications. Advances in mechanics and mathematics. Springer; 2008.
[4] Puterman ML. Markov decision processes. Wiley; 2005.
[5] Gross D, Miller D. The randomization technique as a modeling tool and solution procedure for transient Markov processes. Operations Research 1984;32(2):926–44.
[6] Jensen A. Markoff chains as an aid in the study of Markoff processes. Skandinavisk Aktuarietidskritft 1953;36:87–91.
[7] Beutler FJ, Ross KW. Uniformization for semi-Markov decision processes under stationary policies. Journal of Applied Probability 1987;24:644–56.
[8] Serfozo RF. An equivalence between continuous and discrete time Markov decision processes. Operations Research 1979;27(3):616–20.
[9] Miller BL. Finite state continuous time Markov decision processes with a finite planning horizon. SIAM Journal on Control 1968;6(2):266–80.
[10] Yasuda M. On the existence of optimal control in continuous time Markov decision processes. Bulletin of Mathematical Statistics 1972;15:7–17.
[11] Lembersky MR. On maximal rewards and ε-optimal policies in continuous time Markov decision chains. The Annals of Statistics 1974;2(4):159–69.
[12] Martin-Löfs A. Optimal control of a continuous-time Markov chain with periodic transition probabilities. Operations Research 1967;15:872–81.
[13] Lippman SA. Applying a new device in the optimization of exponential queuing systems. Operations Research 1975;23(4):687–710.
[14] Lippman SA. Countable-state, continuous-time dynamic programming with structure. Operations Research 1976;24(3):477–90.
[15] Bhatnagar S, Abdulla MS. Simulation-based optimization algorithms for finite-horizon Markov decision processes. Simulation 2008;84(12):577–600.
[16] Stewart WJ. Introduction to the numerical solution of Markov chains. Princeton University Press; 1994.
[17] Rindos A, Woolet S, Viniotis I, Trivedi K. Exact methods for the transient analysis of nonhomogeneous continuous time Markov chains. In: Stewart WJ, editor. Computations with Markov chains. Kluwer Academic Publishers; 1995. p. 121–33.
[18] Arns M, Buchholz P, Panchenko A. On the numerical analysis of inhomogeneous continuous time Markov chains. INFORMS Journal on Computing 2010;22(3):416–32.
[19] Baier C, Hermanns H, Katoen JP, Haverkort BR. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. Theoretical Computer Science 2005;345(1):2–26.