Rheinisch-Westfälische Technische Hochschule Aachen

Lehrstuhl für Informatik 2
Software Modeling and Verification
Prof. Dr. Ir. Joost-Pieter Katoen

Analysis and scheduler synthesis of time-bounded reachability in continuous-time Markov decision processes

von

Dennis Guck

Bachelorarbeit
im Fach Informatik

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Aachen, 3rd December 2009

Dennis Guck

Abstract

Continuous-time Markov decision processes (CTMDPs) are stochastic models in which probabilistic and nondeterministic choices co-exist. Lately, a discretization technique has been developed to compute time-bounded reachability probabilities in locally uniform CTMDPs, i.e. CTMDPs with state-wise constant sojourn-times. We extend the underlying value iteration algorithm, such that it computes an ϵ-optimal scheduler for time-bounded reachability probabilities in CTMDPs as a byproduct. Further, we give a proof of correctness that the scheduler is ϵ-optimal. The ϵ-optimality stands for an error $\epsilon > 0$, that is induced by the approximation algorithm, and can be specified a priori. As a second contribution, we extend the value iteration algorithm with an incremental computation for several time bounds and prove that we obey the error bound ϵ with this method.

Zusammenfassung

Zeitkontinuierliche Markov Entscheidungsprozesse (CTMDPs) sind Wahrscheinlichkeitsmodelle, in denen probabilistische sowie nichtdeterministische Entscheidungen existieren. Vor Kurzem wurde eine Diskretisierungsmethode vorgestellt, mit welcher zeitbeschränkte Erreichbarkeitswahrscheinlichkeiten für lokal uniforme CTMDPs berechnet werden können, welche zustandsweise eine konstante Verweildauer haben. In dieser Arbeit erweitern wir den zugrundeliegenden value iteration Algorithmus so, dass wir einen ϵ-optimalen Scheduler für zeitbeschräkte Erreichbarkeitswahrscheinlichkeiten in CTMDPs dabei generieren. Weiterhin führen wir einen Korrektheitsbeweis, welcher zeigt, dass der generierte Scheduler ϵ-optimal ist. Die ϵ-Optimalität beschreibt einen Fehler $\epsilon > 0$, der durch die approximations Technik zustande kommt, und von vornherein festgelegt werden kann. Darüber hinaus führen wir eine inkrementelle Berechnungsmethode in die value iteration ein, mit welcher wir verschiedene Zeitschranken innerhalb einer Berechnung erhalten. Zudem beweisen wir, dass die Ergebnisse die Fehlerschranke ϵ einhalten.

Contents

1 Introduction

Continuous-time Markov decision processes (CTMDPs) [3, 9, 10] are stochastic models that are used for modelling decision-making problems, and have been applied, among others, in communication engineering, control of epidemics and manufacturing processes. The advantage of CTMDPs is, that they allow a nondeterministic choice between transitions whose delay is determined by nonegative exponential distributions.

To account for the nondeterminism in CTMDPs, we need schedulers. Schedulers can also be interpreted as policies or strategies and have the task to resolve the nondeterministic choice. In previous solutions, the nondeterminism was solved by time-abstract schedulers and only for globally uniform CTMDPs [3]. In this thesis, we focus on time-dependent schedulers, which solve the problem for time-bounded reachability probabilities better than time-abstract schedulers [3]. Moreover, we focus on locally uniform CTMDPs [9, 10], which allows us to delay the choice of the scheduler until the state is left. This is the case, because the sojourn-time in locally uniform CTMDPs is independent of the choice of the scheduler.

For the computation of the maximum probability for time-bounded reachability in CTMDPs, we have to discretize the CTMDP into an approximately equivalent discrete-time Markov decision process (MDP). Therefore, we use the discretization which is presented and proved in [9]. Thus, we can approximate the time-bounded reachability probability up to a given error bound $\epsilon > 0$ by considering the CTMDPs induced discrete-time MDP.

The main focus of this thesis is on synthesizing an ϵ-optimal scheduler for time-bounded reachability probabilities in CTMDPs. Therefore, we use an adaption of the value iteration technique which allows us to compute the probability for time bounded reachability up to an error bound ϵ and synthesizes the ϵ-optimal scheduler.
Moreover, we introduce a technique to compute the maximum time-bounded reachability probability for several time bounds within one computation. Therefore, we can reduce the complexity to compute various time-bounded reachability probabilities for the same model.

We describe and prove all results for the maximum time-bounded reachability in CTMDPs. All results and proofs can easily be adapted to the minimum time-bounded reachability.

The thesis is organized as follows: In Chapter 2, we recall the necessary preliminaries that we need for our approach. We will introduce locally uniform CTMDPs, the scheduler classes and a discretization technique [9] to transform a CTMDP into an approximately equivalent discrete-time MDP. In Chapter 3 we describe the value iteration algorithm and also introduce an incremental probability computation. Moreover, we focus in this chapter on the synthesis of the ϵ-optimal scheduler and prove that it is correct. Finally, Chapter 4 demonstrates the applicability of our results and provides some examples.

2 Preliminaries

In this thesis, we focus on a probabilistic model that describes the behaviour of a system in continuous time. Further, we work on time-dependent schedulers. Therefore, this chapter introduces the basic models and schedulers that we use.

We start and describe continuous-time Markov decision processes, which are the basis of our thesis. Moreover, we discuss the restriction to local uniformity. Further, we introduce the scheduler classes. Finally, we recall Markov decision processes and a discretization technique for continuous-time Markov decision processes.

2.1 Continuous-time Markov decision processes

First of all, we recall the notion of continuous-time Markov decision processes (CTMDPs) which have been introduced in [3, 9, 10].

Definition 1 (Continuous-time Markov decision process) *A continuous time Markov decision process (CTMDP) is a tuple* $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ *where* $\mathcal{S} = \{s_0, s_1, \ldots\}$ *is a finite set of states and* $Act = \{\alpha, \beta, \ldots\}$ *a finite set of actions,* $\mathbf{R} : \mathcal{S} \times Act \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ *a three dimensional rate matrix and* $\nu \in Distr(\mathcal{S})$ *is the initial distribution.*

The CTMDP behaves as follows. An α-transition from s to s' is enabled iff $\mathbf{R}(s, \alpha, s') = \lambda$ and $\lambda > 0$, where λ denotes the exponential distribution for the transition's delay. We denote with $Act(s)$ the set of enabled actions for s such that

$$Act(s) = \{\alpha \in Act \mid \exists s' \in \mathcal{S}.\mathbf{R}(s, \alpha, s') > 0\} \tag{1}$$

Definition 2 (exit rate) *Let* $\mathcal{C}$ *be a CTMDP. Then the exit rate is* $\lambda(s, \alpha) = \sum_{s' \in S} \mathbf{R}(s, \alpha, s')$ *for* $s \in S$ *and* $\alpha \in Act$*. The maximum exit rate is denoted* $\lambda = \max_{s \in S, \alpha \in Act} \lambda(s, \alpha)$.

With a well-formed CTMDP $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$, which means that $Act(s) \neq \emptyset$ for all $s \in \mathcal{S}$, we obtain an embedded MDP $\mathcal{M} = (\mathcal{S}, Act, \mathbf{P}, \nu)$ where the time abstract transition probability is given by $\mathbf{P}(s, \alpha, s') = \frac{\mathbf{R}(s,\alpha,s')}{\lambda(s,\alpha)}$ for $\lambda(s, \alpha) > 0$ and 0, otherwise.

Example 1 *Let* $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ *be the CTMDP in Figure 1 with* $\mathcal{S} = \{s_0, s_1, s_2, s_3\}$ *and* $Act = \{\alpha, \beta, \gamma\}$*. Further* $\mathbf{R}$ *is given by the labeled arcs in Figure 1: For example* $\mathbf{R}(s_1, \alpha, s_3) = 0.5$*, and* $\nu = \{s_0 \mapsto 1\}$.

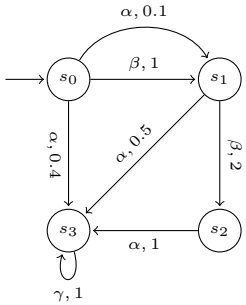


Figure 1: CTMDP for Example 1.

Consider the CTMDP in Example 1. There are two states, s_0 and s_1, with a nondeterministic choice. When we focus on state s_0 and choose action α, the two α-transitions will compete for execution and the sojourn-time is exponentially distributed with $\lambda(s_o, \alpha) = 0.5$. The probability to reach state s_3 is given by $\mathbf{P}(s_0, \alpha, s_3) = \frac{\mathbf{R}(s_0,\alpha,s_3)}{\lambda(s_0,\alpha)} = \frac{4}{5}$

Paths in CTMDPs are defined as follows. A finite path in a CTMDP $\mathcal{C}$ is a sequence of the form $\pi = s_0 \xrightarrow{t_0,\alpha_0} s_1 \xrightarrow{t_1,\alpha_1} \ldots \xrightarrow{t_{n-1},\alpha_{n-1}} s_n$ where $s_i \in \mathcal{S}, \alpha_i \in Act$ and $t_i \in \mathbb{R}_{\geq 0}$ for $i \leq n$, where n is the length of π denoted with $|\pi|$. The total time spent on π is denoted by $\Delta(\pi) = \sum_{k=0}^{n-1} t_k$, where t_k is the sojourn-time for state $\pi[k] = s_k$. Therefore, $\pi[k]$ denotes the k-th state on π and $\pi \downarrow = s_n$ is the last state on π. Let $Paths^*(\mathcal{C})$, $Paths^\omega(\mathcal{C})$ and $Paths(\mathcal{C})$ denote the set of finite, infinite and all paths of $\mathcal{C}$.

2.2 Local uniformity

In general, the exit rate of a state $s \in \mathcal{S}$ depends on the action $\alpha \in Act(s)$ which is chosen by the scheduler. For example, we have a CTMDP which is not locally uniform. Thus, there exists a state $s \in \mathcal{S}$ with $\lambda(s, \alpha) \neq \lambda(s, \beta)$ for $\alpha, \beta \in Act(s)$. That means, if the scheduler does not choose an action by entering state s and resolves the nondeterminism, it is not clear according to what exit rate the delay in state s is distributed. The motivation to consider local uniformity is, that in this setting we only have one distribution for the delay in state s, such that the scheduler can wait with the decision of choosing an action $\alpha \in Act(s)$ until state s is left.

Definition 3 (Local uniformity) *Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a CTMDP. $\mathcal{C}$ is locally uniform iff $\forall s \in \mathcal{S}. \forall \alpha, \beta \in Act(s). \lambda(s, \alpha) = \lambda(s, \beta)$.*

With locally uniform CTMDPs we achieve that the exit rates are state-wise constant. That means, the sojourn-time of any state $s \in \mathcal{S}$ does not depend on the action that is chosen in s. Therefore, we have solved the problem of choosing an action by entering a state. Note that from now on we denote the exit rate with $\lambda(s)$.

Example 2 *Consider the CTMDP $\mathcal{C}$ in Figure 2. The CTMDP is locally uniform as on the one hand, in state s_0 the exit rate under action α is $\lambda(s_0, \alpha) = \mathbf{R}(s_0, \alpha, s_0^\alpha) + \mathbf{R}(s_0, \alpha, s_1) + \mathbf{R}(s_0, \alpha, s_3) = 0.5 + 0.1 + 0.4 = 1$ which is equal to the exit rate for action β with $\lambda(s_0, \alpha) = \mathbf{R}(s_0, \alpha, s_1) = 1$. On the other hand, in state s_1 the exit rate under action α and β is $\lambda(s_1, \alpha) = 2 = \lambda(s_1, \beta)$ which is equal too.*

For our approach, we restrict to locally uniform CTMDPs.

As an approach to use also non-locally uniform CTMDPs, we show how we can transform a given CTMDP into a locally uniform CTMDP by recalling the definition in [10].

Definition 4 (Local uniformization) *Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a CTMDP. Then $\mathcal{C}' = (\mathcal{S}', Act, \mathbf{R}', \nu')$ is the locally uniform CTMDP induced by $\mathcal{C}$ with $\mu(s) = max\{\lambda(s, \alpha) | \alpha \in Act\}$, $\mathcal{S}' = \mathcal{S} \cup \mathcal{S}_{cp}$, $\mathcal{S}_{cp} = \{s^\alpha | \lambda(s, \alpha) < \mu(s)\}$,*

$$\mathbf{R}'(s, \alpha, s') = \begin{cases} \mathbf{R}(s, \alpha, s') & \text{if } s, s' \in \mathcal{S} \\ \mathbf{R}(t, \alpha, s') & \text{if } s' \in \mathcal{S} \wedge s = t^\alpha \\ \mu(s) - E(s, \alpha) & \text{if } s \in \mathcal{S} \wedge s' = s^\alpha \\ 0 & \text{otherwise} \end{cases}$$

and $\nu'(s) = \nu(s)$ if $s \in \mathcal{S}$ and otherwise, $\nu'(s) = 0$.

The local uniformization from a CTMDP to a locally uniform CTMDP is done for each state s separately. To demonstrate this, we show these for state s_0 in Example 1. The maximal exit rate is $\mu(s_0) = 1$ for action β. $\lambda(s_0, \alpha) = 0.5$ is less than $\mu(s_0)$. Now we introduce a copy state s_0^α and a new α-transition with rate $\mathbf{R}(s, \alpha, s_0^\alpha) = \mu(s_0) - \lambda(s_0, \alpha) = 0.5$. Therefore s_0^α is a deterministic state, because $Act(s_0^\alpha) = \{\alpha\}$. Furthermore, from s_0^α α-transitions lead to s_1 and s_3 with the same rate as from s_0. The complete locally uniform CTMDP is shown in Figure 2.
In [10], it was shown that local uniformization is measure preserving, but only for the standard GM-schedulers. We assume that it is also correct for late schedulers, which we use in our approach. Though we do not prove this here, we only work with locally uniform CTMDPs.

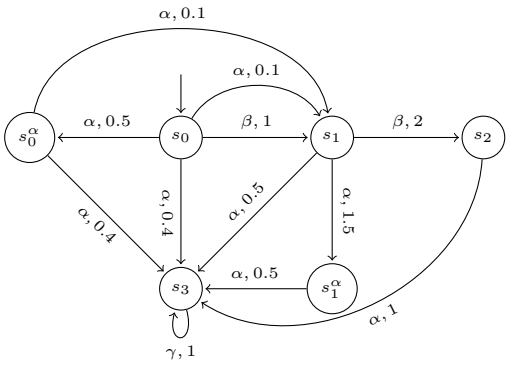


Figure 2: locally uniform CTMDP.

2.3 Markov decision processes

Before we describe how to discretize a CTMDP, we give a brief introduction of Markov decision processes (MDPs).

Definition 5 (Markov decision process)
A Markov decision process (MDP) is a tuple $\mathcal{M} = (\mathcal{S}, Act, \mathbf{P}, \nu)$ *where* $\mathcal{S} = \{s_0, s_1, \ldots\}$ *is a finite set of states and* $Act = \{\alpha, \beta, \ldots\}$ *a finite set of actions,* $\mathbf{P} : \mathcal{S} \times Act \times \mathcal{S} \to [0, 1]$ *the transition probability function such that* $\sum_{s' \in \mathcal{S}} \mathbf{P}(s, \alpha, s') \in \{0, 1\}$ *for all* $s \in \mathcal{S}$ *and* $\alpha \in Act$*. Further,* $\nu \in Distr(\mathcal{S})$ *is the initial distribution.*

An action α in state s is called enabled iff $\sum_{s' \in \mathcal{S}} \mathbf{P}(s, \alpha, s') = 1$. We denote the set of enabled actions for state s as in (1) with $Act(s)$ and $\sum_{s' \in \mathcal{S}} \mathbf{P}(s, \alpha, s') = 1$ instead of $\exists s' \in \mathcal{S}.\mathbf{R}(s, \alpha, s') > 0$. It is required that for every state $Act(s) \neq \emptyset$.
The main difference between an MDP and a CTMDP is the transition function. Instead of an exponential distribution for the transition's delay, the transition function $\mathbf{P}(s, \alpha, s')$ of an MDP describes the time-abstract probability to reach state s' from state s with an α-transition.

2.4 Scheduler

In general, scheduling is the process of deciding how to commit resources between several possible tasks. Schedulers are sometimes also referred to as policies or strategies. In our case, we need schedulers to resolve the nondeterminism in CTMDPs, which occurs in states with more than one enabled action. In the case of $Act(s) = \{\alpha_1, \ldots, \alpha_n\}$, a scheduler yields a distribution over $\alpha_1, \ldots, \alpha_n$, which resolves the nondeterminism. For example, let

$\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a CTMDP. A scheduler for $\mathcal{C}$ is a mapping $D : \mathcal{S} \to Act$ such that $D(s) \in Act(s)$ for all $s \in \mathcal{S}$. For MDPs, a scheduler works as follows: $D : Paths^* \to Distr(Act)$. That means we have no time information and the scheduler only chooses an action with information over the path. Besides that, a scheduler for a CTMDP needs the information over the time such that $D : Paths^* \times \mathbb{R}_{\geq 0} \to Distr(Act)$.

In the classic setting, a scheduler decides for an action $\alpha \in Act$ by entering the state $s \in \mathcal{S}$ [3]. The locally uniform CTMDP $\mathcal{C}$ allows us to define a generic measurable scheduler that uses the fact that the sojourn-time distribution is not dependent by the action that is chosen by the scheduler.

Definition 6 (Generic measurable scheduler) *A generic measurable scheduler for a locally uniform CTMDP $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ is a mapping $D : Paths^* \times \mathbb{R}_{\geq 0} \times \mathfrak{F}_{Act} \to [0, 1]$ where $\mathfrak{F}_{Act} = 2^{Act}$, $D(\pi, t, \cdot) \in Distr(Act(\pi \downarrow))$ for all $t \in \mathbb{R}_{\geq 0}$ and $\pi \in Paths^*$. D is a generic measurable scheduler (GM-scheduler) iff the function $D(\cdot, \cdot, A) : Paths^* \times \mathbb{R} \to [0, 1]$ are measurable for all $A \in \mathfrak{F}_{Act}$.*

We denote the probability measure over σ-fields with

$$Pr^n_{\nu,D} : \mathfrak{F}_{Paths^n} \to [0, 1] \tag{2}$$
$$Pr^\omega_{\nu,D} : \mathfrak{F}_{Paths^\omega} \to [0, 1] \tag{3}$$

where (2) is for set of finite paths and generated by measurable rectangles and (3) is for set of infinite paths and generated by the cylinder set construction. For more details about the probability measures and the GM-schedulers, we refer to [9].

Now we introduce total time positional schedulers. The decision of these schedulers are only based on the information of the current state and the total elapsed time. Let $D \in GM$ be a total time positional randomized scheduler (TTPR), such that for all $\pi_1, \pi_2 \in Paths^*$ where π_1 and π_2 end in the same state and the time spent on each paths are equal, it holds that D yields the same distribution.

Definition 7 (Total time positional scheduler) *Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a locally uniform CTMDP. A total time positional scheduler is a mapping $D : \mathcal{S} \times \mathbb{R}_{\geq 0} \to Distr(Act)$ where $\forall s \in \mathcal{S}. \forall t \in \mathbb{R}_{\geq 0}. D(s, t)(\alpha) > 0 \Rightarrow \alpha \in Act(s)$.*

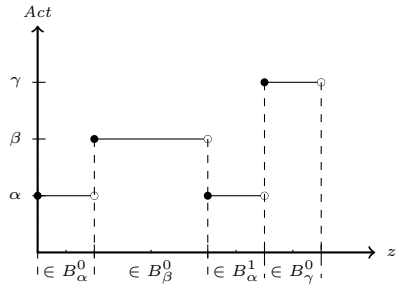


Figure 3: PCD schedule for a state s.

Hence, any TTPR scheduler D' is isomorphic to a mapping in Definition 7 where $D(s,t) = D'(\pi, t')$ for all paths $\pi \in Paths^*$ with $s = \pi\downarrow$ and $t \in \mathbb{R}_{\geq 0}$ where $\Delta(\pi) + t' = t$. Therefore, we specify the TTPR schedulers as functions in Definition 7. Note that the class of total time positional deterministic schedulers (TTPD) is a subclass of the TTPR schedulers, where $D(s,t)(\alpha) = 1$ for some $\alpha \in Act(s)$.

The problem for in approach with TTPD schedulers is, that they are continuous in their second argument. Thus, they may yield different actions for any point $t \in \mathbb{R}_{\geq 0}$ and therefore the set of TTPD schedulers is uncountable. For that purpose, we use piecewise constant schedulers (PCD).

Definition 8 (Piecewise constant TTPD scheduler)
Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a locally uniform CTMDP and $D : \mathcal{S} \times \mathbb{R}_{\geq 0} \to Act$ a TTPD scheduler. D is piecewise constant iff for all $\alpha \in Act(s)$ there exist disjoint intervals $B^0_{s,\alpha}, B^1_{s,\alpha}, \ldots \subseteq \mathbb{R}_{\geq 0}$ such that $D(s,t) = \alpha$ iff $t \in \bigcup_{i=0}^{\infty} B^i_{s,\alpha}$. A piecewise constant scheduler D is non-zero iff for all $z \in \mathbb{R}_{\geq 0}, s \in \mathcal{S}$ and $\alpha \in Act : |\{B^i_{s,\alpha} | \inf B^i_{s,\alpha} < z\}| < \infty$.

Figure 9 depicts a section of a PCD scheduler.

A subclass of the non-zero PCD scheduler are the τ-schedulers:

Definition 9 (τ-scheduler) *Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a locally uniform CTMDP, $D \in PCD$ a non-zero scheduler and $\tau \in \mathbb{R}_{>0}$. D is a τ-scheduler iff for all $s \in \mathcal{S}, k \in \mathbb{N}$*

$$\exists \alpha \in Act(s). \forall t \in [k\tau, (k+1)\tau). D(s,t) = \alpha.$$

These schedulers are constant on all intervals of length τ. Further, they are sufficient for our approach [9].

2.5 Discretization

CTMDPs work in continuous time, but to compute the maximum time-bounded reachability probability, we use an approach over discrete time. That means, we reduce the problem of computing the maximum time-bounded reachability in CTMDPs to the problem of computing the maximum step-bounded reachability in MDPs. Therefore, we recall the discretization which has been described in [9].

First of all, we discuss how to compute the maximum time-bounded reachability in CTMDPs:

Definition 10 (Maximum time-bounded reachability) .
Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a CTMDP, $G \subseteq \mathcal{S}$ a set of goal states, $s \in \mathcal{S}$ and $z \in \mathbb{R}_{\geq 0}$ a time bound. Then

$$p^{\mathcal{C}}_{max} : \mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1] : (s,z) \mapsto \sup_{D \in GM} Pr^{\omega}_{\nu_s, D}(\Diamond^{[0,z]} G)$$

is the maximum time-bounded reachability for the set G of goal states and time bound z.

In [9] it is shown that $p^{\mathcal{C}}_{max}$ is the least fixed point of a higher order operator Ω such that

$$p^{\mathcal{C}}_{max}(s,z) = \int_0^z \lambda(s) e^{-\lambda(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot p^{\mathcal{C}}_{max}(s, z-t) dt. \qquad (4)$$

Now we compute the discretization for a CTMDP according to [9] as follows:

Definition 11 (Discretization) *Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a CTMDP and let $\tau > 0$ be a step duration. The induced MDP $\mathcal{C}_\tau = (\mathcal{S}, Act, \mathbf{P}_\tau, \nu)$ is defined such that for all $s, s' \in \mathcal{S}$ and $\alpha \in Act(s)$:*

$$\mathbf{P}_\tau(s,\alpha,s') = \begin{cases} (1 - e^{-\lambda(s)\tau}) \cdot \mathbf{P}(s,\alpha,s') & if\ s \neq s' \\ (1 - e^{-\lambda(s)\tau}) \cdot \mathbf{P}(s,\alpha,s') + e^{-\lambda(s)\tau} & if\ s = s' \end{cases}$$

and for all $\alpha \notin Act$.$\mathbf{P}_\tau(s,\alpha,s') = 0$.

In the resulting MDP $\mathcal{C}_\tau$, each step corresponds to one time slice of length τ in the original CTMDP $\mathcal{C}$. The probability $\mathbf{P}_\tau(s,\alpha,s')$ of $\mathcal{C}_\tau$ for one step and $s \neq s'$ equals the probability in $\mathcal{C}$, that a transition occurs in τ time units and α is chosen. The other case is that $s = s'$. Therefore, the first summand of $\mathbf{P}_\tau(s,\alpha,s')$ denotes the probability to take a self loop back to s and the second summand denotes that no transition occurs within time slice τ.

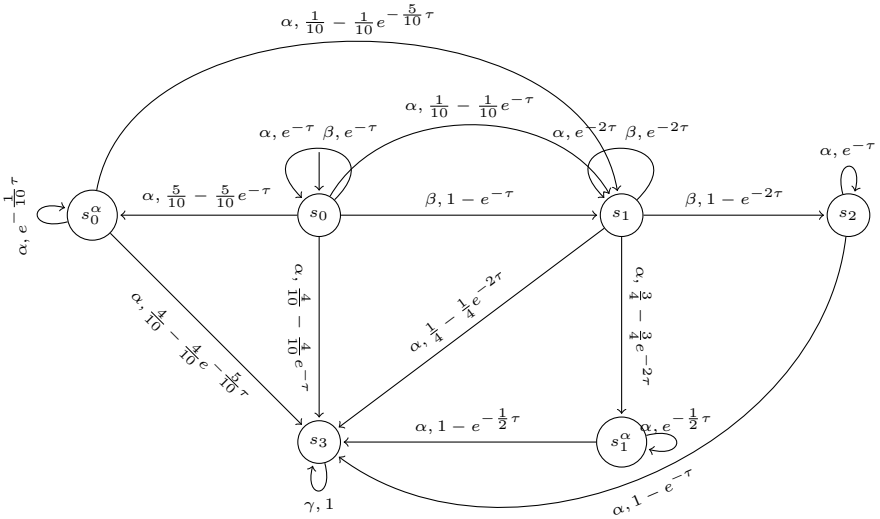


Figure 4: Induced MDP of Figure 2.

Example 3 *Let us consider the locally uniform CTMDP $\mathcal{C}$ in Figure 2 and the induced MDP $\mathcal{C}_\tau$ in Figure 4.*

The discretization from the locally uniform CTMDP $\mathcal{C}$ to the induced MDP $\mathcal{C}_\tau$ is done for each state s separately. We demonstrate this for state s_0. The exit rate for state s_0 is $\lambda(s_0) = 1$. First of all, we consider the action α. For the case that $s = s'$ we obtain $\mathbf{P}_\tau(s_0, \alpha, s_0) = (1 - e^{-\tau}) \cdot \mathbf{P}(s_0, \alpha, s_0) + e^{-\tau} = (1 - e^{-\tau}) \cdot 0 + e^{-\tau} = e^{-\tau}$. In the other case $s \neq s'$ we have to compute $\mathbf{P}_\tau(s_0, \alpha, s_0^\alpha), \mathbf{P}_\tau(s_0, \alpha, s_1)$ and $\mathbf{P}_\tau(s_0, \alpha, s_3)$. Therefore $\mathbf{P}_\tau(s_0, \alpha, s_0^\alpha) = (1 - e^{-\tau}) \cdot \mathbf{P}(s_0, \alpha, s_0^\alpha) = (1 - e^{-\tau}) \cdot \frac{0.5}{1} = \frac{5}{10} - \frac{5}{10}e^{-\tau}$. The other probabilities, also for action β, are computed analogously.

Definition 12 (Maximum step-bounded reachability)
Let $\mathcal{C}_\tau = (\mathcal{S}, Act, \mathbf{P}, \nu)$ be the induced MDP, $G \subseteq \mathcal{S}$ a set of goal states, $s \in \mathcal{S}$ and $k \in \mathbb{N}$ the number of time steps. Then $p_{max}^{\mathcal{C}_\tau}(s, k) = 1$ for $s \in G$ and $p_{max}^{\mathcal{C}_\tau}(s, 0) = 0$ for $s \notin G$. Further, for $s \notin G$ and $k > 0$:

$$p_{max}^{\mathcal{C}_\tau}(s, k) = \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} P_\tau(s, \alpha, s') \cdot p_{max}^{\mathcal{C}_\tau}(s', k-1)$$

is the maximum step-bounded reachability for the set G of goal states and k steps.

Lemma 1 (Error bound) *Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a CTMDP, $\mathcal{C}_\tau$ the induced MDP, $z \in \mathbb{R}_{\geq 0}$ a time bound and $\epsilon > 0$ an error bound such that there exists a $k \in \mathbb{N}$ with $k \geq \frac{(\lambda z)^2}{2\epsilon}$. Then it follows:*

$$p^{\mathcal{C}_\tau}_{max}(s,k) \leq p^{\mathcal{C}}_{max}(s,z) \leq p^{\mathcal{C}_\tau}_{max}(s,k) + \epsilon \tag{5}$$

The Inequality 5 is shown in [9]. By Lemma 1 and the computation of the maximum step-bounded reachability it follows, that we choose the number of time steps $k \in \mathbb{N}$, such that $k = \left\lceil \frac{(\lambda z)^2}{2\epsilon} \right\rceil$.

3 Analysis and scheduler synthesis

In this chapter, we introduce a value iteration technique which allows to compute the maximum probability for time bounded reachability in CTMDPs up to a given accuracy and to synthesize an ϵ-optimal scheduler. For the computation, we have to abstract from the timing-information and transform the CTMDP into an approximately equivalent discrete-time MDP with the discretization technique, introduced in Definition 11.

The organization of this chapter is as follows: In Section 3.1 we introduce the value iteration technique. Section 3.2 presents the possibility to compute the probability for various time-bounds within one computation. In Section 3.3 we prove that the generated scheduler is indeed ϵ-optimal. Section 3.4 shows some issues of the implementation.

3.1 Value iteration

In this section, we introduce a value iteration algorithm to compute the time-bounded reachability probability in locally uniform CTMDPs. A similar value iteration algorithm is presented in [4].

Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a locally uniform CTMDP, G a set of goal states and z a time bound. Then we choose a $k = \left\lceil \frac{(\lambda z)^2}{2\epsilon} \right\rceil$ according to the error bound ϵ. Thus we obtain the discretized MDP $\mathcal{C}_\tau$ with discretization step $\tau = \frac{z}{k}$.

The iterative approximation technique to calculate the maximum probability for $v(s) = Pr^{\omega}_{\nu_s, D_\tau}(\Diamond^{[0,z]} G)$ works as follows. We start with a probability vector v_0 such that

$$v_0(s) = 1 \text{ if } s \in G \text{ and } 0 \text{ otherwise.}$$

There, we set the probability of the goal set G to 1. For each iteration step $i > 0$ and state $s \notin G$, the probability vector v_i is obtained from v_{i-1}, such that

$$v_i(s) = \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}_\tau(s, \alpha, s') \cdot v_{i-1}(s'). \tag{6}$$

For $s \in G$ it holds that $v_i(s) = v_{i-1}(s)$. With this value iteration algorithm we use a backward search which starts in the goal states G. That means v_i is the probability vector that contains the probabilities to reach a goal state in i steps.
If we compare Equation 6 with Definition 12 it is apparent that $v_i(s) \equiv$

$p^{\mathcal{C}_\tau}_{max}(s,i)$. In both Equations we choose $\max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}_\tau(s,\alpha,s')$ and multiply this with the probability of the previous step.

Example 4 *First steps of the value iteration algorithm on Figure* (5).

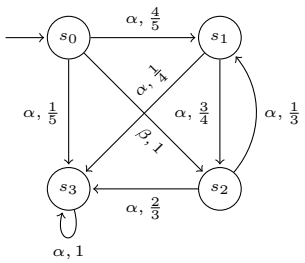


Figure 5: Example MDP for the value iteration algorithm.

We want to illustrate on an example how the value iteration algorithm works. For that purpose we use the MDP in Figure 5. Let $G = \{s_3\}$. It follows that $v_i(s_3) = 1$ for all $i \in \mathbb{N}$. For all other states we obtain

$$\begin{aligned} v_{i+1}(s_0) &= max\{1/5 \cdot v_i(s_3) + 4/5 \cdot v_i(s_1),\ v_i(s_2)\} \\ v_{i+1}(s_1) &= 1/4 \cdot v_i(s_3) + 3/4 \cdot v_i(s_2) \\ v_{i+1}(s_2) &= 2/3 \cdot v_i(s_3) + 1/3 \cdot v_i(s_1) \end{aligned}$$

The computation of the vectors $(v(s))_{s \in \mathcal{S}}$ yields

$$v_0 = (0,0,0,1) \qquad v_1 = (\tfrac{1}{5}, \tfrac{1}{4}, \tfrac{2}{3}, 1) \qquad v_2 = (\tfrac{2}{3}, \tfrac{3}{4}, \tfrac{3}{4}, 1)$$

$$v_3 = (\tfrac{4}{5}, \tfrac{13}{16}, \tfrac{11}{12}, 1) \quad v_4 = (\tfrac{11}{12}, \tfrac{15}{16}, \tfrac{15}{16}, 1) \quad v_5 = (\tfrac{19}{20}, \tfrac{61}{64}, \tfrac{47}{48}, 1)$$

$$\ldots$$

until we obtain for a given step number k the final result at vector v_k.

The complexity of our approach will be described in the following as in [9]. The size of the CTMDP $\mathcal{C}$ is given by $m = \sum_{s \in \mathcal{S}} \sum_{\alpha \in Act} |post(s,\alpha)|$ with $post(s,\alpha) = \{s' \in \mathcal{S} \mid \mathbf{R}(s,\alpha,s') > 0\}$. In the worst case, we obtain the discretized MDP $\mathcal{C}_\tau$ by adding a self-loop for each state $s \in \mathcal{S}$ and action $\alpha \in Act$. Hence the size of $\mathcal{C}_\tau$ is bounded by $2m$. For a given accuracy ϵ

we can derive the number k of value iteration steps. Let $\epsilon \geq \frac{(\lambda z)^2}{2k}$, thus the smallest k to guarantee ϵ is $k = \frac{(\lambda z)^2}{2\epsilon}$. In each value iteration step, the update of the vector v_i takes in the worst case time $2m$. Thus, the complexity of our value iteration algorithm is $\mathcal{O}(m \cdot (\lambda z)^2/\epsilon)$.

3.2 Incremental probability output in value iteration

In the most trivial approach, we compute the probabilities for different time-bounds by executing the value iteration algorithm for each time-bound separately. For each time bound z, this costs us in the worst case $\mathcal{O}(m \cdot (\lambda z)^2/\epsilon)$. Our goal is to reduce these costs. Therefore, we will calculate all probabilities during one value iteration.

Theorem 1 (incremental probability output) *Let $z \in \mathbb{R}_{>0}$ be the highest time bound, $k = \frac{(\lambda z)^2}{2\epsilon}$ the smallest amount of steps to guarantee ϵ, $\tau = \frac{z}{k}$ the discretization step and let $t \in \mathbb{R}_{>0}$ be such that $t < z \wedge t$ is a multiple of τ. Further, let $k_t = \frac{t}{\tau}$ such that*

$$|p^{\mathcal{C}_\tau}_{max}(s, k_t) - p^{\mathcal{C}}_{max}(s, t)| \leq \epsilon.$$

Proof. By Lemma 1 the smallest k to guarantee an error bound ϵ for a given time bound $z \in \mathbb{R}_{>0}$ is $k = \frac{(\lambda z)^2}{2\epsilon}$. For all t as above there exists a smallest time bound $i_t = \frac{(\lambda t)^2}{2\epsilon}$ to guarantee ϵ. Thus, it is sufficient to calculate the probability $p^{\mathcal{C}_\tau}_{max}(s, t)$ in i_t steps. That means, i_t are the necessary number of steps to obey the error bound ϵ and k_t are the number of steps we actually compute. Now we show that $k_t > i_t$.

$$k_t = \frac{t}{\tau} = \frac{t(\lambda z)^2}{2\epsilon z} = \frac{\lambda^2 z t}{2\epsilon} \geq \frac{(\lambda t)^2}{2\epsilon} = i_t.$$

Hence, it follows that we have at least i_t steps to calculate the probability $p^{\mathcal{C}_\tau}_{max}(s, t)$. Thus we obey the error bound ϵ by Lemma (1). Moreover, we have a positive side-effect on it. We achieve a smaller error bound and thereby a more precise calculation of the probability $p^{\mathcal{C}_\tau}_{max}(s, t)$. Let $i_t = \frac{(\lambda t)^2}{2\epsilon}$. Therefore $\epsilon = \frac{(\lambda t)^2}{2i_t}$. Hence $k_t > i_t$ and $\epsilon' = \frac{(\lambda t)^2}{2k_t}$ it follows $\epsilon > \epsilon'$.

□

We restrict the time bound t on multiples of τ because otherwise we get the probability for $t \in [k_t\tau; (k_t + 1)\tau]$ and not for the exact time-bound t. An

approach to permit every time bound $t \in \mathbb{Q}_{>0}$ is as follows. Let $\tau = \frac{z}{k}$. Now find a τ' that $\forall t \in \mathbb{Q}_{>0}.\exists k \in \mathbb{N}.t = k \cdot \tau' \wedge \exists k \in \mathbb{N}.z = k \cdot \tau' \wedge \tau \geq \tau'$. For $\tau' = \tau$ we achieve the same error bound ϵ and for $\tau' < \tau$ a lower error bound ϵ.
We have to restrict this approach on rational numbers $t \in \mathbb{Q}$, because the problem on real numbers is, that we can't express all $t \in \mathbb{R}$ as multiples of τ. Now we demonstrate two cases. The first one shows a good example, where we just gain our results in the previously calculated k steps. The second example shows a bad case, where we have to choose a τ' such that we have to calculate much more than k steps.

Example 5 (Wise choice) *Let $\mathcal{C}$ be a locally uniform CTMDP with $\lambda = 2$. The accuracy is given by $\epsilon = 10^{-4}$, the highest time bound by $z = 10$ and some smaller time bounds with $t_i = i$ for $i \in \{1, 2, \ldots, 9\}$.*

Therefore, we obtain for time bound z : $k = 2 \cdot 10^6, \tau = 5 \cdot 10^{-6}$ and for all t_i there exists a $k_i = \frac{i}{\tau} \in \mathbb{N}$. Thus we obtain all solutions in the original k steps for time bound z.

Example 6 (Bad choice) *Let $\mathcal{C}$ be a locally uniform CTMDP with $\lambda = 2$. The accuracy is given by $\epsilon = 10^{-4}$, the highest time bound by $z = 2.3$ and a smaller time bound by $t = 2\frac{1087}{10^8}$.*

Therefore, we obtain for time bound z : $k = 105800, \tau = 2.1739 \cdot 10^{-5}$, though for time bound t it follows that $k_t \notin \mathbb{N}$. Thus, we have to find a $\tau' \leq \tau$ such that there exists $k', k'_t \in \mathbb{N}$ with $z = k' \cdot \tau'$ and $t = k'_t \cdot \tau'$. For example, we have found $\tau' = 1.087 \cdot 10^{-5}$. Therefore, we obtain $k'_t = 184001$ and $k' = 211600$. If we compute the probability for time bound t separately, we need 80001 steps ($i_t = \frac{(\lambda t)^2}{2\epsilon} = 80000.8698$). Thus, with two separate calculations, we only have 185801 iteration steps instead of 211600.

Let us remark, that the incremental probability computation is not featured in standard model checking tools like the Probabilistic Symbol Model Checker (PRISM) [8] or the Markov Reward Model Checker (MRMC) [7]. Though the computation is faster with these tools, they have to start a new computation for every given time-bound. Further, PRISM does not support CTMDPs and MRMC supports only globally uniform CTMDPs. We discuss some more differences between MRMC and our approach later in Section 4.3.

3.3 Scheduler synthesis

Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a CTMDP, G a set of goal states, z a time bound, $\tau = \frac{z}{k}$ the length of one time interval and $\mathcal{C}_\tau$ the induced MDP. Now we get as a byproduct of the value iteration on $\mathcal{C}_\tau$ an ϵ-optimal scheduler for the set of goal states G in time bound z for each state $s \in \mathcal{S}$. According to Definition 12 we choose in any of the i value iteration steps and in each state $s \in \mathcal{S}$ an action $\alpha_{s,i} \in Act$ and get from that a step dependent scheduler in $\mathcal{C}_\tau$. This scheduler induces a τ-scheduler of the original CTMDP $\mathcal{C}$, denoted D^z_τ. This works as follows for all $s \in \mathcal{S}$ and $t \leq z$:

$$D^z_\tau(s,t) = \alpha_{s,i} \text{ if } t \in [(k-i)\tau, ((k-i)+1)\tau) \tag{7}$$

where $\alpha_{s,i}$ denotes the action $\alpha \in Act$, that is chosen for state $s \in \mathcal{S}$ at the i-th value iteration step. From Equation 7, it follows that D^z_τ is constant on the interval $[(k-i)\tau, ((k-i)+1)\tau)$.

Theorem 2 (ϵ-optimal scheduler) *Let D^z_τ be defined as in Equation 7 an ϵ-optimal scheduler for $\mathcal{C}$ w.r.t. the maximum time-bounded reachability probability. Then it holds that $|p^{\mathcal{C}}_{max}(s,z) - Pr^\omega_{\nu_s, D^z_\tau}(\Diamond^{[0,z]} G)| \leq \epsilon$ for all $s \in \mathcal{S}$.*

Proof. Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a locally uniform CTMDP, G a set of goal states and z a time bound. For some $\epsilon > 0$, let k be the number of steps necessary to satisfy $\epsilon > \frac{(\lambda z)^2}{2k}$. Let $\mathcal{C}_\tau$ be the induced MDP, with $\tau = \frac{z}{k}$. By definition, D^z_τ is an ϵ-optimal scheduler for $\mathcal{C}$ in time z. With Theorem 5 of [9] it is sufficient to show the following inequality:

$$p^{\mathcal{C}_\tau}_{max}(s,k) \leq Pr^\omega_{\nu_s, D_\tau}(\Diamond^{[0,z]} G) \leq p^{\mathcal{C}}_{max}(s,z). \tag{8}$$

First we show the upper bound. Definition 7 of [9] says that $p^{\mathcal{C}}_{max}$ is the maximum time bounded reachability for the set of goal states G and time-bound z, such that $p^{\mathcal{C}}_{max}(s,z)$ represents $\sup_{D \in GM} Pr^\omega_{\nu_s, D}(\Diamond^{[0,z]} G)$. Therefore it follows that $Pr^\omega_{\nu_s, D^z_\tau}(\Diamond^{[0,z]} G) \leq p^{\mathcal{C}}_{max}(s,z)$.

Now we have to show the lower bound of Inequality (8). For $s \in G$ it holds that $p^{\mathcal{C}_\tau}_{max}(s,k) = 1 = Pr^\omega_{\nu_s, D^z_\tau}(\Diamond^{[0,z]} G)$. For $s \notin G$ the probability $Pr^\omega_{\nu_s, D^z_\tau}(\Diamond^{[0,z]} G)$ can be computed by

$$Pr^\omega_{\nu_s, D^z_\tau}(\Diamond^{[0,z]} G) = \int_0^z \lambda(s) e^{-\lambda(s)t} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, D^z_\tau(s,t), s') \cdot Pr^\omega_{v_{s'}, D^{z-t}_\tau}(\Diamond^{[0,z-t]} G) dt$$

Now we divide the proof in three parts

1. Split $Pr^{\omega}_{\nu_s, D^z_\tau}(\Diamond^{[0,z]} G)$ in two parts at time $t = \tau$, such that we consider the first sub-interval $[0, \tau]$.
2. Approximate the probability for one step during $[0, \tau]$
3. Induction on k

For the first part we obtain following

$$Pr^{\omega}_{\nu_s, D^z_\tau}(\Diamond^{[0,z]} G) = \int_0^\tau \lambda(s) e^{-\lambda(s)t} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, D^z_\tau(s,t), s') \cdot Pr^{\omega}_{v_{s'}, D^{z-t}_\tau}(\Diamond^{[0,z-t]} G) dt$$
$$+ \int_\tau^z \lambda(s) e^{-\lambda(s)t} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, D^z_\tau(s,t), s') \cdot Pr^{\omega}_{v_{s'}, D^{z-t}_\tau}(\Diamond^{[0,z-t]} G) dt.$$

We denote the first summand with $A(s,z)$ and the second summand with $B(s,z)$. Now we shift the range of the integration in $B(s,z)$ by $(-\tau)$ and derive

$$\begin{aligned}
B(s,z) &= \int_0^{z-\tau} \lambda(s) e^{-\lambda(s)(t+\tau)} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, D^z_\tau(s, t+\tau), s') \\
&\quad \cdot Pr^{\omega}_{v_{s'}, D^{z-(t+\tau)}_\tau}(\Diamond^{[0,z-(t+\tau)]} G) dt \\
&= \int_0^{z-\tau} \lambda(s) e^{-\lambda(s)t} e^{-\lambda(s)\tau} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, D^z_\tau(s, t+\tau), s') \\
&\quad \cdot Pr^{\omega}_{v_{s'}, D^{z-(t+\tau)}_\tau}(\Diamond^{[0,z-(t+\tau)]} G) dt \\
&\overset{(*)}{=} e^{-\lambda(s)\tau} \int_0^{z-\tau} \lambda(s) e^{-\lambda(s)t} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, D^{z-\tau}_\tau(s,t), s') \\
&\quad \cdot Pr^{\omega}_{v_{s'}, D^{(z-\tau)-t}_\tau}(\Diamond^{[0,(z-\tau)-t]} G) dt \\
&= e^{-\lambda(s)\tau} \cdot Pr^{\omega}_{\nu_s, D^{z-\tau}_\tau}(\Diamond^{[0,z-\tau]} G)
\end{aligned}$$

where $e^{-\lambda(s)\tau}$ denotes the probability that within $[0, \tau]$ no transition occurs. More precisely, $B(s,z)$ denotes the maximum probability to reach from state s in time-bound z a set of goal states G and that no transition occurs in $[0, \tau]$. The step $(*)$ where we set $D^z_\tau(s, t+\tau) = D^{z-\tau}_\tau(s,t)$ can be explained

as follows: $t+\tau \leq z \Leftrightarrow t \leq z-\tau$. That means we gain the same information from scheduler D_τ^z at point $t+\tau$ as for scheduler $D_\tau^{z-\tau}$ at point t.
For $A(s,z)$ it is necessary that at least one transition occurs in $[0,\tau]$. We denote with $A_n(s,z)$ the maximum probability to reach G in time bound z with exactly n transitions during $[0,\tau]$. Let $\Delta_{[0,\tau]} : Paths^\omega \to \mathbb{N}$ be the number of transitions that occur during $[0,\tau]$. Therefore, $A_n(s,z) = Pr^\omega_{v_{s'},D_\tau^z}(\Diamond^{[0,z]}G \wedge \Delta_{[0,\tau]} = n)$.

Now we can split $A(s,z)$ into a disjoint union of $A_n(s,z)$.

$$\begin{aligned} A(s,z) &= Pr^\omega_{v_{s'},D_\tau^z}(\Diamond^{[0,z]}G \wedge \Delta_{[0,\tau]} \geq 1) \\ &= Pr^\omega_{v_{s'},D_\tau^z}\left(\biguplus_{n=1}^{\infty}(\Diamond^{[0,z]}G \wedge \Delta_{[0,\tau]} = n)\right) \\ &= \sum_{n=1}^{\infty}\left(Pr^\omega_{v_{s'},D_\tau^z}(\Diamond^{[0,z]}G \wedge \Delta_{[0,\tau]} = n)\right) \\ &= \sum_{n=1}^{\infty} A_n(s,z). \end{aligned}$$

The next step is to approximate the probability for one step during $[0,\tau]$. For $A_1(s,z)$ it holds

$$\begin{aligned} A_1(s,z) &= Pr^\omega_{v_{s'},D_\tau^z}(\Diamond^{[0,z]}G \wedge \Delta_{[0;\tau]} = 1) \\ &= \int_0^\tau \lambda(s)e^{-\lambda(s)t}\sum_{s'\in\mathcal{S}}\mathbf{P}(s,D_\tau^z(s,t),s')\cdot e^{-\lambda(s')(\tau-t)} \\ &\quad \cdot Pr^\omega_{v_{s'},D_\tau^{z-\tau}}(\Diamond^{[0,z-\tau]}G)dt. \end{aligned} \tag{9}$$

$A_1(s,z)$ describes the maximum probability to reach G within z time units, such that within time interval $[0,\tau]$ only one transition occurs. The behaviour of the integral in (9) is as follows. At a precise point t in time we leave state s and move to state s' where we stay at least for $(\tau-t)$ time units, which is described by the probability $e^{-\lambda(s')(\tau-t)}$. Finally we multiply with the maximum achievable probability to reach G in the remaining $(z-\tau)$ time units from s' under scheduler $D_\tau^{z-\tau}$.
Now we approximate $A_1(s,z)$ by bounding the probability $e^{-\lambda(s')(\tau-t)}$ from

above by 1 and denote this with $X_\tau(s,z)$

$$\begin{aligned} A_1(s,z) &= \int_0^\tau \lambda(s)e^{-\lambda(s)t}\sum_{s'\in\mathcal{S}}\mathbf{P}(s,D_\tau^z(s,t),s')\cdot e^{-\lambda(s')(\tau-t)} \\ &\quad \cdot Pr^\omega_{v_{s'},D_\tau^{z-\tau}}(\Diamond^{[0,z-\tau]}G)dt \\ &\leq \int_0^\tau \lambda(s)e^{-\lambda(s)t}\sum_{s'\in\mathcal{S}}\mathbf{P}(s,D_\tau^z(s,t),s')\cdot Pr^\omega_{v_{s'},D_\tau^{z-\tau}}(\Diamond^{[0,z-\tau]}G)dt \\ &= X_\tau(s,z). \end{aligned}$$

The next step is to show that

$$\begin{aligned} X_\tau(s,z) &= \int_0^\tau \lambda(s)e^{-\lambda(s)t}\sum_{s'\in\mathcal{S}}\mathbf{P}(s,D_\tau^z(s,t),s')\cdot Pr^\omega_{v_{s'},D_\tau^{z-\tau}}(\Diamond^{[0,z-\tau]}G)dt \\ &\leq \int_0^\tau \lambda(s)e^{-\lambda(s)t}\sum_{s'\in\mathcal{S}}\mathbf{P}(s,D_\tau^z(s,t),s')\cdot Pr^\omega_{v_{s'},D_\tau^{z-t}}(\Diamond^{[0,z-t]}G)dt \\ &= A(s,z). \end{aligned}$$

For this purpose we only have to show that

$$Pr^\omega_{v_{s'},D_\tau^{z-\tau}}(\Diamond^{[0,z-\tau]}G) \leq Pr^\omega_{v_{s'},D_\tau^{z-t}}(\Diamond^{[0,z-t]}G). \tag{10}$$

Therefore, we can use the property that $Pr^\omega_{v_{s'},D_\tau}(\Diamond^{[0,\cdot]}G)$ is monotonically increasing with increasing time-bounds for all $s'\in\mathcal{S}$. Thus, for $t\leq\tau$ we have shown Equation (10) from which follows that $X_\tau(s,z)\leq A(s,z)$.

Now, after these preparations, we can prove the lower bound $p^{\mathcal{C}_\tau}_{max}(s,k)\leq Pr^\omega_{\nu_s,D_\tau^z}(\Diamond^{[0,z]}G)$ by induction on k.
For $k=1$, we have $z=\tau$ and also $i=1$. Thus, with Equation 7 it follows that D_τ^τ is constant on $[0,\tau)$. If $s\in G$ then $p^{\mathcal{C}_\tau}_{max}(s,1)=1=Pr^\omega_{v_s,D_\tau^\tau}(\Diamond^{[0,\tau]}G)$. For $s\notin G$ we obtain

$$p^{\mathcal{C}_\tau}_{max}(s,1) = \max_{\alpha\in Act}\sum_{s'\in\mathcal{S}}P_\tau(s,\alpha,s')\cdot p^{\mathcal{C}_\tau}_{max}(s',0).$$

Since $p^{\mathcal{C}_\tau}_{max}(s,0) = 1$ if $s \in G$ and $p^{\mathcal{C}_\tau}_{max}(s,0) = 0$ otherwise, and with Definition (11) we obtain

$$\begin{aligned}
p^{\mathcal{C}_\tau}_{max}(s,1) &= \max_{\alpha \in Act}(1 - e^{-\lambda(s)\tau}) \cdot \mathbf{P}(s,\alpha,G) \\
&\overset{(7)}{=} (1 - e^{-\lambda(s)\tau}) \cdot \mathbf{P}(s,\alpha_{s,1},G) \\
&\overset{t\in[0,\tau)}{=} (1 - e^{-\lambda(s)\tau}) \cdot \mathbf{P}(s,D^\tau_\tau(s,t),G) \\
&= X_\tau(s,\tau) \\
&\leq A(s,\tau) \overset{(**)}{=} Pr^\omega_{v_s,D^\tau_\tau}(\Diamond^{[0,\tau]}G).
\end{aligned}$$

For step $(**)$ recall that $Pr^\omega_{v_s,D^z_\tau}(\Diamond^{[0,z]}G)$ is defined as $A(s,z) + B(s,z)$. Note that $B(s,\tau) = 0$ for $s \notin G$. Therefore it follows that $Pr^\omega_{v_s,D^\tau_\tau}(\Diamond^{[0,\tau]}G) = A(s,\tau) + B(s,\tau) = A(s,\tau) + 0 = A(s,\tau)$.
By induction hypothesis, it holds that

$$p^{\mathcal{C}_\tau}_{max}(s',k-1) \leq Pr^\omega_{v_{s'},D^{z-\tau}_\tau}(\Diamond^{[0,z-\tau]}G)$$

for all $s' \in \mathcal{S}$.
Now we can provide the induction step. Recall that $Pr^\omega_{v_s,D^z_\tau}(\Diamond^{[0,z]}G) = A(s,z) + B(s,z)$.

$$\begin{aligned}
Pr^\omega_{v_s,D^z_\tau}(\Diamond^{[0,z]}G) &\geq X_\tau(s,z) + B(s,z) \\
&= \int_0^\tau \lambda(s)e^{-\lambda(s)t} \sum_{s'\in\mathcal{S}} \mathbf{P}(s,D^z_\tau(s,t),s') \cdot Pr^\omega_{v_{s'},D^{z-\tau}_\tau}(\Diamond^{[0,z-\tau]}G)dt \\
&\quad + e^{-\lambda(s)\tau} \cdot Pr^\omega_{v_s,D^{z-\tau}_\tau}(\Diamond^{[0,z-\tau]}G) \\
&\overset{(***)}{=} \sum_{s'\in\mathcal{S}} \mathbf{P}(s,D^z_\tau(s,0),s') \cdot Pr^\omega_{v_{s'},D^{z-\tau}_\tau}(\Diamond^{[0,z-\tau]}G) \cdot \int_0^\tau \lambda(s)e^{-\lambda(s)t}dt \\
&\quad + e^{-\lambda(s)\tau} \cdot Pr^\omega_{v_s,D^{z-\tau}_\tau}(\Diamond^{[0,z-\tau]}G) \\
&= (1 - e^{-\lambda(s)\tau}) \cdot \sum_{s'\in\mathcal{S}} \mathbf{P}(s,D^z_\tau(s,0),s') \cdot Pr^\omega_{v_{s'},D^{z-\tau}_\tau}(\Diamond^{[0,z-\tau]}G) \\
&\quad + e^{-\lambda(s)\tau} \cdot Pr^\omega_{v_s,D^{z-\tau}_\tau}(\Diamond^{[0,z-\tau]}G) \\
&\overset{(11)}{=} \sum_{s\in\mathcal{S}} \mathbf{P}_\tau(s,D^z_\tau(s,0),s') \cdot Pr^\omega_{v_{s'},D^{z-\tau}_\tau}(\Diamond^{[0,z-\tau]}G) \\
&\overset{i.h.}{\geq} \sum_{s\in\mathcal{S}} \mathbf{P}_\tau(s,D^z_\tau(s,0),s') \cdot p^{\mathcal{C}_\tau}_{max}(s,k-1) \\
&= \sum_{s\in\mathcal{S}} \mathbf{P}_\tau(s,\alpha_{s,i},s') \cdot p^{\mathcal{C}_\tau}_{max}(s,k-1)
\end{aligned}$$

$$\stackrel{(7)}{=} \max_{\alpha \in Act} \sum_{s \in \mathcal{S}} \mathbf{P}_\tau(s, \alpha, s') \cdot p^{\mathcal{C}_\tau}_{max}(s, k-1)$$
$$= p^{\mathcal{C}_\tau}_{max}(s, k)$$

The step $(***)$ is correct, because the ϵ-optimal scheduler D^z_τ is constant on $t \in [0, \tau)$. Thus, we choose $t = 0$ and therefore D^z_τ is no longer dependent on t and can be extracted from the integral. Hence, the lower bound is shown. With Theorem 5 of [9], it follows that $p^{\mathcal{C}_\tau}_{max}(s, k)$ differ by at most ϵ of $p^{\mathcal{C}}_{max}(s, z)$. Thus, with Inequality (8), it follows $|p^{\mathcal{C}}_{max}(s, z) - Pr^{\omega}_{\nu_s, D^z_\tau}(\Diamond^{[0,z]} G)| \leq \epsilon$.

□

3.4 Implementation

In this section we discuss some practical issues. First of all, we have to reconsider how we obtain the time complexity $\mathcal{O}(m \cdot (\lambda z)^2 / 2\epsilon)$ from the value iteration algorithm, where $m = \sum_{s \in \mathcal{S}} \sum_{\alpha \in Act} |post(s, \alpha)|$ with $post(s, \alpha) = \{s' \in \mathcal{S} \mid \mathbf{P}(s, \alpha, s') > 0\}$. An earlier approach was to iterate over all states $s \in \mathcal{S}$ and actions $\alpha \in Act$ to update the probability vector v_i. Thus, we obtain a complexity with $m_{trivial} = |\mathbf{P}| := |\mathcal{S}|^2 + |Act|$. Therefore, we iterate over enabled and not enabled α-transitions. Thus, we have to define an efficient data structure to save $Act(s)$ for all $s \in \mathcal{S}$, such that we obtain m as above. For this purpose, we introduce a sparse matrix $\mathcal{A}$ which stores only non zero entries.

Definition 13 (Sparse matrix) *Let* $\mathbf{P}$ *the probability matrix of the MDP* $\mathcal{M}$. *We define the sparse matrix* $\mathcal{A}$ *as follows: Let* $succ, act, prob, pos$ *be arrays where* $succ, act$ *and* $prob$ *have length* m, *which denotes the size of the model* $\mathcal{M}$ *and* pos *the length* $|\mathcal{S}| + 1$ *such that*

$$\begin{aligned}
succ &= [s'_1, \ldots, s'_m] \text{ for all } s, s' \in \mathcal{S}.\alpha \in Act \text{ with } P(s, \alpha, s') > 0 \\
act &= [\alpha_1, \ldots, \alpha_m] \text{ for all } s, s' \in \mathcal{S}.\alpha \in Act \text{ with } P(s, \alpha, s') > 0 \\
prob &= [\mu_1, \ldots, \mu_m] \text{ for all } s, s' \in \mathcal{S}.\alpha \in Act \text{ with } P(s, \alpha, s') = \mu > 0 \\
pos &= [n_1, n_2, \ldots, n_{|\mathcal{S}|+1}] \text{ with } n_i \in \mathbb{N}
\end{aligned}$$

The *succ* array contains the α-successors of all states, the *act* array contains the actions of α-transitions of all states, the *prob* array contains the probabilities of α-transitions of all states, and the *pos* array contains the index for state $s \in \mathcal{S}$ where $pos[i]$ is the beginning and $pos[i+1] - 1$ the end for state

s in the array of *succ*, *act* and *prob*, where i denotes the number of state s.

As an example, we transform the following probability matrix into a sparse matrix. Let the row i denote states $s_i \in \mathcal{S}$ and the column j the successors $s'_j \in \mathcal{S}$. Let $\mathcal{M}$ be a MDP with $\mathcal{S} = \{s_0, s_1, s_2, s_3\}$, $Act = \{a, b\}$ and $\mathbf{P}$:

$$\mathbf{P}(\cdot, a, \cdot) = \begin{pmatrix} 0 & 0 & 0.25 & 0.75 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \mathbf{P}(\cdot, b, \cdot) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Therefore, we obtain following sparse matrix $\mathcal{A}$:

$$\begin{aligned} succ &= [s_2, s_3, s_1, s_2, s_2, s_3] \\ act &= [a, a, b, b, a, a] \\ prob &= [0.25, 0.75, 1, 1, 1, 1] \\ pos &= [0, 3, 4, 5, 6] \end{aligned}$$

Now we can interpret the sparse matrix $\mathcal{A}$ as follows. For state s_0 we have to look at the array entries $pos[0] = 0$ to $pos[1]-1 = 2$. Therefore, we obtain the following information: $\mathbf{P}(s_0, a, s_2) = 0.25$, $\mathbf{P}(s_0, a, s_3) = 0.75$, $\mathbf{P}(s_0, b, s_1) = 1$. For state s_1 to s_3 it works analogously.
The space we need to save a sparse matrix is $\mathcal{O}(3m + (|S| + 1))$. Moreover, the complexity to build the sparse matrix is given by $\mathcal{O}(|\mathbf{P}|)$.

4 Case Studies

In this chapter we discuss some case studies. First of all, we give an example where we synthesize an ϵ-optimal scheduler. The next case study is an application, such that we can compare our results with results in [5]. In the last case study, we give an example that demonstrates that the late schedulers, which we use in our approach, are more powerful than early schedulers [2, 3].

4.1 Synthesis of an ϵ-optimal scheduler

Now we provide some results for our approach. We use Example 7 to provide an ϵ-optimal scheduler, where we can see some action changes over the time.

Example 7 *Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be the locally uniform CTMDP depicted in Figure 6 with $\mathcal{S} = \{start, s_{600}, \ldots, s_0\}$, $Act = \{\alpha, \beta, \gamma\}$, $\forall i \in \{1, \ldots, 600\}$. $\mathbf{R}(s_i, \gamma, s_{i-1}) = \mu$ and $\mathbf{R}(s_0, \gamma, s_0) = \mu$, where $\mu = 100$, $\mathbf{R}(start, \cdot, \cdot)$ like in Figure 6 and $\nu = \{start \mapsto 1\}$.*

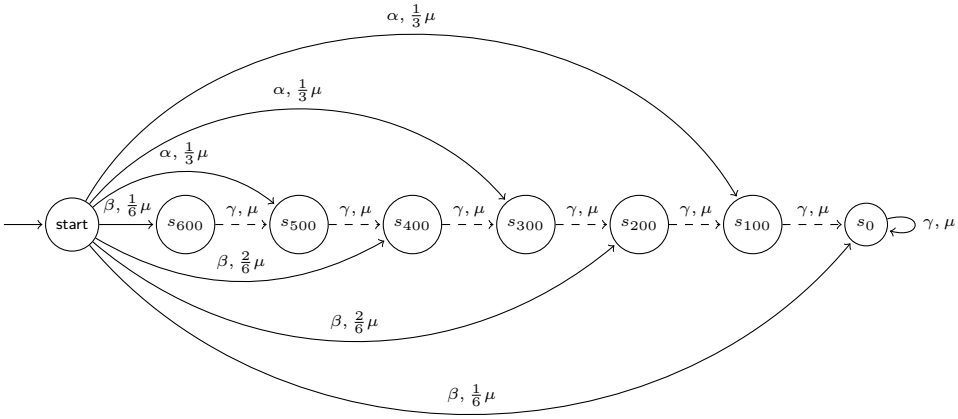


Figure 6: A locally uniform CTMDP with several Erlang-distributions.

In Figure 6 we can easily see that the CTMDP is locally uniform. The only nondeterministic state is *start*, and therefore $\lambda(start, \alpha) = \lambda(start, \beta) = 1$. Besides, the dashed arrows stand for γ-transitions with rate $\mu = 100$ between s_i and s_{i-1} for all $i \in \{1, \ldots, 600\}$. Therefore, we have several Erlang-distributions with an Erlang (k, λ) distributed delay, denoted with $Erl(k, \lambda)$. That means, we have k consecutive α-transitions where each transition has rate λ. The consequence is, that the time to move from a state $s_k \in \mathcal{S}$ for

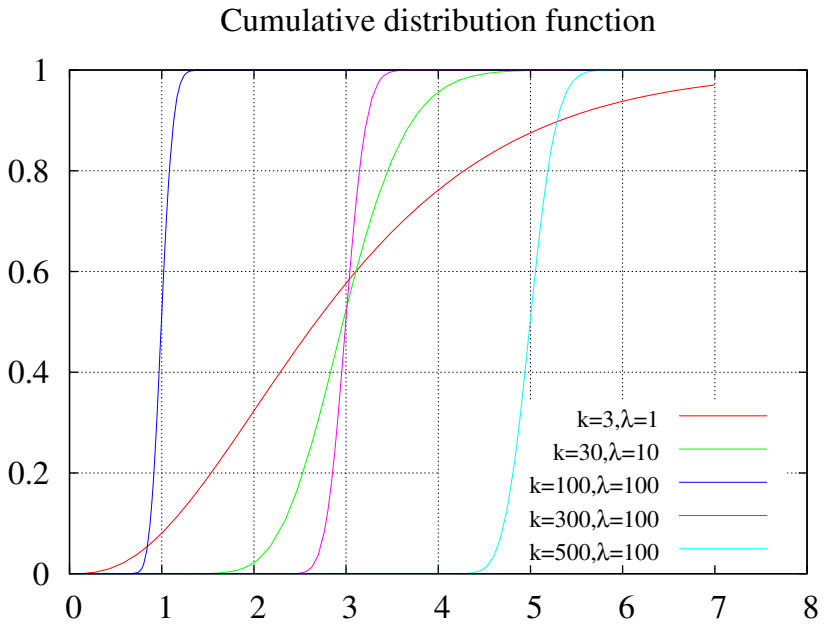
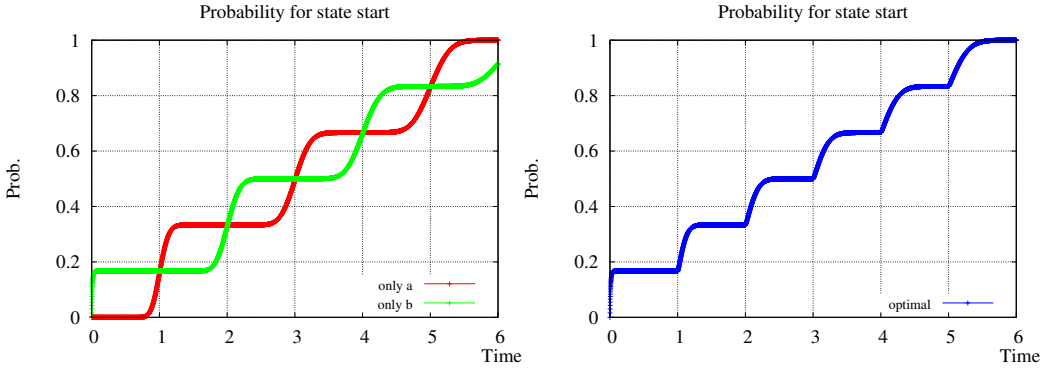


Figure 7: Several cumulative distribution functions of the Erlang-distribution.

$k \in \{1, \ldots, 600\}$ to the goal state s_0 is $\frac{k}{\lambda}$ with a variance of $\frac{k}{\lambda^2}$, where the variance describes how fast the distribution increases. Figure 7 depicts several Erlang distributions. With $Erl(100, 100), Erl(300, 100)$ and $Erl(500, 100)$ we describe the behaviour for states s_{100}, s_{300} and s_{500} to reach state s_0 in Example 7. When we compare $Erl(3, 1), Erl(30, 10)$ and $Erl(300, 100)$, the expected value $\frac{k}{\lambda} = 3$ is equal, but the gradient of the curves is different. This effect is caused by the variance $\frac{k}{\lambda^2}$, which is quadratically affected by rate λ. Therefore, with increasing rate, the variance decreases and the gradient of the distribution increases.

Now we compute the maximum time-bounded reachability probability for an error bound $\epsilon = 10^{-2}$ and time-bound $z = 6$. We choose such a large error bound to achieve a moderate time complexity. Due to the high rate $\lambda = 100$, we obtain a high time complexity $\mathcal{O}(m \cdot (\lambda z)^2/2\epsilon)$, because λ is squared. The result of the computation is presented in Figure 8b. When we compute the probability at the one hand with a scheduler that chooses only α and on the other hand only β in state *start*, we can see the difference of the maximum probability between these two actions in Figure 8a. In comparison of Figure 8a and 8b, we can see that with our approach the maximum probability for time-bounded reachability is at least the maximum probability between α and β. In Section 4.3 we will show an example where



(a) only α and only β (b) optimal

Figure 8: Maximum time-bounded reachability probabilities.

the maximum probability is even higher.

The generated curves in Figure 8 can be explained by the underlying Erlang-distributions. In the case of action α we have three different Erlang distributions. With rate $\frac{1}{3}$ we can go to state s_{100} with $Erl(100, 100)$, in state s_{300} with $Erl(300, 100)$ and in state s_{500} with $Erl(500, 100)$. Therefore, we can see the three probability increasing steps in Figure 8a for α, e.g. with $Erl(300, 100)$ we move with a high probability from s_{300} to s_0 in $\frac{300}{100} = 3$ time units. In the case of action β the explanation is analogous.

Furthermore, we gain the information within which time intervals we have to choose action α and action β in state *start* to maximize the probability to reach G within time-bound $z = 6$. This information is given in Figure 9, which depicts the ϵ-optimal scheduler for state *start*. The synthesis of the ϵ-optimal scheduler was made during the value iteration. That means, we obtain within one computation the maximum probability and the ϵ-optimal scheduler for time-bounded reachability.

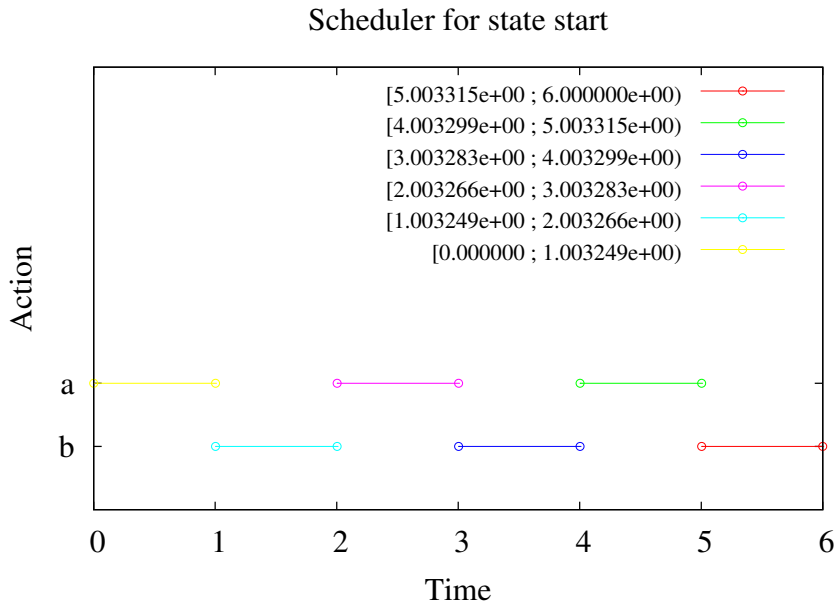


Figure 9: Scheduler for state *start* in Example 7.

4.2 The stochastic job scheduling problem

In this section, we demonstrate some results of our approach according to the stochastic job scheduling problem (sJSP) [5]. In their paper, the authors consider the problem of minimizing the expected makespan and the expected flow time on more than one identical processor for a finite set of independent stochastic jobs under a preemptive scheduling policy. They prove that a policy with a shortest expected processing time first (SEPT) strategy minimize the expected flow time for a sJSP and a policy with a longest expected time (LEPT) strategy minimize the expected makespan.

In our case, we consider the time-bounded reachability instead of the expected completion time. We will show that the ϵ-optimal scheduler, that maximizes the probability for time-bounded reachability, obeys the SEPT strategy and the ϵ-optimal scheduler, that minimizes the probability for time-bounded reachability, obeys the LEPT strategy.

The sJSP is defined as follows. An instance of the sJSP is given by (m, n, μ) where $m \geq 2$ is the number of processors, n the number of jobs identified by $J = \{1, \ldots, n\}$ and $\mu_i \in \mathbb{R}_{\geq 0}$ the rate of the exponential distribution of the processing time of job $i \in J$. Further a configuration is given by a tuple (R, W) where $R, W \subseteq J$ are the sets of running and waiting jobs. We use the modelling of the sJSP as an CTMDP in the form as it is shown in [9]:

Let $\mathcal{C}$ be a locally uniform CTMDP that describes a $(2, 4, \mu)$ sJSP with initial distribution (R, W) and the rates $\mu_1 = 0.2,\ \mu_2 = 0.8,\ \mu_3 = 0.4,\ \mu_4 = 0.6$.

Now we compute with our approach the maximum probability for time-bounded reachability with error bound $\epsilon = 10^{-4}$ and time-bound $z = 15$ for each possible start configuration (R, W). In Figure 10, we present the scheduler choices that correspond to the maximum probability for time-bounded reachability. The actions can be read as follows: An action $\alpha(R'_i; R'_j)$ with $R'_i, R'_j \subseteq J$ and $i, j \in J$ means, that if we chose $\alpha(R'_i; R'_j)$ we reach a state with $R = R'_i$ if process i finishes first and a state with $R = R'_j$ if process j finishes first. For example we choose $\alpha(\{14\}_3; \{13\}_4)$ for $(R, W) = (\{3, 4\}, \{1, 2\})$. Therefore, we reach state $(\{1, 4\}, \{2\})$ if job 3 finishes first and state $(\{1, 3\}, \{2\})$ if job 4 finishes first. With the given sJSP, we only have nondeterministic choices in the initial state. In Figure 10 we present in every table the scheduler choice of the initial state and the successor states with their unique action.

Now we want to analyze the chosen action of the different initial states (R, W). We start with $(R, W) = (\{1, 2\}, \{3, 4\})$. Therefore, the scheduler yields $\alpha(\{34\}_1; \{13\}_2)$. That means, we reach state $(\{3, 4\}, \{2\})$ if job 1 finishes first and state $(\{1, 3\}, \{4\})$ if job 2 finishes first. Therefore, we gain in the first case a rate of $\mu_1 + \mu_3 = 0.6$ and in the second case of $\mu_3 + \mu_4 = 1$, which describes in both cases the shortest processing time. If we consider the waiting processes of the successor states, there is in every case the job with the highest rate. Thus, in every case the scheduler has chosen the action which leads to successor states with the smallest possible processing time for R.

In the case for the ϵ-optimal scheduler that minimizes the probability for time-bounded reachability and the LEPT strategy, the results are analogous. They are presented in Figure 11.

(R,W)	Action
$(\{1,2\},\{3,4\})$	$\alpha(\{34\}_1;\{13\}_2)$
$(\{1,3\},\{4\})$	$\alpha(\{34\}_1;\{14\}_3)$
$(\{3,4\},\{2\})$	$\alpha(\{24\}_3;\{23\}_4)$

(R,W)	Action
$(\{1,3\},\{2,4\})$	$\alpha(\{34\}_1;\{14\}_3)$
$(\{1,4\},\{2\})$	$\alpha(\{24\}_1;\{12\}_4)$
$(\{3,4\},\{2\})$	$\alpha(\{24\}_3;\{23\}_4)$

(R,W)	Action
$(\{1,4\},\{2,3\})$	$\alpha(\{34\}_1;\{13\}_4)$
$(\{1,3\},\{2\})$	$\alpha(\{23\}_1;\{12\}_3)$
$(\{3,4\},\{2\})$	$\alpha(\{24\}_3;\{23\}_4)$

(R,W)	Action
$(\{2,3\},\{1,4\})$	$\alpha(\{13\}_2;\{14\}_3)$
$(\{1,3\},\{4\})$	$\alpha(\{34\}_1;\{14\}_3)$
$(\{1,4\},\{2\})$	$\alpha(\{24\}_1;\{12\}_4)$

(R,W)	Action
$(\{2,4\},\{1,3\})$	$\alpha(\{13\}_2;\{13\}_4)$
$(\{1,3\},\{2\})$	$\alpha(\{23\}_1;\{12\}_3)$
$(\{1,3\},\{4\})$	$\alpha(\{34\}_1;\{14\}_3)$

(R,W)	Action
$(\{3,4\},\{1,2\})$	$\alpha(\{14\}_3;\{13\}_4)$
$(\{1,3\},\{2\})$	$\alpha(\{23\}_1;\{12\}_3)$
$(\{1,4\},\{2\})$	$\alpha(\{24\}_1;\{12\}_4)$

Figure 10: Scheduler for maximum probability.

(R,W)	Action
$(\{1,2\},\{3,4\})$	$\alpha(\{24\}_1;\{34\}_2)$
$(\{2,4\},\{3\})$	$\alpha(\{34\}_2;\{23\}_4)$
$(\{3,4\},\{1\})$	$\alpha(\{24\}_3;\{23\}_4)$

(R,W)	Action
$(\{1,3\},\{2,4\})$	$\alpha(\{24\}_1;\{24\}_3)$
$(\{2,4\},\{1\})$	$\alpha(\{14\}_2;\{12\}_4)$
$(\{2,4\},\{3\})$	$\alpha(\{34\}_2;\{23\}_4)$

(R,W)	Action
$(\{1,4\},\{2,3\})$	$\alpha(\{24\}_1;\{23\}_4)$
$(\{2,4\},\{3\})$	$\alpha(\{34\}_2;\{12\}_4)$
$(\{2,3\},\{1\})$	$\alpha(\{13\}_2;\{12\}_3)$

(R,W)	Action
$(\{2,3\},\{1,4\})$	$\alpha(\{34\}_2;\{24\}_3)$
$(\{3,4\},\{1\})$	$\alpha(\{24\}_3;\{23\}_4)$
$(\{2,4\},\{1\})$	$\alpha(\{14\}_2;\{12\}_4)$

(R,W)	Action
$(\{2,4\},\{1,3\})$	$\alpha(\{34\}_2;\{23\}_4)$
$(\{3,4\},\{1\})$	$\alpha(\{24\}_3;\{23\}_4)$
$(\{2,3\},\{1\})$	$\alpha(\{13\}_2;\{12\}_3)$

(R,W)	Action
$(\{3,4\},\{1,2\})$	$\alpha(\{24\}_3;\{23\}_4)$
$(\{2,4\},\{1\})$	$\alpha(\{14\}_2;\{12\}_4)$
$(\{2,3\},\{1\})$	$\alpha(\{13\}_2;\{12\}_3)$

Figure 11: Scheduler for minimum probability.

4.3 Early scheduler vs. late scheduler

Now we discuss the difference between early and late schedulers. The schedulers we have worked with in this thesis and which we have defined in the preliminaries are denoted as late schedulers. With early schedulers, we denote schedulers which are used in [2, 3]. These schedulers choose an action directly upon entering the current state. Therefore, the sojourn time distribution of the current state depends on the choice of the scheduler. On the contrary, with late schedulers we can delay the decision of choosing an action until the current state is left. This is a result of local uniformity, where the sojourn time distribution is independent of the choice of the scheduler.

To analyze the behaviour of early schedulers, we use the Markov Reward Model Checker (MRMC) [7]. With the MRMC tool, we can compute the maximal probabilities for time-bounded reachability in globally uniform CTMDPs. Those are a subclass of locally uniform CTMDPs. More precisely, the class of locally uniform CTMDPs includes all CTMDPs where the exit rate is state-wise constant. The class of globally uniform CTMDPs has the restriction that the exit rate is constant for all states such that $\forall s \in \mathcal{S}.\lambda(s) = \lambda$. Further, the algorithm works on time-abstract schedulers.
Let us recall some definitions. Let $Paths^*_{abs}$ be a time abstract finite path such that $\pi_{abs} = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} s_n$ where $s_i \in \mathcal{S}$ and $\alpha_i \in Act$ for $i \leq n$, where n is the length of π_{abs} denoted with $|\pi_{abs}|$. Further, let $\pi_{abs} \downarrow = s_n$ be denoted the last state of path π_{abs}. A time-abstract history-depended deterministic (TAHD) scheduler is defined as

$$D : Paths^*_{abs} \to Act \text{ such that } D(\pi_{abs}) \in Act(\pi_{abs} \downarrow).$$

Therefore, TAHD schedulers decide based on the sequence of states and actions in π_{abs}. For the computation of the maximum probability for time-bounded reachability, it is sufficient to use Markovian deterministic schedulers (MD) [3]. These schedulers can be described as step-dependent schedulers, such that

$$D : \mathcal{S} \times \mathbb{N} \to Act \text{ such that } D(s, n) \in Act(s),$$

where the second argument describes the number of taken steps. MD schedulers are a subclass of TAHD scheduler, such that every MD scheduler can be described as a TAHD scheduler by ignoring everything from the history expect its length.

To analyze the behaviour of late schedulers, we use our approach of the value iteration algorithm. Therefore, we use a subclass of time-dependent deterministic scheduler, viz. TTPD scheduler. Those are defined in Definition 7 according to:

$$D : \mathcal{S} \times \mathbb{R}_{\geq 0} \to Act \text{ such that } D(s,t) \in Act(s).$$

Further, it is sufficient to use τ-scheduler (Definition 9) [9].

For our comparison we will use a globally uniform CTMDP that has one state with a nondeterministic choice.

Example 8 *Let $\mathcal{C} = (\mathcal{S}, Act, \mathbf{R}, \nu)$ be a globally uniform CTMDP with $\mathcal{S} = \{s_0, s_1, s_2, s_3\}$, $Act = \{\alpha, \beta\}, \mathbf{R}$ given by the labeled arcs in Figure 12 and $\nu = \{s_0\}$. The set of goal states is given by $G = \{s_2\}$, the error bound by $\epsilon = 10^{-6}$ and time bounds $z_1 = 0.1,\ z_2 = 0.2, \ldots,\ z_9 = 0.9$ and $z_{10} = 1$.*

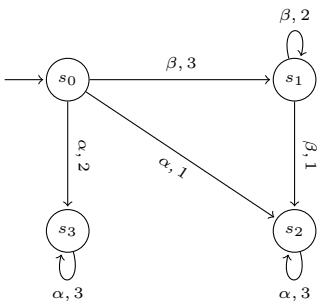


Figure 12: Globally uniform CTMDP for Example 8.

It is apparent that the CTMDP in Example 8 is globally uniform. For all states $s \in \mathcal{S}$ it holds that $\lambda(s) = 3$. Furthermore, state s_0 has a nondeterministic choice between action α and β.
Besides of the early and late schedulers, we also compute the maximum probability for time-bounded reachability in Example 8 for an α-scheduler and a β-scheduler. Those scheduler are defined for $Act = \{\alpha, \beta\}$ as follows:

α-scheduler
$D : \mathcal{S} \to Act$ such that $D(s) = \alpha$ if $\alpha \in Act(s)$ otherwise $D(s) = \beta$.

β-scheduler
$D : \mathcal{S} \to Act$ such that $D(s) = \beta$ if $\beta \in Act(s)$ otherwise $D(s) = \alpha$.

Time	Late	Early	α	β
0.1	0.0863939	0.0863939	0.0863939	0.0131529
0.2	0.1503961	0.1503961	0.1503961	0.0463096
0.3	0.1978101	0.1978101	0.1978101	0.0920573
0.4	0.2329352	0.2329353	0.2329352	0.1451169
0.5	0.2668487	0.2589566	0.2589566	0.2017691
0.6	0.3076441	0.2782337	0.2782337	0.2594319
0.7	0.3520667	0.3163503	0.2925145	0.3163502
0.8	0.3978249	0.3713655	0.3030940	0.3713654
0.9	0.4433498	0.4237483	0.3109315	0.4237482
1.0	0.4875955	0.4730744	0.3167376	0.4730742

Figure 13: Probabilities for time-bounded reachability in Example 8.

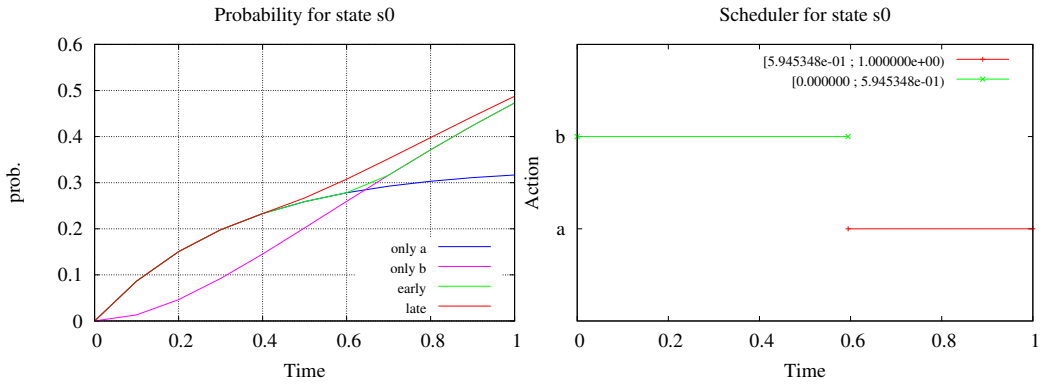
We want to compute the maximum probability to reach state s_2 from state s_0 in time-bounds $z \in \{z_1, \ldots, z_{10}\}$.

First of all, we compute the maximum probability for time-bounded reachability in Example 8 with the MRMC tool. Therefore, we have to start for every time- bound $z \in \{z_1, \ldots, z_{10}\}$ a new computation, where every computation needs less than a millisecond. The results are presented in the Early column of the probability table in Figure 13.

The next step is the computation with our approach. Therefore, we only have to start one computation according to Theorem 1 of the incremental probability output. Regardless, that we can compute the probabilities in one computation, we need ~ 28 seconds to compute all probabilities. The results are presented in the Late column of the probability table in Figure 13.

Moreover, we also have presented the probabilities of the α- and β-scheduler in the probability table in Figure 13 at column α and β.

Now we can compare the results that we have gained with the computations utilizing the different schedulers. A visualisation of the probabilities is given in Figure 14a. First of all, we can observe that the maximum probability grows if we change the decision of the action in state s_0 at one point. That means, if we have enough time left, the probability to reach a goal state is higher if we take action β and otherwise if we take α. Further, we can observe that in time interval $[0.4, 0.5]$ the difference between the maximum probability, based on late and early schedulers, starts to grow. Moreover, the early scheduler follows mostly the maximum of the α- and β-scheduler. In Figure 14b we can see at what point the late scheduler changes its decision between action α and β.



(a) Probabilities (b) ϵ-optimal scheduler

Figure 14: Visualisation of time-bounded reachability in Example 8.

On the basis of these observations, we have provided an example for the claim that late schedulers are more powerful than early schedulers:

$$\sup_{D \in late} Pr^{\omega}_{v_s,D}(\Diamond^{[0,z]} G) > \sup_{D \in early} Pr^{\omega}_{v_s,D}(\Diamond^{[0,z]} G).$$

This is also proven in [3], where the authors say that time-dependent schedulers are more powerful than time-abstract schedulers.

5 Conclusion

In this thesis we have solved the problem to synthesize an ϵ-optimal scheduler for time-bounded reachability probabilities in locally uniform CTMDPs. An advantage of our approach is, that we can bound the error for the approximation algorithm which computes the ϵ-optimal scheduler. That means, that we can specify the maximal error $\epsilon > 0$ a priori. Further, we have proven that the obtained scheduler is ϵ-optimal. Moreover, we have extended the value iteration algorithm with an incremental computation, such that we obtain several time-bounded reachability probabilities within one computation, where all probabilities obey the given error bound ϵ. This feature is not yet available in tools like PRISM [8] and MRMC [7]. In the case of the time complexity $\mathcal{O}(m \cdot (\lambda z)^2/2\epsilon)$ of our approach, we can save time with the incremental computation.
For future research, since the schedulers are limited to locally uniform CTMDPs, it is interesting to find a way to define corresponding schedulers for arbitrary CTMDPs. Further, in case of the time complexity of our value iteration algorithm, it is also interesting to reduce its complexity. For example, by adaption of the reduction techniques presented in [6], such that the value iteration algorithm can be executed more efficiently.

References

[1] C. Baier, B. R. H. M. Haverkort, H. Hermanns, and J.-P. Katoen. Reachability in continuous-time Markov reward decision processes. In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata: History and Perspectives, Aachen, Germany*, volume 2 of *Texts in Logic and Games*, pages 53–71, Amsterdam, February 2008. Amsterdam University Press.

[2] C. Baier and H. Hermanns. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003.

[3] C. Baier, H. Hermanns, J.-P. Katoen, and B. R. Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theoretical Computer Science*, 345(1):2–26, 2005.

[4] C. Baier and J.-P. Katoen. *Principles of Model Checking*. The MIT Press, May 2008.

[5] J. Bruno, P. Downey, and G. N. Frederickson. Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *Journal of the ACM*, 28(1):100–113, 1981.

[6] F. Ciesinski, C. Baier, M. Grosser, and J. Klein. Reduction techniques for model checking markov decision processes. In *Quantitative Evaluation of Systems, 2008. QEST '08. Fifth International Conference on*, pages 45–54, Sept. 2008.

[7] J.-P. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, and D. N. Jansen. The Ins and Outs of The Probabilistic Model Checker MRMC. In *Quantitative Evaluation of Systems (QEST)*, pages 167–176. IEEE Computer Society, 2009. `www.mrmc-tool.org`.

[8] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. In *Proc. Tools Session of Aachen 2001 International Multiconference on Measurement, Modelling and Evaluation of Computer-Communication Systems*, pages 7–12, September 2001.

[9] M. Neuhäußer and L. Zhang. Time-bounded reachability in continuous-time Markov decision processes. Technical Report AIB-2009-12, RWTH Aachen, May 2009.

[10] M. R. Neuhäußer, M. Stoelinga, and J.-P. Katoen. Delayed nondeterminism in continuous-time Markov decision processes. In *FOSSACS '09: Proceedings of the 12th International Conference on Foundations of Software Science and Computational Structures*, pages 364–379, Berlin, Heidelberg, 2009. Springer-Verlag.