Available online at www.sciencedirect.com
SCIENCE DIRECT®

Theoretical Computer Science 345 (2005) 2–26

Theoretical Computer Science

www.elsevier.com/locate/tcs

Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes ☆

Christel Baier[a], Holger Hermanns[b, c, *], Joost-Pieter Katoen[c], Boudewijn R. Haverkort[c]

[a] University of Bonn, Römerstraße 164, D-53117 Bonn, Germany
[b] Saarland University, Im Stadtwald, D-66123 Saarbrücken, Germany
[c] University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

A continuous-time Markov decision process (CTMDP) is a generalization of a continuous-time Markov chain in which both probabilistic and nondeterministic choices co-exist. This paper presents an efficient algorithm to compute the maximum (or minimum) probability to reach a set of goal states within a given time bound in a uniform CTMDP, i.e., a CTMDP in which the delay time distribution per state visit is the same for all states. It furthermore proves that these probabilities coincide for (time-abstract) history-dependent and Markovian schedulers that resolve nondeterminism either deterministically or in a randomized way.

Keywords: Continuous-time; Markov decision process; Temporal logic; Model checking; Time-bounded reachability

1. Introduction

A continuous-time Markov decision process (CTMDP) [10,20, 31,34] is a generalization of a continuous-time Markov chain (CTMC) in which both probabilistic and

☆ This work is supported by the NWO-DFG bilateral project *Validation of Stochastic Systems*.

* Corresponding author. Saarland University, Department of Computer Science, IM Stadtwald, D-66123 Saarbrücken, Germany. Tel.: +49 681 302 5631; fax: +49 681 302 5636.
E-mail address: hermanns@cs.uni-sb.de (H. Hermanns).

0304-3975/$ - see front matter © 2005 Published by Elsevier B.V.
doi:10.1016/j.tcs.2005.07.022

nondeterministic choices co-exist. CTMDPs are a natural modeling formalism applicable in many contexts, ranging from stochastic control theory [20] and scheduling [13,1] to dynamic power management [32].

Importance of CTMDPs: The class of CTMDPs is particularly interesting, because it can be viewed as a common semantic model for various performance and dependability modelling formalisms including generalized stochastic Petri nets [2], Markovian stochastic activity networks [33], and interactive Markov chains (IMC) [23]. So far, the analysis of models developed in these and related formalisms was restricted to the subset that corresponds to CTMCs, usually referred to as "non-confused", "well-defined", or "well-specified" models [16,17,19,23]. All these notions are semantic notions. They are usually checked by an exhaustive exploration of the state space associated with a given model. A model is discarded if the check fails. In other words, no specification-level check is available, and the offered analysis algorithms are actually partial algorithms.

Model checking: Model checking of CTMCs [6] has received remarkable attention in recent years. Various model checkers exist [25,27,15], answering questions such as: *Is the probability to hop along Φ-states, until reaching a Ψ-state within* 5 *to* 10 *time units greater than* 0.95? The core algorithmic innovation allowing to answer such questions is a mapping from interval-bounded until-formulae—specified in the continuous stochastic logic CSL [5]—to time-bounded reachability problems [6], which in turn can be approximated efficiently using a stable numerical technique called uniformization [26]. To enable the same kind of questions being answered for models specified in any of the above mentioned formalisms, the key problem is how to compute time-bounded reachability probabilities in CTMDPs. This is the problem we address in this paper. With the notable exception of De Alfaro [3,4], who studied long-run properties of semi-Markov decision processes, we are not aware of any model checking algorithm for CTMDPs. This stands in sharp contrast to discrete-time Markov decision processes, for which model checking algorithms are well-understood [12,9] and, for instance, implemented in tools like PRISM [30] or RAPTURE [18].

Contribution: Given a CTMDP, our aim is to compute the maximum (or minimum) probability to reach—under a given class of schedulers—a certain set of states within t time units, given a starting state. We consider this problem for uniform CTMDPs, a class of CTMDPs in which the delay time distribution per state visit is the same for all states, governed by a unique exit rate E. We show that an efficient greedy algorithm can be obtained using truncated Markovian deterministic (MD)-schedulers, that is, step-dependent schedulers which schedule up to a limited depth. The algorithm computes the maximum (or minimum) probabilities for timed reachability. It is then shown that these probabilities for truncated MD-schedulers coincide with the maximum (or minimum) probabilities for timed reachability for Markovian and history-dependent schedulers (both deterministic and randomized). We show that stationary Markovian schedulers—as opposed to the discrete case [12,9]—yield a smaller maximum, whereas timed history-dependent schedulers may yield a higher probability.

The main result of this paper is a computationally efficient approximation algorithm for computing maximum probabilities for timed reachability in uniform CTMDPs under *all* time-abstract schedulers. The time complexity is in $\mathcal{O}(t\cdot E\cdot N^2\cdot M)$ and the space complexity in $\mathcal{O}(N^2\cdot M)$ where t is the time bound, E is the uniform exit rate of the CTMDP under consideration, N is the number of states, and M the number of actions. The results in this paper are presented only for *maximum* probabilities. Unless otherwise stated, the results are straightforwardly adapted to the dual problem of *minimum* probabilities.

Organization of the paper: Section 2 introduces the necessary background. Section 3 presents an algorithm for uniform CTMDP which relies only on step-dependent truncated schedulers. Section 4 places the algorithm in the context of more general classes of schedulers. Section 5 discusses the problem of uniformizing arbitrary CTMDPs. Section 6 concludes the paper.

This paper is an extended version of the conference paper [7].

2. Preliminaries

This section sets the stage for the results presented in the sequel, by presenting the definitions and notations used throughout the paper.

2.1. Markov decision processes

Definition 1. A *continuous-time Markov decision process* (CTMDP) $\mathcal{M}$ is a tuple $(S, Act, \mathbf{R})$ with

- S, a finite set of *states*,
- Act, a finite set of *actions*, and
- $\mathbf{R} : (S \times Act \times S) \to \mathbb{R}_{\geqslant 0}$, a three-dimensional *rate matrix*.

For each state $s \in S$ we require the existence of at least one pair $(\alpha, s') \in Act \times S$ with $\mathbf{R}(s, \alpha, s') > 0$. Note that this can easily be established by adding self-loops, i.e., having $\mathbf{R}(s, \alpha, s) > 0$ for some $\alpha \in Act$.

For $B \subseteq S$, let $\mathbf{R}(s, \alpha, B)$ denote the total rate to move from state s via action α to some state in B, i.e.,

$$\mathbf{R}(s, \alpha, B) = \sum_{s' \in B} \mathbf{R}(s, \alpha, s').$$

The behavior of a CTMDP is as follows. $\mathbf{R}(s, \alpha, s') > 0$ means that there is a transition from s to s' under action α. If state s has outgoing transitions for distinct actions, one of these actions is selected nondeterministically where we assume that the nondeterminism is resolved by means of a scheduler (also called policy or adversary). Given that action α has been chosen, $1 - \mathrm{e}^{-\mathbf{R}(s,\alpha,s')\cdot t}$ is the probability that the α-transition $s \to s'$ can be triggered within t time units. Thus, the delay of α-transition $s \to s'$ is governed by the negative exponential distribution with rate $\mathbf{R}(s, \alpha, s')$. If $\mathbf{R}(s, \alpha, s') > 0$ for more than one state s', a competition between the α-transitions originating in s exists, known as the race condition.

The set of enabled actions in a state s is

$$Act(s) = \{\, \alpha \in Act \mid E(s, \alpha) > 0 \,\}$$

where $E(s, \alpha) = \mathbf{R}(s, \alpha, S)$, is the exit rate of state s via some α-transition. An alternative formulation of the above requirement that in every state at least one action is enabled, can be stated as

$$Act(s) \neq \emptyset \quad \text{for any state } s.$$

Definition 2. A *discrete-time Markov decision process* (DTMDP) $\mathcal{M}$ is a tuple $(S, Act, \mathbf{P})$ with
- S, a finite set of states,
- Act, a finite set of actions, and
- $\mathbf{P} : (S \times Act \times S) \to [0, 1]$, a three-dimensional *probability matrix* satisfying for each state and action pair (s, α) that $\sum_{s' \in S} \mathbf{P}(s, \alpha, s') \in \{0, 1\}$.

For a given CTMDP $\mathcal{M} = (S, Act, \mathbf{R})$, the discrete probability of selecting α-transition $s \to s'$ is determined by the embedded DTMDP, denoted $emb(\mathcal{M}) = (S, Act, \mathbf{P})$ with

$$\mathbf{P}(s, \alpha, s') = \begin{cases} \dfrac{\mathbf{R}(s, \alpha, s')}{E(s, \alpha)} & \text{if } E(s, \alpha) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\mathbf{P}(s, \alpha, s')$ is the time-abstract probability for the α-transition from s to s' when action α is chosen. For $B \subseteq S$ let

$$\mathbf{P}(s, \alpha, B) = \sum_{s' \in B} \mathbf{P}(s, \alpha, s')$$

denote the probability to move from s to some state in B via an α-transition.

Definition 3. A CTMDP $(S, Act, \mathbf{R})$ is *uniform* if for some $E > 0$ it holds $E(s, \alpha) = E$ for any state $s \in S$ and $\alpha \in Act(s)$.

Note that $E(s, \alpha) = 0$ (whence $\alpha \notin Act(s)$ follows) is possible in uniform CTMDPs. Stated in words, in a uniform CTMDP the exit rates for all states and all enabled actions are equal.

2.2. Paths

A (timed) *path* σ in a CTMDP $\mathcal{M}$ is a finite or infinite sequence

$$\sigma \in (S \times Act \times \mathbb{R}_{>0})^* \times S \cup (S \times Act \times \mathbb{R}_{>0})^{\omega}.$$

For infinite path $\sigma = s_0, \alpha_0, t_0, s_1, \alpha_1, t_1, s_2, \alpha_2, t_2, \ldots$ we require time-divergence, i.e., $\sum t_i = \infty$. We write

$$s_0 \xrightarrow{\alpha_0, t_0} s_1 \xrightarrow{\alpha_1, t_1} s_2 \xrightarrow{\alpha_2, t_2} \cdots$$

rather than $s_0, \alpha_0, t_0, s_1, \alpha_1, t_1, s_2, \alpha_2, t_2, \ldots$. The corresponding *time-abstract* path is: $s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \ldots$, and the corresponding *action-abstract* path is: $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \cdots$. In the remainder of this paper we use the term *path* for timed, time-abstract, action-abstract, and time- and action-abstract paths whenever the kind of path is clear from the context. Let *first*(σ) denote the state in which σ starts. For finite path σ, *last*(σ) denotes the last state of σ, and we write $\sigma \to s$ if the finite time- and action-abstract path σ is followed by state s.

2.3. Markov chains

If for a CTMDP $(S, Act, \mathbf{R})$ the set Act is a singleton, we can project $\mathbf{R}$ on an $(S \times S)$ matrix, resulting in a continuous-time Markov chain.

Definition 4. A *continuous-time Markov chain* (CTMC) $\mathcal{C}$ is a tuple $(S, \mathbf{R})$ with
- S, a finite or countable set of states,
- $\mathbf{R} : (S \times S) \to \mathbb{R}_{\geqslant 0}$, a two-dimensional rate matrix such that $\sum_{s' \in S} \mathbf{R}(s, s')$ is convergent for all states $s \in S$. [1]

A *discrete-time Mar-kov chain* (DTMC) $\mathcal{C}$ is a tuple $(S, \mathbf{P})$ with
- S, a finite or countable set of states, and
- $\mathbf{P} : (S \times S) \to [0, 1]$, a two-dimensional probability matrix satisfying for each state s that $\sum_{s'} \mathbf{P}(s, s') \in \{0, 1\}$.

A CTMC is *uniform* if for some $E > 0$ it holds $E(s) = E$ for any state $s \in S$, where $E(s) = \mathbf{R}(s, S)$. Any CTMC can be transformed into a uniform CTMC by adding self-loops [31]. For CTMC $\mathcal{C} = (S, \mathbf{R})$ let (uniformization rate) E be a real number such that $E \geqslant \max_{s \in S} E(s)$. Then, *unif*$(\mathcal{C}, E) = (S, \overline{\mathbf{R}})$ is a uniform CTMC with

$$\overline{\mathbf{R}}(s, s') = \begin{cases} \mathbf{R}(s, s) + E - E(s) & \text{if } s = s', \\ \mathbf{R}(s, s') & \text{otherwise.} \end{cases}$$

In *unif*$(\mathcal{C}, E)$ all rates of self-loops are "normalized" with respect to E, such that state transitions occur with an "average pace" of E, uniform for all states of the chain. The behaviors exhibited by $\mathcal{C}$ and *unif*$(\mathcal{C}, E)$ are almost indistinguishable, in particular timed-reachability properties are preserved. In formal terms, $\mathcal{C}$ and *unif*$(\mathcal{C}, E)$ are weakly bisimilar [8].

Probability measure. In contrast to a CTMDP (or DTMDP), a CTMC (or DTMC) is a fully determined stochastic process. For a given initial state s_0 in CTMC $\mathcal{C}$, a unique probability measure Pr on *Path*(s_0) exists, where *Path*(s_0) denotes the set of timed paths that start in s_0. Timed paths through a CTMC are defined as for CTMDPs, but by nature are action-abstract. The inductive construction of the probability measure follows [6], the fact that we allow countable-state Markov chains does not alter the construction. Let $\mathbf{P}$ be the probability matrix of the embedded DTMC of $\mathcal{C}$ and let $C(s_0 \xrightarrow{I_0} \cdots \xrightarrow{I_{k-1}} s_k)$ denote the cylinder set consisting of all timed paths σ that start in state s_0 such that s_i $(i \leqslant k)$ is the $(i+1)$th state on σ and the time spent in s_i lies in the non-empty interval I_i $(i < k)$ in $\mathbb{R}_{\geqslant 0}$.

[1] For our purposes, it suffices to require that for any state s the set $\{s' \in S : \mathbf{R}(s, s') > 0\}$ of successors of s is finite.

The cylinder sets induce the probability measure Pr on the timed paths through $\mathcal{C}$, defined by induction on k by $\Pr(C(s_0)) = 1$, and, for $k > 0$:

$$\Pr(C(s_0 \xrightarrow{I_0} \cdots \xrightarrow{I_{k-1}} s_k \xrightarrow{I'} s')) = \Pr(C(s_0 \xrightarrow{I_0} \cdots \xrightarrow{I_{k-1}} s_k)) \cdot \mathbf{P}(s_k, s') \cdot \left(\mathrm{e}^{-E(s_k)\cdot a} - \mathrm{e}^{-E(s_k)\cdot b}\right)$$

where $a = \inf\ I'$ and $b = \sup\ I'$.

2.4. Schedulers

CTMDPs incorporate nondeterministic decisions, as opposed to CTMCs. Nondeterminism in a CTMDP is resolved by a *scheduler*. For deciding which of the next nondeterministic actions to take, a scheduler may have access to the current state only or to the path from the initial to the current state (either with or without timing information). Schedulers may select the next action either (i) *deterministically*, i.e., depending on the available information, the next action is chosen in a deterministic way, or (ii) in a *randomized* fashion, i.e., depending on the available information the next action is chosen probabilistically. Accordingly, the following classes of schedulers D are distinguished [31], where *Distr*(*Act*) denotes the collection of all distributions on *Act*:

- stationary Markovian deterministic (SMD, also called simple schedulers),

  $$D : S \to Act$$

  such that

  $$D(s) \in Act(s);$$

- stationary Markovian randomized (SMR),

  $$D : S \to Distr(Act)$$

  such that

  $$D(s)(\alpha) > 0 \quad \text{implies} \ \ \alpha \in Act(s);$$

- Markovian deterministic (MD, also called step-dependent schedulers),

  $$D : S \times \mathbb{N} \to Act$$

  such that

  $$D(s, n) \in Act(s);$$

- Markovian randomized (MR),

  $$D : S \times \mathbb{N} \to Distr(Act)$$

such that

$$D(s, n)(\alpha) > 0 \quad \text{implies} \quad \alpha \in Act(s);$$

- (time-abstract) history-dependent, deterministic (HD),

$$D : (S \times Act)^* \times S \to Act$$

such that

$$D(\underbrace{s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_{n-1}}}_{\text{time-abstract history}}, s_n) \in Act(s_n);$$

- (time-abstract) history-dependent, randomized (HR),

$$D : (S \times Act)^* \times S \to Distr(Act)$$

such that

$$D(s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_{n-1}}, s_n)(\alpha) > 0 \quad \text{implies} \quad \alpha \in Act(s_n).$$

All these schedulers are time-abstract; time-dependent schedulers will be discussed in Section 4. We write X to denote the class of all X-schedulers over a fixed CTMDP $\mathcal{M}$.[2]

Note that for any HD-scheduler, the actions can be dropped from the history, i.e., HD-schedulers may be considered as functions $D : S^+ \to Act$, as for any sequence $s_0, s_1, \ldots, s_n$ the relevant actions α_i are given by $\alpha_i = D(s_0, s_1, \ldots, s_i)$, and, hence, the scheduled action sequence can be constructed from prefixes of the path at hand. Thus, any state-action sequence $s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_{n-1}} s_n$ where $\alpha_i \neq D(s_0, s_1, \ldots, s_i)$ for some i, does not describe a path fragment that can be obtained from D.

The scheduler-types form a hierarchy, e.g., any SMD-scheduler can be viewed as a MD-scheduler (by ignoring parameter n) which, in turn, can be viewed as a HD-scheduler (by ignoring everything from the history except its length). A similar hierarchy exists between SMR, MR, and HR schedulers. Moreover, deterministic schedulers can be regarded as trivial versions of their corresponding randomized schedulers that assign probability 1 to the actions selected.

2.5. Induced stochastic process

Given a scheduler D (of arbitrary type) and a starting state, D induces a stochastic process on a CTMDP $\mathcal{M}$. For deterministic schedulers (HD, MD, and SMD), the induced process is a CTMC, referred to as $\mathcal{C}_D$ in the sequel. For MD- and HD-schedulers, though, the state space of $\mathcal{C}_D$ will in general be infinitely large (but countable). Formally, an HD-scheduler $D : S^+ \to Act$ on the CTMDP $\mathcal{M} = (S, Act, \mathbf{R})$ induces the CTMC $\mathcal{C}_D = (S_D, \mathbf{R}_D)$ with

[2] Strictly speaking we should write $X(\mathcal{M})$, but $\mathcal{M}$ is omitted as it should be clear from the context.

$S_D = S^+$ as state space, and

$$\mathbf{R}_D(\sigma, \sigma') = \begin{cases} \mathbf{R}(\mathit{last}(\sigma), D(\sigma), s) & \text{if } \sigma' = \sigma \to s, \\ 0 & \text{otherwise.} \end{cases}$$

In this particular construction, the embedded DTMC $\mathit{emb}(\mathcal{C}_D)$ is a tuple $(S_D, \mathbf{P}_D)$ where

$$\mathbf{P}_D(\sigma, \sigma') = \begin{cases} \dfrac{\mathbf{R}_D(\sigma, \sigma')}{E_D(\sigma)} & \text{if } E_D(\sigma) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Here, $E_D(\sigma) = \mathbf{R}_D(\sigma, S_D)$, i.e., the exit rate of σ in $\mathcal{C}_D$. States in CTMC $\mathcal{C}_D$ can be seen as state sequences $s_0 \to s_1 \to \cdots \to s_{n-1} \to s_n$ corresponding to time- and action-abstract path fragments in the original CTMDP $\mathcal{M}$. State s_n stands for the current state in the CTMDP whereas states s_0 through s_{n-1} describe the history. Intuitively, the stochastic process induced by an HD-scheduler D on the CTMDP $\mathcal{M}$ results from unfolding $\mathcal{M}$ into an (infinite) tree while resolving the nondeterministic choices according to D. For SMD-schedulers, the induced CTMC is guaranteed to be finite. More precisely, for SMD-scheduler D, $\mathcal{C}_D$ can be viewed as a CTMC with the original state space S, as all sequences that end in s, say, are lumping equivalent [14].

3. Maximum probability for timed reachability

Given a CTMDP $\mathcal{M}$, our aim is to compute the maximum (or minimum) probability to reach—under a given class of schedulers—a certain set B of states within t time units, when starting from a given state s. That is, we are looking for a method to calculate for time $t \geqslant 0$, $B \subseteq S$, $s \in S$ and class of X-schedulers:

$$\sup_{D \in X} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B)$$

up to some a priori given accuracy ε. Here $\Pr_D$ denotes the induced probability measure in $\mathcal{C}_D$. Intuitively, if B is considered as the set of “bad” states, then the value to be computed is the sharpest bound p for which it is guaranteed that the probability to reach a bad state from s in the next t time units is at most p under all “relevant” schedulers, i.e., all schedulers of type X.

In the sequel, unless otherwise stated, let $\mathcal{M}$ be uniform and E be its unique exit rate. Note that CTMC $\mathcal{C}_D$ which is obtained from the uniform CTMDP $\mathcal{M}$ by an HD-scheduler D is also uniform.

3.1. Approximation

To set the stage for the transformations that follow, we briefly discuss transient analysis of uniform CTMCs [26]. In a CTMC, the vector $\underline{\pi}(s, t)$ of time-dependent state probabilities can be written as:

$$\underline{\pi}(s,t) = (\Pr\{\sigma \in \mathit{Path}(s) \mid \sigma@t = s'\})_{s' \in S}$$

where $\sigma@t$ denotes the state occupied at time t on path σ. $\underline{\pi}(s,t)$ determines the probability to be in any of the states at time t, if starting in state s at time 0, and is characterised by a system of linear differential equations (cf. e.g. [29])

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\underline{\pi}(s,t) = \underline{\pi}(s,t)\cdot\mathbf{Q} \quad \text{given } \underline{\pi}(s,0) = \underline{i}_s,$$

where $\underline{i}_s$ denotes the characteristic vector for state s, and $\mathbf{Q} = \mathbf{R} - \mathit{diag}(\underline{E})$, with $\mathit{diag}(\underline{E})$ denoting the diagonal matrix with $\mathit{diag}(\underline{E})(s,s) = E(s)$ and 0 otherwise. If the CTMC $\mathcal{C} = (S, \mathbf{R})$ is uniform with rate E, a solution to these differential equations is given by the Taylor–MacLaurin series:

$$\underline{\pi}(s,t) = \sum_{n=0}^{\infty} \mathrm{e}^{-E\cdot t} \cdot \frac{(E\cdot t)^n}{n!} \cdot \mathbf{P}^n = \sum_{n=0}^{\infty} \psi(n) \cdot \mathbf{P}^n$$

where $\mathbf{P}$ is the probability matrix of the embedded DTMC of $\mathcal{C}$, and

$$\psi(n) = \mathrm{e}^{-E\cdot t} \cdot \frac{(E\cdot t)^n}{n!}$$

is used for fixed E and t as an abbreviation denoting the nth Poisson probability, i.e., $\psi(n)$ is the probability of n events occurring within t time units in a Poisson process with rate E. This abbreviation will re-occur in the sequel.

After these preliminaries, we now turn our attention to the problem of calculating timed-reachability probabilities. For a uniform CTMDP $\mathcal{M} = (S, \mathit{Act}, \mathbf{R})$ and an HD-scheduler D, the (infinite) vector of the probabilities $\Pr_D(\sigma, \overset{\leqslant t}{\rightsquigarrow} B)$ for all states σ in the CTMC $\mathcal{C}_D$ (i.e., all $\sigma \in S^+$) can now be given by:

$$\left(\Pr_D(\sigma, \overset{\leqslant t}{\rightsquigarrow} B)\right)_{\sigma \in S^+} = \sum_{n=0}^{\infty} \mathrm{e}^{-E\cdot t} \cdot \frac{(E\cdot t)^n}{n!} \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B = \sum_{n=0}^{\infty} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B,$$

where $\underline{i}_B = (i_B(\sigma))_{\sigma \in S^+}$ with $i_B(\sigma) = 1$ if $\mathit{last}(\sigma) \in B$, and 0 otherwise, and

$$\mathbf{P}_{D,B}(\sigma,\sigma') = \begin{cases} \mathbf{P}_D(\sigma,\sigma') & \text{if } \mathit{last}(\sigma) \notin B, \\ 1 & \text{if } \mathit{last}(\sigma) \in B \text{ and } \sigma' = \sigma, \\ 0 & \text{otherwise.} \end{cases}$$

$\mathbf{P}_{D,B}$ is the (infinite) transition probability matrix of the CTMC $\mathcal{C}_{D,B} = (S_D, \mathbf{R}_{D,B})$ that is obtained from $\mathcal{C}_D$ by equipping any B-state (i.e., any path $\sigma \in S^+$ with $\mathit{last}(\sigma) \in B$) with a self-loop and removing all its other outgoing transitions; similarly, we have:

$$\mathbf{R}_{D,B}(\sigma,\sigma') = \begin{cases} \mathbf{R}_D(\sigma,\sigma') & \text{if } \mathit{last}(\sigma) \notin B, \\ E & \text{if } \mathit{last}(\sigma) \in B \text{ and } \sigma' = \sigma, \\ 0 & \text{otherwise.} \end{cases}$$

The justification of this transformation is as in [6]: As the aim is to compute the probability to reach a B-state before a certain time bound, it is not of importance what happens once such a state has been visited, and therefore its outgoing transitions can be replaced by a self-loop.

Note that, for $s \in S$:

$$\begin{aligned}\Pr_D(s, \overset{\leqslant t}{\leadsto} B) &= \left(\sum_{n=0}^{\infty} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B\right)(s) \\ &= \psi(0)\cdot i_B(s) + \left(\sum_{n=1}^{\infty} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B\right)(s).\end{aligned}$$

Later we will exploit that for $s \notin B$, $i_B(s) = 0$ and therefore

$$\Pr_D(s, \overset{\leqslant t}{\leadsto} B) = \left(\sum_{n=1}^{\infty} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B\right)(s) \quad \text{provided } s \notin B.$$

Rather than computing the precise maximum probabilities we use an approximation in the following way: For any state s, the value $\Pr_D(s, \overset{\leqslant t}{\leadsto} B)$ will be approximated, up to a given accuracy ε, by

$$\widetilde{\Pr}_D(s, \overset{\leqslant t}{\leadsto} B) = \left(\sum_{n=0}^{k} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B\right)(s),$$

where $k = k(\varepsilon, E, t)$ depends on ε, E and t, but neither on state s nor on scheduler D. This can be seen as follows: Let $\|\cdot\|$ denote the row-sum norm. Then, for any $k \geqslant 0$:

$$\begin{aligned}&\left\| \sum_{n=0}^{\infty} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B - \sum_{n=0}^{k} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B \right\| \\ &\quad = \left\| \sum_{n=k+1}^{\infty} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B \right\| \\ &\quad \leqslant \sum_{n=k+1}^{\infty} \psi(n) \cdot \underbrace{\|\mathbf{P}^n_{D,B}\|}_{\leqslant 1} \cdot \underbrace{\|\underline{i}_B\|}_{\leqslant 1} \leqslant \sum_{n=k+1}^{\infty} \psi(n).\end{aligned}$$

Hence, for sufficiently large $k = k(\varepsilon, E, t)$:

$$\left\| \sum_{n=0}^{\infty} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B - \sum_{n=0}^{k} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B \right\| \leqslant \sum_{n=k+1}^{\infty} \psi(n) \leqslant \varepsilon.$$

Note that

$$\sum_{n=0}^{\infty} \mathrm{e}^{-E\cdot t} \cdot \frac{(E\cdot t)^n}{n!} = \sum_{n=0}^{\infty} \psi(n) = 1.$$

Hence, for any scheduler D and state s:

$$\Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B) - \varepsilon \leqslant \widetilde{\Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B) = \left(\sum_{n=0}^{k} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B\right)(s) \leqslant \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B).$$

Our strategy is to construct some HD-scheduler D_0 such that for any state $s \in S$:

$$\widetilde{\Pr}_{D_0}(s, \overset{\leqslant t}{\rightsquigarrow} B) \geqslant \sup_{D \in HD} \widetilde{\Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B). \tag{1}$$

This yields:

$$\begin{aligned}\sup_{D\in HD} \underbrace{\Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B) - \varepsilon}_{\leqslant \widetilde{\Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B)} &\leqslant \widetilde{\Pr}_{D_0}(s, \overset{\leqslant t}{\rightsquigarrow} B) \\ &\leqslant \Pr_{D_0}(s, \overset{\leqslant t}{\rightsquigarrow} B) \\ &\leqslant \sup_{D\in HD} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B).\end{aligned}$$

Thus, (1) implies that D_0 approximates $\sup_{D\in HD} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B)$ up to ε.

Since $\mathbf{P}^n_{D,B}(s, \sigma) = 0$ for any σ containing more than n transitions, i.e., more than $n+1$ states, the value

$$\widetilde{\Pr}_{D_0}(s, \overset{\leqslant t}{\rightsquigarrow} B) = \left(\sum_{n=0}^{k} \psi(n) \cdot \mathbf{P}^n_{D_0,B} \cdot \underline{i}_B\right)(s)$$

only depends on the kth truncation of D_0, i.e., the function

$$D_0\Big|_k : \bigcup_{0<n\leqslant k} S^n \to Act, \quad D_0\Big|_k (\sigma) = D_0(\sigma).$$

Intuitively speaking, only the first k decisions of D_0 are relevant (and not "later" ones) for determining the value $\widetilde{\Pr}_{D_0}(s, \overset{\leqslant t}{\rightsquigarrow} B)$. There are only finitely many such truncations when ranging over all HD-schedulers. A brute-force approach would consider all of them in order to determine the maximum. This technique is effective, but is highly inefficient because the total number of such truncations, $\prod_{s\in S} |Act(s)|^k$, grows exponentially in the number of states s with $|Act(s)| > 1$. Note that

$$\prod_{s\in S} |Act(s)|^k \geqslant 2^{|T|k} \quad \text{if } |Act(s)| \geqslant 2 \quad \text{for all } s \in T \subseteq S,$$

i.e., the total number of truncations to be considered is exponential in k.

3.2. A greedy algorithm

Due to the inefficiency of the above brute-force method, we are striving for a more practical solution to the timed reachability problem. To this end, we consider only a limited fragment of HD-schedulers. We restrict to truncated MD-schedulers of the form $D : S \times \{1, \ldots, k\} \to Act$. Later on, it is shown that considering such schedulers suffices.

The actions $act(s,i) \in Act(s)$ for $0 < i \leqslant k$ will be determined such that the truncated MD-scheduler D_0 with $D_0(s,i) = act(s,i)$ fulfills Eq. (1). Let $\mathbf{P}_i$ denote the probability matrix of cardinality $|S| \times |S|$ where the row $\mathbf{P}_i(s,\cdot) = \mathbf{P}(s, act(s,i), \cdot)$ if $s \notin B$ and $\mathbf{P}_i(s,\cdot) = \underline{i}_s$ if $s \in B$. $\mathbf{P}_i$ thus denotes the probability matrix induced by the scheduler D_0 at step i.

For $s \notin B$, the actions $act(s,i)$ will be determined in a backward manner, i.e., starting from $i = k$. For $i = k$, the selected action $act(s,k) \in Act(s)$ satisfies:

$$\mathbf{P}_k(s,B) = \mathbf{P}(s, act(s,k), B) = \max_{\alpha \in Act(s)} \mathbf{P}(s, \alpha, B).$$

That is, $\mathbf{P}_k(s,\cdot)$ is determined such that for any state s the probability to move to a B-state within at most one step is maximized. Generalizing this strategy, for $i < k$, we assume that we are given actions $act(s,j)$ for $i < j \leqslant k$ and choose action $act(s,i)$ such that the probability to move to a B-state within at most $k-i+1$ steps is maximized under the truncated MD-scheduler $D : S \times \{1, \ldots, k-i+1\} \to Act$ defined by:

$$D(s,j) = act(s, i+j-1) \quad \text{for } 0 < j \leqslant k-i+1.$$

That is, $\mathbf{P}_i$ is constructed such that for $i \geqslant 1$ the vector

$$\underline{q}_i = \sum_{n=i}^{k} \psi(n) \cdot \mathbf{P}_i \cdot \mathbf{P}_{i+1} \cdot \ldots \cdot \mathbf{P}_n \cdot \underline{i}_B$$

is state-wise maximized under all vectors of the form

$$\sum_{n=i}^{k} \psi(n) \cdot \mathbf{P}_* \cdot \mathbf{P}_{i+1} \cdot \ldots \cdot \mathbf{P}_n \cdot \underline{i}_B$$

where $\mathbf{P}_*$ is an $|S| \times |S|$-matrix with $\mathbf{P}_*(s,\cdot) = \mathbf{P}(s,\alpha,\cdot)$ for some action $\alpha \in Act(s)$ if $s \notin B$ and $\mathbf{P}_*(s,\cdot) = \underline{i}_s$ if $s \in B$. In the above equations, $\underline{i}_B = (i_B(s))_{s \in S}$ stands for the bit-vector that represents the characteristic function of B (as a subset of the original state space S), i.e., $i_B(s) = 1$ if $s \in B$ and $i_B(s) = 0$ if $s \in S \setminus B$.[3]

Informally, $q_i(s)$ is the maximum conditional probability to reach B taking i to k steps within t time units, given that state s is occupied before the ith step. We let $\underline{q} = \psi(0) \cdot \underline{i}_B + \underline{q}_1$, which for the $(S \setminus B)$-states agrees with the desired probability vector to reach a B-state within at most k steps when the time bound to reach B is t. For $s \in B$ we have $\Pr_D(s, \stackrel{\leqslant t}{\rightsquigarrow} B) = 1$. Moreover, for $s \notin B$ it holds

$$q(s) = \psi(0) \cdot i_B(s) + q_1(s) = q_1(s)$$

as $i_B(s) = 0$. In the sequel, we are therefore only interested in the calculation of the vector $\underline{q}_1$.

[3] At several other places, we shall use the same notation $\underline{i}_B$ for the bit-vector $(i_B(\sigma))_{\sigma \in S^+}$ that represents the characteristic function of B viewed as subset of the state space of the CTMC induced by an HD-scheduler. Here, we identify B with the set of finite paths σ where $last(\sigma) \in B$. Whenever the notation $\underline{i}_B$ occurs in our formulae the dimension of $\underline{i}_B$ should be clear from the context.

The main steps of our procedure are summarized in Algorithm 1. A stable and efficient algorithm to compute the Poisson probabilities $\psi(i)$ has been proposed in [22] and can be adopted here. Note that for the computation of the values $\sup_{D\in HD} \widetilde{\Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B)$ there is no need to compute (and store) the matrices $\mathbf{P}_i$. Instead, it suffices to compute the vectors

$$\begin{aligned}
\underline{q}_i &= \sum_{n=i}^{k} \psi(n) \cdot \mathbf{P}_i \cdot \mathbf{P}_{i+1} \cdot \ldots \cdot \mathbf{P}_n \cdot \underline{i}_B \\
&= \psi(i) \cdot \mathbf{P}_i \cdot \underline{i}_B + \sum_{n=i+1}^{k} \psi(n) \cdot \mathbf{P}_i \cdot \mathbf{P}_{i+1} \cdot \ldots \cdot \mathbf{P}_n \cdot \underline{i}_B \\
&= \psi(i) \cdot \mathbf{P}_i \cdot \underline{i}_B + \mathbf{P}_i \cdot \sum_{n=i+1}^{k} \psi(n) \cdot \mathbf{P}_{i+1} \cdot \ldots \cdot \mathbf{P}_n \cdot \underline{i}_B \\
&= \psi(i) \cdot \mathbf{P}_i \cdot \underline{i}_B + \mathbf{P}_i \cdot \underline{q}_{i+1}
\end{aligned}$$

This equality holds for $1 \leqslant i < k$, but can be extended to $i \leqslant k$ by setting $\underline{q}_{k+1} = \mathbf{0}$, i.e., to the 0-vector. For $s \notin B$, we have $(\mathbf{P}_i \cdot \underline{i}_B)(s) = \mathbf{P}(s, \alpha, B)$ if $\alpha = act(s, i)$.

Algorithm 1 Greedy approximation algorithm for computing $\sup_{D\in HD} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B)$

```
k := k(ε, E, t);                                   (* determine number of required steps *)
for all s ∈ S do q_{k+1}(s) := 0; od               (* initialize q_{k+1} as null-vector *)
for all i = k, k−1, ..., 1 do
   for all s ∈ S \ B do
      m := −1;
      for all α ∈ Act(s) do                         (* search the optimal row P_i(s, ·) *)
         m := max( m, ψ(i) · P(s, α, B) + Σ_{s'∈S} P(s, α, s') · q_{i+1}(s') );
      od
      q_i(s) := m;                                  (* choose maximum *)
   od
   for all s ∈ B do q_i(s) := ψ(i) + q_{i+1}(s); od  (* P_i(s, ·) := i_s for all s ∈ B *)
od
for all s ∈ S do
   if s ∉ B then q(s) := q_1(s); else q(s) := 1; fi
od
return the vector q.
```

3.3. Complexity of the algorithm

Algorithm 1 can be implemented with a space complexity in $\mathcal{O}(|S|^2 \cdot |Act| + |S|)$, where the term $|S|^2 \cdot |Act|$ stands for the representation of the uniform CTMDP $\mathcal{M}$ while the term $|S|$ stands for the vectors $\underline{q}_{i+1}$ and $\underline{q}_i$. Note that there is no need to store $\underline{q}_{i+1}$ once $\underline{q}_i$ has been computed. The values $q_i(s, \alpha)$ are only needed temporarily, and as mentioned before, there is no need to compute and store the matrices $\mathbf{P}_i$. Inspection of the pseudo-code of Algorithm 1 reveals that the worst-case time complexity is asymptotically bounded by:

$$k \cdot \sum_{s \in S \setminus B} \sum_{\alpha \in Act(s)} |\{ s' \in S \mid \mathbf{R}(s, \alpha, s') > 0 \}|$$

which is in $\mathcal{O}\left(E \cdot t \cdot |S|^2 \cdot |Act|\right)$. Note that $k = k(\varepsilon, E, t)$ grows proportionally with $E \cdot t$. This bound on the running time can be improved by performing a reachability analysis (as a preprocessing phase of Algorithm 1) to determine the set T of states from which a B-state can be reached. The main iteration then only needs to be performed for all states in $T \setminus B$ rather than $S \setminus B$. For the other states we have, for any scheduler D, $\Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B) = 0$ for $s \in S \setminus T$, and $\Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B) = 1$ for $s \in B$.

3.4. Correctness of the algorithm

Although our greedy algorithm is based on a truncated MD-scheduler—only the first k steps are memorized—it approximates the maximum probability to reach the set of states B within t time units under *all* HD-schedulers. This is shown by the following theorem where $q(s)$ is the s-component of the vector $\underline{q}$ as returned by Algorithm 1.

Theorem 5. $\sup_{D \in HD} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B) - \varepsilon \leqslant q(s) \leqslant \sup_{D \in HD} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B)$.

Proof. The rightmost inequality follows immediately. For $s \in B$ the inequality reduces to $1-\varepsilon \leqslant 1 \leqslant 1$ which is obviously fulfilled. For $s \notin B$, it suffices for the leftmost inequality to show that for any HD-scheduler D and $\sigma \in S^+$:

$$q_i(last(\sigma)) \geqslant q_i^D(\sigma), \quad i = 1, 2, \ldots, k, \quad \text{where } \underline{q}_i^D = \sum_{n=i}^{k} \psi(n) \cdot \mathbf{P}_{D,B}^{n-i+1} \cdot \underline{i}_B. \quad (2)$$

Note that $\underline{q}_i^D$ is an infinite vector with a component for each finite path $\sigma \in S^+$. Let

$$\widehat{q}^D(\sigma) = \sum_{n=0}^{k} \psi(n) \cdot \mathbf{P}_{D,B}^n \cdot i_B(\sigma) = \psi(0) \cdot i_B(\sigma) + q_1^D(\sigma)$$

denote the probability in the CTMC $\mathcal{C}_D$ induced by D to reach a B-state (i.e., a path σ' with $last(\sigma') \in B$) from σ—viewed as state in $\mathcal{C}_D$—within k steps in at most t time units. Note that if $last(\sigma) \notin B$ then the first summand equals 0, i.e., for that case $\widehat{q}^D(\sigma) = q_1^D(\sigma)$.

The reason why it is sufficient to consider (2) is as follows. If $s \notin B$ then

$$\begin{aligned}\Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B) &= \sum_{n=1}^{\infty} \psi(n) \cdot \mathbf{P}_{D,B}^n \cdot i_B(s) \\ &\leqslant \sum_{n=1}^{k} \psi(n) \cdot \mathbf{P}_{D,B}^n \cdot i_B(s) + \varepsilon\end{aligned}$$

$$\leqslant \sup_{\substack{\sigma\in S^+\\ last(\sigma)=s}} \left(\sum_{n=1}^{k} \psi(n)\cdot \mathbf{P}^n_{D,B}\cdot \underline{i}_B\right)(\sigma)+\varepsilon$$
$$= \sup_{\substack{\sigma\in S^+\\ last(\sigma)=s}} q_1^D(\sigma)+\varepsilon.$$

Hence, from (2) we derive:

$$\sup_{D\in HD} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B)-\varepsilon \leqslant \sup_{D\in HD}\ \sup_{\substack{\sigma\in S^+\\ last(\sigma)=s}} q_1^D(\sigma)\leqslant q(s).$$

We now prove (2), distinguishing two cases. If $\sigma \in S^+$ and $s = last(\sigma) \in B$ then $q_i^D(s) = \sum_{n=i}^k \psi(n) = q_i(s)$. For $last(\sigma) \in S\backslash B$ we prove that $q_i(last(\sigma))\geqslant q_i^D(\sigma)$ by a "downward" induction on i. Let $\sigma \in S^+$, $s = last(\sigma) \in S\backslash B$, and $\alpha = D(\sigma)$ (recall that D is an HD-scheduler and thus may be considered as a function $S^+ \to Act$).

Base of induction $i = k$:

$$\begin{aligned}
q_k^D(\sigma) &= (\psi(k)\cdot \mathbf{P}_{D,B}\cdot \underline{i}_B)(\sigma)\\
&= \psi(k)\cdot \sum_{\sigma'\in S^+} \underbrace{\mathbf{P}_{D,B}(\sigma,\sigma')}_{\substack{\mathbf{P}(s,\alpha,s'),\text{if } \sigma'=\sigma\to s'\\ \text{and 0 otherwise.}}} \cdot \underbrace{i_B(\sigma')}_{\substack{1,\text{if } last(\sigma')\in B\\ 0,\text{ if } last(\sigma')\notin B}}\\
&= \psi(k)\cdot \mathbf{P}(s,\alpha,B)\\
&\leqslant \max_{\beta\in Act(s)} \psi(k)\cdot \mathbf{P}(s,\beta,B)\\
&= q_k(s).
\end{aligned}$$

Induction step $i+1 \Longrightarrow i$ (where $k > i\geqslant 1$): First, observe that

$$q_i(s) = \max_{\beta\in Act(s)} \left(\psi(i)\cdot \mathbf{P}(s,\beta,B) + \sum_{s'\in S} \mathbf{P}(s,\beta,s')\cdot q_{i+1}(s')\right).$$

Consider an arbitrary HD-scheduler D. As before, $\alpha = D(\sigma)$ and $s = last(\sigma)$. Then:

$$\begin{aligned}
q_i^D(\sigma) &= \left(\sum_{n=i}^{k} \psi(n)\cdot \mathbf{P}^{n-i+1}_{D,B}\cdot \underline{i}_B\right)(\sigma)\\
&= (\psi(i)\cdot \mathbf{P}_{D,B}\cdot \underline{i}_B)(\sigma) + \left(\sum_{n=i+1}^{k} \psi(n)\cdot \mathbf{P}_{D,B}\cdot \mathbf{P}^{n-i}_{D,B}\cdot \underline{i}_B\right)(\sigma)\\
&= \psi(i)\cdot \mathbf{P}(s,\alpha,B)\\
&\quad + \sum_{n=i+1}^{k} \psi(n)\cdot \sum_{s'\in S} \underbrace{\mathbf{P}_{D,B}(\sigma,\sigma\to s')}_{=\mathbf{P}(s,\alpha,s')}\cdot \underbrace{(\mathbf{P}^{n-i}_{D,B}\cdot \underline{i}_B)(\sigma\to s')}_{(\sigma\to s')\text{-component of } \mathbf{P}^{n-i}_{D,B}\cdot \underline{i}_B}\\
&= \psi(i)\cdot \mathbf{P}(s,\alpha,B)\\
&\quad + \sum_{s'\in S} \mathbf{P}(s,\alpha,s')\cdot \underbrace{\left(\sum_{n=i+1}^{k} \psi(n)\cdot \mathbf{P}^{n-i}_{D,B}\cdot \underline{i}_B\right)(\sigma\to s')}_{q^D_{i+1}(\sigma\to s')}
\end{aligned}$$

$$\begin{aligned}&\leqslant \psi(i)\cdot \mathbf{P}(s,\alpha,B)\\&\quad+\sum_{s'\in S}\mathbf{P}(s,\alpha,s')\cdot q_{i+1}(s') \qquad \text{(* by induction hypothesis *)}\\&\leqslant q_i(s). \qquad \square\end{aligned}$$

As a result, the vector computed by Algorithm 1 is state-wise optimal under all HD-schedulers, up to the accuracy ε.

4. Other scheduling disciplines

By Theorem 5 it follows that our greedy algorithm computes the maximum probability for timed reachability under all HD-schedulers. In this section, we show that this also applies to any MR-, MD-, and, more importantly, to any HR-scheduler. In addition, we will show that this does neither hold for SMD-schedulers nor for schedulers that can base their decision on the timing of actions. Finally, it is shown that adding a simple notion of fairness is invariant under these maximum probabilities for HD-schedulers.

Throughout this section, we assume a fixed uniform CTMDP $\mathcal{M}$.

4.1. Markovian deterministic schedulers

In the sequel, let $s \in S$ be a state, $t \geqslant 0$ a time point and $B \subseteq S$ a set of states. Theorem 5 states that the vector computed by Algorithm 1 is state-wise optimal under all HD-schedulers, up to a given accuracy ε. As Algorithm 1 calculates, in fact, a truncation of an MD-scheduler, it directly follows that the suprema under MD- and HD-schedulers agree:

Theorem 6. $\sup_{D\in MD} \Pr_D(s, \stackrel{\leqslant t}{\rightsquigarrow} B) = \sup_{D\in HD} \Pr_D(s, \stackrel{\leqslant t}{\rightsquigarrow} B)$.

4.2. History-dependent randomized schedulers

The next result yields that the supremum under HD- and HR-schedulers coincides:

Theorem 7. $\sup_{D\in HD} \Pr_D(s, \stackrel{\leqslant t}{\rightsquigarrow} B) = \sup_{D\in HR} \Pr_D(s, \stackrel{\leqslant t}{\rightsquigarrow} B)$.

Proof. The proof is based on the cylinder set construction for a CTMC given in Section 2.3. We have that under each HD-scheduler D,

$$\Pr_D(s, \stackrel{\leqslant t}{\rightsquigarrow} B) = \lim_{n\to\infty} \Pr_D(s, \stackrel{\leqslant t}{\rightsquigarrow}_{\leqslant n} B),$$

where the subscript $\leqslant n$ denotes that B has to be reached within at most n steps.

Hence, it suffices to show that for fixed $n \in \mathbb{N}$ there is a finite family $(D_i)_{i\in J_n}$ (with J_n an index set) of HD-schedulers such that the measure $\Pr_{D'}$ induced by an HR-scheduler D' for the cylinder sets induced by path fragments consisting of n transitions is a convex combination of the measures $\Pr_{D_i}$, $i \in J_n$. We prove this claim by induction on n.

Base of induction $n = 0$: The basic cylinder induced by a path fragment with 0 transitions (i.e., a path fragment consisting just of a state s) is the set of all paths that start in state s. The measure of this set (for starting state s) is 1 under all schedulers.

Induction step $n \Longrightarrow n+1$: Assume by induction hypothesis that there is a finite family $(D_i)_{i\in J_n}$ of HD-schedulers and values $p_i \in [0,1]$, $i \in J_n$ with $\sum_{i\in J_n} p_i = 1$ and for all basic cylinders $\mathcal{C} = C(s_0 \xrightarrow{\alpha_0, I_0} \cdots \xrightarrow{\alpha_{n-1}, I_{n-1}} s_n)$, we have:

$$\Pr_{D'}(\mathcal{C}) = \sum_{i\in J_n} p_i \cdot \Pr_{D_i}(\mathcal{C})$$

(with the obvious lifting of the cylinders introduced in section refmcprelim from action-abstract to action-labelled paths).

Let $\mathcal{F}$ denote the set of functions

$$f : (S \times Act)^{n+1} \times S \to Act$$

such that $f(\sigma \to s) \in Act(s)$. That is, $\mathcal{F}$ is a finite set of schedulers, containing those HD-schedulers that decide after $n+1$ steps. For $f \in \mathcal{F}$, let $\mu_f : (S\times Act)^{n+1} \times S \to Distr(Act)$ be the trivial probability distribution induced by f, defined by:

$$\mu_f(\sigma)(\alpha) = \begin{cases} 1 & \text{if } f(\sigma) = \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

We now consider how to construct any HR-scheduler μ which decides after $n+1$ steps. Each function $\mu : (S \times Act)^{n+1} \times S \to Distr(Act)$ (such that $\mu(\sigma)(\alpha) > 0$ implies $\alpha \in Act(s)$ for all σ of length $n+1$) can be written as a finite convex combination of the (distributions induced by the) functions $f \in \mathcal{F}$:

$$\mu = \sum_{f\in\mathcal{F}} q_f \cdot \mu_f \quad \text{where} \sum_{f\in\mathcal{F}} q_f = 1 \quad \text{and} \quad 0 \leqslant q_f \leqslant 1.$$

for appropriately chosen q_f, $f \in \mathcal{F}$. This fact can be seen as follows.

(1) Choose some $f_1 \in \mathcal{F}$ such that for all σ and α: $f_1(\sigma) = \alpha$ implies $\mu(\sigma)(\alpha) > 0$. Let

$$q_{f_1} = \min\{\mu(\sigma)(f_1(\sigma)) \mid \sigma \in (S \times Act)^{n+1} \times S\}$$

be the minimal probability with which some action may be selected after having performed $n+1$ steps, and

$$\mu_1(\sigma, \alpha) = \begin{cases} \mu(\sigma)(\alpha) - q_{f_1} & \text{if } f_1(\sigma) = \alpha, \\ \mu(\sigma)(\alpha) & \text{otherwise} \end{cases}$$

be the remaining probability mass. Then, for all σ we have:

$$\sum_{\alpha\in Act} \mu_1(\sigma, \alpha) = 1 - q_{f_1}$$

(2) As a next step, we choose some $f_2 \in \mathcal{F}$ such that for all σ and α: $f_2(\sigma) = \alpha$ implies $\mu_1(\sigma, \alpha) > 0$. (Note the slight, but essential, difference with f_1.) Let

$$q_{f_2} = \min\{\mu_1(\sigma, f_2(\sigma)) \mid \sigma \in (S \times Act)^{n+1} \times S\}$$

be the minimal probability with which some action may be selected after having performed $n+1$ steps, *and* after having "spent" the probability mass to select an action according to f_1, and

$$\mu_2(\sigma,\alpha)=\begin{cases}\mu_1(\sigma,\alpha)-q_{f_2} & \text{if } f_2(\sigma)=\alpha,\\ \mu_1(\sigma,\alpha), & \text{otherwise}\end{cases}$$

be the remaining probability mass after having spent probability q_{f_2}. Then, for all σ:

$$\sum_{\alpha\in Act}\mu_2(\sigma,\alpha)=1-q_{f_1}-q_{f_2}.$$

(3) This recipe is repeated until $\mu_j(\sigma,\alpha)=0$ for all σ and α, i.e., until there is no probability mass left to be distributed among possible actions.

We now consider the function μ where $\mu(\sigma)(\alpha)=D'(\sigma)(\alpha)$. Let q_f be as above, i.e.,

$$D'(\sigma)=\sum_{f\in\mathcal{F}}q_f\cdot\mu_f(\sigma,\cdot)\quad\text{for all } \sigma\in(S\times Act)^{n+1}\times S. \tag{3}$$

Let $\sigma = s_0\xrightarrow{\alpha_0}s_1\xrightarrow{\alpha_1}\cdots,\xrightarrow{\alpha_{n-1}}s_n\xrightarrow{\alpha_n}s_{n+1}\in(S\times Act)^{n+1}\times S$ and $\mathcal{C}$ be a basic cylinder which relies on the time-abstract path σ, but which has arbitrary time-intervals:

$$\mathcal{C}=C(s_0\xrightarrow{\alpha_0,I_0}\cdots\xrightarrow{\alpha_{n-1},I_{n-1}}s_n\xrightarrow{\alpha_n,I_n}s_{n+1}).$$

Furthermore, let

$$\mathcal{C}'=C(s_0\xrightarrow{\alpha_0,I_0}s_1\xrightarrow{\alpha_1,I_1}\cdots\xrightarrow{\alpha_{n-1},I_{n-1}}s_n).$$

Now, setting $\mathbf{P}(s,\alpha,I,s')=\mathbf{P}(s,\alpha,s')(\mathrm{e}^{-E(s,\alpha)t}-\mathrm{e}^{-E(s,\alpha)t'})$ with $t=\inf I$, $t'=\sup I$, we have

$$\begin{aligned}\Pr_{D'}(\mathcal{C}) &= \Pr_{D'}(\mathcal{C}')\cdot D'(\sigma)(\alpha_n)\cdot\mathbf{P}(s_n,\alpha_n,I_n,s_{n+1})\\ &\overset{\text{ind.hypo.}}{=} \sum_{i\in J_n}p_i\cdot\Pr_{D_i}(\mathcal{C}')\cdot D'(\sigma)(\alpha_n)\cdot\mathbf{P}(s_n,\alpha_n,I_n,s_{n+1})\\ &\overset{(3)}{=} \sum_{i\in J_n}p_i\cdot\Pr_{D_i}(\mathcal{C}')\cdot\sum_{f\in\mathcal{F}}q_f\cdot\mu_f(\sigma)(\alpha_n)\cdot\mathbf{P}(s_n,\alpha_n,I_n,s_{n+1})\\ &= \sum_{(i,f)\in J_n\times\mathcal{F}}\underbrace{p_i\cdot q_f}_{p_{i,f}:=}\cdot\underbrace{\Pr_{D_i}(\mathcal{C}')\cdot\mu_f(\sigma)(\alpha_n)\cdot\mathbf{P}(s_n,\alpha_n,I_n,s_{n+1})}_{\Pr_{D_{i,f}}(\mathcal{C}):=}\\ &= \sum_{(i,f)\in J_n\times\mathcal{F}}p_{i,f}\cdot\Pr_{D_{i,f}}(\mathcal{C}),\end{aligned}$$

where $D_{i,f}$ is an HD-scheduler which agrees with D_i on $\bigcup_{0\leqslant m\leqslant n}(S\times Act)^m\times S$ and with f on $(S\times Act)^{n+1}\times S$. We may now define $J_{n+1}=J_n\times\mathcal{F}$. □

Let us illustrate the crucial part of the proof, the recipe to calculate the weights q_{f_i}, by means of a small example. Suppose that there are two paths, σ and σ' say, with $last(\sigma)\neq last(\sigma')$, both of length $n+1$. Let $Act(last(\sigma))=\{\alpha,\beta\}$ and $Act(last(\sigma'))=\{\alpha,\gamma\}$. Assume μ is defined such that $\mu(\sigma)(\alpha)=\frac{5}{6}$, $\mu(\sigma)(\beta)=\frac{1}{6}$, $\mu(\sigma')(\alpha)=\frac{1}{3}$ and $\mu(\sigma')(\gamma)=\frac{2}{3}$.

- We choose $f_1(\sigma) = f_1(\sigma') = \alpha$. Then $q_{f_1} = \min(\frac{5}{6}, \frac{1}{3}) = \frac{1}{3}$, whence $\mu_1(\sigma)(\alpha) = \frac{1}{2}$, $\mu_1(\sigma')(\alpha) = 0$, and all other values of $\mu_1(\cdot)(\cdot)$ agree with $\mu(\cdot)(\cdot)$.
- We now choose $f_2(\sigma) = \alpha$ and $f_2(\sigma')=\gamma$. Then $q_{f_2}=\min(\frac{1}{2}, \frac{2}{3}) = \frac{1}{2}$, whence $\mu_2(\sigma)(\alpha) = 0$, $\mu_2(\sigma')(\gamma) = \frac{1}{6}$, and all other values of $\mu_2(\cdot)(\cdot)$ agree with $\mu_1(\cdot)(\cdot)$. The only remaining non-zero value is $\mu(\sigma)(\beta) = \frac{1}{6}$.
- We now choose $f_3(\sigma) = \beta$ and $f_2(\sigma') = \gamma$. Then $q_{f_3} = \min(\frac{1}{6}, \frac{1}{6}) = \frac{1}{6}$, whence $\mu_3(\cdot)(\cdot)$ is constant 0. Thus the process terminates after 3 steps with a μ_3 which assigns probability 0 to all paths and all actions.

A few remarks are in order. Theorems 6 and 7 show that the suprema for the probabilities to reach a set of goal states within a given time bound under the classes of scheduler MD, HD, MR and HR coincide. (For MR-schedulers this stems from the fact that $MD \subseteq MR \subseteq HR$.) For probabilities of some other types of events, however, such correspondence can not always be established. That is, in general, randomized schedulers can be better than deterministic schedulers. This observation was made by Beutler and Ross [11] who showed that the maximum of time-average rewards under randomized schedulers might be larger than under deterministic schedulers. In fact, the crux of the proof of Theorem 7 is the observation that the values $\Pr_D(s, \overset{\leqslant t}{\rightsquigarrow}_{\leqslant n} B)$ converge to $\Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B)$, where the subscript $\leqslant n$ denotes that B has to be reached within at most n steps. This property is not guaranteed for other types of events.

4.3. Stationary Markovian deterministic schedulers

Different from the discrete time setting, where SMD-schedulers suffice for maximum probabilities to reach a set of goal states within a given number of steps [12,9], this does not hold for the corresponding question—interpreting the number of steps in the discrete case as elapse of time—on CTMDPs. A counterexample is given in Fig. 1(a). Here, states are represented as circles and there is an edge between states s and s' labelled with action α if and only if $\mathbf{R}(s, \alpha, s') > 0$. Action labels and rates are indicated at each edge. Let $B = \{ s_2 \}$, and consider the only two relevant SMD-schedulers, D_α, selecting action α in state s_0, and D_β, selecting action β. Comparing them with $D_{\beta\alpha}$, i.e., the scheduler that after selecting β once switches to selecting α in state s_0, we find that for a certain range of time bounds t, $D_{\beta\alpha}$ outperforms both D_β and D_α. Intuitively, the probability of stuttering in state s_0 (by choosing β initially) may influence the remaining time to reach B to an extent that it becomes profitable to continue choosing α. For $t = 0.5$, for instance, $\Pr_{D_{\beta\alpha}}(s_0, \overset{\leqslant 0.5}{\rightsquigarrow} B) = 0.4152$, whereas for D_α and D_β these probabilities are 0.3935 and 0.3996, respectively. Thus, SMD-schedulers are not expressive enough for maximum probabilities to reach a set of goal states within a given time bound under all HD/HR-schedulers. For SMR-schedulers this is an open issue.

4.4. Timed schedulers

This paper only considers schedulers that do not take the timing information into account. It is, however, worth noticing that timed history-dependent (THD) schedulers are more

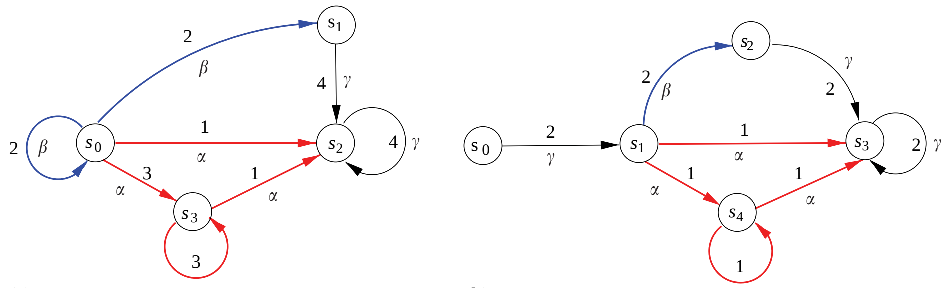

Fig. 1. Uniform CTMDPs where (a) SMD-schedulers are less powerful, and (b) where THD schedulers are more powerful than HD-schedulers.

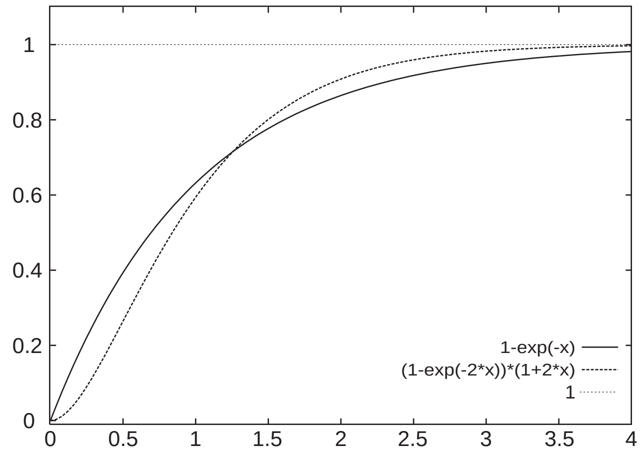

Fig. 2. Functions $1 - e^{-t}$ and $1 - e^{-2t} \cdot (1 + 2t)$ for $t \geqslant 0$.

powerful than time-abstract history-dependent schedulers (class HD and HR), in the sense that it is possible that:

$$\sup_{D \in THD} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B) > \sup_{D \in HD} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B).$$

Here, *THD* refers to the class of schedulers given by functions $D : (S \times Act \times \mathbb{R}_{>0})^* \times S \to Act$ (only choosing from $Act(s)$ for any path ending in state s), i.e., THD-schedulers are able to observe the time points of state changes. To see that they may yield a higher probability, consider for example the uniform CTMDP in Fig. 1(b), with $B = \{s_3\}$. In this example, it depends on the time instance of entering s_1 whether it is more profitable to continue choosing α or β. To be more precise, consider the only relevant HD-schedulers, D_α (choosing α in s_1) and D_β (choosing β). Fig. 2 plots the probability to reach B starting from state s_1 if choosing D_α, respectively D_β, given by

$$\mathrm{Pr}_{D_\alpha}(s_1, \overset{\leqslant t}{\rightsquigarrow} B) = 1 - \mathrm{e}^{-t} \quad \text{and} \quad \mathrm{Pr}_{D_\beta}(s_1, \overset{\leqslant t}{\rightsquigarrow} B) = 1 - \mathrm{e}^{-2t} \cdot (1 + 2t).$$

Let t_0 be the time instance satisfying $\mathrm{e}^{t_0} = 1+2\,t_0$, i.e., the time point where both plots cross. The THD-scheduler D defined by $D(s_0 \xrightarrow{\gamma,u} s_1) = \alpha$ if $t-u < t_0$ and β otherwise, maximizes the probability to reach $B = \{ s_3 \}$ from state s_0 within t time units, and obviously outperforms both D_α and D_β.

4.5. Fairness

We conclude this section by considering a simple notion of fairness for schedulers. Let $\sigma = s_0 \xrightarrow{\alpha_0,t_0} s_1 \xrightarrow{\alpha_1,t_1} \cdots$ be an infinite path. Infinite path σ is called *fair* if and only if for each state s that occurs infinitely often in σ and each action $\alpha \in Act(s)$, there are infinitely many indices n such that $(s_n, \alpha_n) = (s, \alpha)$. Stated in words, for any state that is visited infinitely often, each of its outgoing actions cannot have been selected only a finite number of times. (Note that this notion of fairness is rather weak; for instance, a scheduler that finitely many times selects the same action in a state that is visited only finitely often—without ever considering one of the other possibilities—is considered to be fair.) Scheduler D (of some class) is called fair if and only if

$$\mathrm{Pr}_D\{\sigma \in Path(s) \mid \sigma \text{ is fair }\} = 1$$

for all states $s \in S$. Let FHD denote the set of all fair HD-schedulers. The following result states that maximum probabilities under HD-schedulers and their fair counterparts coincide:

Theorem 8. $\sup_{D \in HD} \mathrm{Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B) = \sup_{D \in FHD} \mathrm{Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B)$.

Proof. As $FHD \subseteq HD$ we have:

$$\sup_{D \in HD} \mathrm{Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B) \geqslant \sup_{D \in FHD} \mathrm{Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B).$$

The converse (i.e., $\leqslant$ instead of $\geqslant$) holds because for any HD-scheduler D and any $\varepsilon > 0$ there is a fair HD-scheduler D' with

$$\mathrm{Pr}_{D'}(s, \overset{\leqslant t}{\rightsquigarrow} B) \geqslant \mathrm{Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B) - \varepsilon.$$

To construct D', select $k \in \mathbb{N}$ such that:

$$\left(\sum_{n=0}^{k} \psi(n) \cdot \mathbf{P}^n_{D,B} \cdot \underline{i}_B \right)(s) \geqslant \mathrm{Pr}_D(s, \overset{\leqslant t}{\rightsquigarrow} B) - \varepsilon.$$

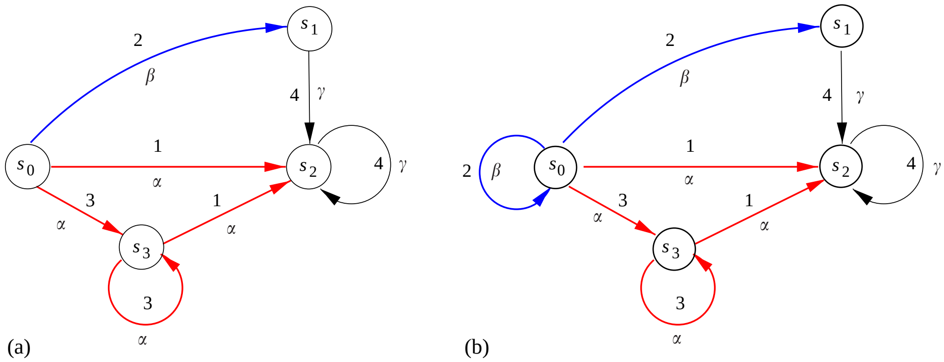

Fig. 3. An example illustrating why uniformization on CTMDPs is not obvious.

Then, we define D' as a fair HD-scheduler which agrees with D for all paths consisting of at most k transitions. (Note that such a fair extension is always possible.) Then,

$$\mathbf{P}^n_{D,B}(s,\sigma) = \mathbf{P}^n_{D',B}(s,\sigma) \quad \text{for all } s \in S, \sigma \in S^+ \text{ and } n \leqslant k.$$

Hence,

$$\begin{aligned}
Pr_{D'}(s, \overset{\leqslant t}{\rightsquigarrow} B) &\geqslant \left(\sum_{n=0}^{k} \psi(n)\cdot \mathbf{P}^n_{D',B}\cdot \underline{i}_B\right)(s) \\
&= \left(\sum_{n=0}^{k} \psi(n)\cdot \mathbf{P}^n_{D,B}\cdot \underline{i}_B\right)(s) \geqslant Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B) - \varepsilon. \qquad \square
\end{aligned}$$

5. The uniformization problem

Algorithm 1 assumes that the CTMDP under consideration is uniform. We now discuss the case in which the CTMDP is not uniform, i.e., the exit rates $E(s,\alpha)$ are not guaranteed to be identical for any state s and any $\alpha \in Act(s)$.

In the setting of CTMCs, uniformization [26] can be employed to transform a CTMC into a uniform one while keeping transient probabilities invariant. For CTMDPs, a similar recipe might be followed. However, a simple adaptation of the uniformization approach for CTMCs (as proposed, for instance, in [10,31]) to CTMDPs is not adequate for our purpose. The problem with such an approach is that the correspondence between schedulers on a uniform CTMDP $\mathcal{M}'$ and its original CTMDP $\mathcal{M}$ is lost. (A similar observation has been made by Beutler and Ross [11] when comparing MD- and MR-schedulers for computing time-average rewards.) This can be illustrated as follows. Applying “standard” uniformization to a CTMDP $\mathcal{M} = (S, Act, \mathbf{R})$ with $E \geqslant \max_{s\in S,\alpha\in Act} E(s,\alpha)$ would yield the CTMDP $unif(\mathcal{M}, E) = (S, Act, \mathbf{R}')$ with

$$\mathbf{R}'(s,\alpha,s') = \begin{cases} \mathbf{R}(s,\alpha,s') & \text{if } s \neq s', \\ \mathbf{R}(s,\alpha,s) + E - E(s,\alpha) & \text{if } s = s' \text{and } \alpha \in Act(s), \\ 0 & \text{otherwise.} \end{cases}$$

That is, each state s is equipped with a self-loop for each action $\alpha \in Act(s)$ if E exceeds the total exit rate to take an α-transition from s. Applying this recipe to the CTMDP $\mathcal{M}$ depicted in Fig. 3(a) for $E = 4$ results in the CTMDP $unif(\mathcal{M}, E)$ in Fig. 3(b). The latter has appeared in Fig. 1(a) already. It is not difficult to see that for any X-scheduler on $\mathcal{M}$ there exists a corresponding X-scheduler on $unif(\mathcal{M}, E)$, as any choice in $\mathcal{M}$ can be matched by the same choice in $unif(\mathcal{M}, E)$. The reverse, however, does not hold. For instance, the MD-scheduler $D_{\beta\alpha}$ on $unif(\mathcal{M}, E)$ discussed in Section 4 does not correspond to any MD-scheduler D on $\mathcal{M}$, since the self-loop in state s_0 in $unif(\mathcal{M}, E)$ cannot be mimicked by $\mathcal{M}$. Recall from Section 4 that $\Pr_{D_{\beta\alpha}}(s_0, \overset{\leqslant 0.5}{\rightsquigarrow} \{ s_2 \})$ is higher than the respective probabilities for D_α and D_β in $unif(\mathcal{M}, E)$. The latter in turn corresponds to the only relevant HD-scheduler on $\mathcal{M}$. As a consequence, the maximum probability (obtained for some MD-scheduler generated by Algorithm 1) to reach the set $\{ s_2 \}$ from state s_0 in 0.5 time units on $unif(\mathcal{M}, E)$ is higher than the probability for any HD-scheduler in $\mathcal{M}$.

6. Concluding remarks

This paper considered the problem of computing the maximum probability to reach a set of goal states within a given time bound in a uniform CTMDP. It is shown that truncated Markovian deterministic schedulers suffice for approximating a solution to this problem in an efficient manner for (time-abstract) history-dependent and Markovian schedulers, both deterministic and randomized ones. This does neither apply to timed history-dependent schedulers nor to Markovian stationary (i.e., simple) schedulers. The question whether SMR-schedulers may yield the same optimum (or a smaller optimum) is open.

Although all results in this paper have been presented for maximum probabilities, the same results can be obtained for minimum probabilities, i.e.,

$$\inf_{D \in X} \Pr_D(s, \overset{\leqslant t}{\rightsquigarrow} B)$$

up to some accuracy ε.[4] Instead of a greedy policy that maximizes the probability to reach the set of goal states in each step of the computation, the algorithm in this case minimizes this quantity in each step.

The presented numerical algorithm is remarkably efficient. Its worst-case time complexity is in $\mathcal{O}(E \cdot t \cdot N^2 \cdot M)$ where E is the unique exit rate of the uniform CTMDP, t is the time bound, N is the number of states, and M is the number of actions. Thus, compared to CTMCs, the increase in computational effort is linear in the number of actions in the CTMDP, i.e., the amount of nondeterminism, but no more than that. This is the best we can hope for, since the time complexity of computing the corresponding probability in a CTMC is in $\mathcal{O}(E \cdot t \cdot N^2)$ [6].

It is not obvious how to extend the presented results beyond uniform CTMDPs, because the basic concept of uniformization blurs the distinction between timed and time-abstract

[4] Only Theorem 8 does not hold when the supremum over all fair schedulers is replaced by the infimum over all fair schedulers. See [9] for a counterexample for DTMDPs.

schedulers. As yet, it is open whether a variation of uniformization can be used to reduce the timed reachability problem for general CTMDPs to that of uniform CTMDPs.

Acknowledgements

The authors thank Luca de Alfaro (University of California at Santa Cruz) for insightful discussions on the issue of uniformization in CTMDPs. David N. Jansen (MPI Saarbrücken) provided various helpful comments on a draft of this paper.

References

[1] Y. Abdeddaïm, E. Asarin, O. Maler, On optimal scheduling under uncertainty, in: Tools and Algorithms for the Construction and Analysis of Systems, Lecture Notes in Computer Science, Vol. 2619, Springer, Berlin, 2003, pp. 240–253.
[2] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis, Modelling with Generalized Stochastic Petri Nets, Wiley, New York, 1995.
[3] L. de Alfaro, Formal verification of probabilistic systems, Ph.D. Dissertation, Stanford University, 1997.
[4] L. de Alfaro, How to specify and verify the long-run average behavior of probabilistic systems, in: IEEE 13th Symp. on Logic in Computer Science, IEEE CS Press, Silver Spring, MD, 1998, pp. 174–183.
[5] A. Aziz, K. Sanwal, V. Singhal, R. Brayton, Model checking continuous time Markov chains, ACM Trans. Comput. Logic 1 (1) (2000) 162–170.
[6] C. Baier, B. Haverkort, H. Hermanns, J.-P. Katoen, Model-checking algorithms for continuous-time Markov chains, IEEE Trans. Software Engrg. 29 (6) (2003) 524–541.
[7] C. Baier, B. Haverkort, H. Hermanns, J.-P. Katoen, Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes, in: Tools and Algorithms for the Construction and Analysis of Systems, Lecture Notes in Computer Science, Vol. 2988, 2004, pp. 61–76.
[8] C. Baier, H. Hermanns, J.-P. Katoen, V. Wolf, Comparative branching-time semantics for Markov chains (extended abstract), in: Concurrency Theory, Lecture Notes in Computer Science, Vol. 2761, Springer, Berlin, 2003, pp. 492–507.
[9] C. Baier, M.Z. Kwiatkowska, Model checking for a probabilistic branching time logic with fairness, Distrib. Comput. 11 (1998) 125–155.
[10] D.P. Bertsekas, Dynamic Programming and Optimal Control. Vol. II, Athena Scientific, 1995.
[11] F. Beutler, K.W. Ross, Uniformization for semi-Markov decision processes under stationary policies, J. Appl. Probab. 24 (1987) 644–656.
[12] A. Bianco, L. de Alfaro, Model checking of probabilistic and nondeterministic systems, in: Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science, Vol. 1026, Springer, Berlin, 1995, 499–513.
[13] J.L. Bruno, P.J. Downey, G.N. Frederickson, Sequencing tasks with exponential service times to minimize the expected flow time or makespan, J. Assoc. Comput. Mach. 28 (1) (1981) 100–113.
[14] P. Buchholz, Exact and ordinary lumpability in finite Markov chains, J. Appl. Probab. 31 (1994) 59–75.
[15] P. Buchholz, J.-P. Katoen, P. Kemper, C. Tepper, Model-checking large structured Markov chains, J. Logic Algebraic Programming 56 (2003) 69–97.
[16] G. Chiola, S. Donatelli, G. Franceschinis, GSPNs versus SPNs: what is the actual role of immediate transitions?, in: Petri Nets and Performance Models 1991, IEEE CS Press, Silver Spring, MD, 1991.
[17] G. Ciardo, R. Zijal, Well-defined stochastic Petri nets, in: Proc. Fourth Internat. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'96), 1996, pp. 278–284.
[18] P.R. D'Argenio, B. Jeannet, H.E. Jensen, K.G. Larsen., Reduction and refinement strategies for probabilistic analysis, in: Process Algebra and Probabilistic Methods, Lecture Notes in Computer Science, Vol. 2399, Springer, Berlin, 2001, pp. 57–76.
[19] D.D. Deavours, W.H. Sanders, An efficient well-specified check, in: Petri Nets and Performance Models, PNPM'99, IEEE CS Press, Silver spring, MD, 1999, pp. 124–133.

[20] E.A. Feinberg, Continuous time discounted jump Markov decision processes: a discrete-event approach, 1998. http://www.ams.sunysb.edu/~feinberg/public/.
[22] B.L. Fox, P.W. Glynn, Computing Poisson probabilities, Comm. ACM 31 (4) (1988) 440–445.
[23] H. Hermanns, Interactive Markov Chains and the Quest for Quantified Quality, Lecture Notes in Computer Science, Vol. 2428, Springer, Berlin, 2002.
[25] H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, M. Siegle, A tool for model-checking Markov chains, Internat. J. Software Tools for Technology Transfer 4 (2) (2003) 153–172.
[26] A. Jensen, Markov chains as an aid in the study of Markov processes, Skand. Aktuarietidskrift 3 (1953) 87–91.
[27] J.-P. Katoen, M.Z. Kwiatkowska, G. Norman, D. Parker, Faster and symbolic CTMC model checking, in: Process Algebra and Probabilistic Methods, Lecture Notes in Computer Science, Vol. 2165, 2001, Springer, Berlin, pp. 23–38.
[29] V.G. Kulkarni, Modeling and Analysis of Stochastic Systems, Chapman & Hall, London, 1995.
[30] M.Z. Kwiatkowska, G. Norman, D. Parker, Probabilistic symbolic model checking with PRISM: a hybrid approach, in: Tools and Algorithms for the Construction and Analysis of Systems, Lecture Notes in Computer Science, Vol. 2280, 2002, pp. 52–66.
[31] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley, New York, 1994.
[32] Q. Qiu, M. Pedram, Dynamic power management based on continuous-time Markov decision processes, in: Design Automation Conference 99, ACM Press, New York, 1999, pp. 555–561.
[33] W.H. Sanders, J.F. Meyer, Reduced base model construction methods for stochastic activity networks, IEEE J. Selected Areas in Commun. 9 (1) (1991) 25–36.
[34] L. Sennot, Stochastic Dynamic Programming and the Control of Queueing Systems, Wiley, New York, 1999.