Efficient Approximation of Optimal Control for Continuous-Time Markov Games*

John Fearnley[1], Markus Rabe[2], Sven Schewe[1], and Lijun Zhang[3]

1 Department of Computer Science, University of Liverpool, Liverpool, United Kingdom
2 Department of Computer Science, Universität des Saarlandes, Saarbrücken, Germany
3 DTU Informatics, Technical University of Denmark, Lyngby, Denmark

Abstract

We study the time-bounded reachability problem for continuous-time Markov decision processes (CTMDPs) and games (CTMGs). Existing techniques for this problem use discretisation techniques to break time into discrete intervals of size ε, and optimal control is approximated for each interval separately. Current techniques provide an accuracy of $O(\varepsilon^2)$ on each interval, which leads to an infeasibly large number of intervals. We propose a sequence of approximations that achieve accuracies of $O(\varepsilon^3)$, $O(\varepsilon^4)$, and $O(\varepsilon^5)$, that allow us to drastically reduce the number of intervals that are considered. For CTMDPs, the performance of the resulting algorithms is comparable to the heuristic approach given by Buckholz and Schulz [5], while also being theoretically justified. All of our results generalise to CTMGs, where our results yield the first practically implementable algorithms for this problem. We also provide memoryless strategies for both players that achieve similar error bounds.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, G.1.2 Approximation

Keywords and phrases Continuous time Markov decision processes and games, Discretisation

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2011.399

1 Introduction

Probabilistic models are being used extensively in the formal analysis of complex systems, including networked, distributed, and most recently, biological systems. Over the past 15 years, probabilistic model checking for discrete-time Markov decision processes (MDPs) and continuous-time Markov chains (CTMCs) has been successfully applied to these rich academic and industrial applications [8, 7, 9, 3]. However, the theory for continuous-time Markov decision processes (CTMDPs), which mix the non-determinism of MDPs with the continuous-time setting of CTMCs [2], is less well developed.

This paper studies the *time-bounded reachability* problem for CTMDPs and their extension to continuous-time Markov games, which is a model with both helpful and hostile non-determinism. This problem is of paramount importance for model checking applications [4]. The non-determinism in the system is resolved by providing a scheduler. The

* This work was partly supported by the Engineering and Physical Science Research Council (EPSRC) through the grant EP/H046623/1 ‘Synthesis and Verification in Markov Game Structures’, the Transregional Collaborative Research Center “Automatic Verification and Analysis of Complex Systems” (SFB/TR 14 AVACS) of the DFG, and the VKR Centre of Excellence MT-LAB.

31st Int’l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011).
Editors: Supratik Chakraborty, Amit Kumar; pp. 399–410
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

time-bounded reachability problem is to determine or to approximate, for a given set of goal locations G and time bound T, the maximal (or minimal) probability of reaching G before the deadline T that can be achieved by a scheduler.

For CTMCs, this problem can be solved efficiently by the Runge-Kutta method. However, this method requires that the target function can be continuously differentiated four times. Once we move to the CTMDP setting, our target function is not continuously differentiable at all. This is because changing the choice of action at a state introduces a discontinuity in the derivative of the time bounded-reachability probability.

Early work on this problem for CTMDPs focused on restricted classes of schedulers, such schedulers without any access to time in systems with uniform transition rates [1]. Recently however, results have been proved for the more general class of *late schedulers* [13], which will be studied in this paper. The different classes of schedulers are contrasted by Neuhäußer et. al. [12], and they show that late schedulers are the most powerful class. Several algorithms have been given to approximate the time-bounded reachability probabilities for CTMDPs using this scheduler class [4, 6, 13, 15].

The current state-of-the-art techniques for solving this problem are based on different forms of *discretisation*. This technique splits the time bound T into small intervals of length ε. Optimal control is approximated for each interval separately, and these approximations are combined to produce the final result. Current techniques can approximate optimal control on an interval of length ε with an accuracy of $O(\varepsilon^2)$. However, to achieve a precision of π with these techniques, one must choose $\varepsilon \approx \pi/T$, which leads to $O(T^2/\pi)$ many intervals. Since the desired precision is often high (it is common to require that $\pi \leq 10^{-6}$), this leads to an infeasibly large number of intervals that must be considered by the algorithms.

A recent paper of Buckholz and Schulz [5] has addressed this problem for practical applications, by allowing the interval sizes to vary. In addition to computing an approximation of the maximal time-bounded reachability probability, which provides a lower bound on the optimum, they also compute an upper bound. As long as the upper and lower bounds do not diverge too far, the interval can be extended indefinitely. In practical applications, where the optimal choice of action changes infrequently, this idea allows their algorithm to consider far fewer intervals while still maintaining high precision. However, from a theoretical perspective, their algorithm is not particularly satisfying. Their method for extending interval lengths depends on a heuristic, and in the worst case their algorithm may consider $O(T^2/\pi)$ intervals, which is not better than other discretisation based techniques.

Our contribution. In this paper we present a method of obtaining larger interval sizes that satisfies both theoretical and practical concerns. Our approach is to provide more precise approximations for each ε length interval. While current techniques provide an accuracy of $O(\varepsilon^2)$, we propose a sequence of approximations, called double ε-nets, triple ε-nets, and quadruple ε-nets, with accuracies $O(\varepsilon^3)$, $O(\varepsilon^4)$, and $O(\varepsilon^5)$, respectively. Since these approximations are much more precise on each interval, they allow us to consider far fewer intervals while still maintaining high precision. For example, Table 1 gives the number of intervals considered by our algorithms, in the worst case, for a normed CTMDP with time bound $T = 10$.

Of course, in order to become more precise, we must spend additional computational effort. However, the cost of using double ε-nets instead of using current techniques requires only an extra factor of $\log |\Sigma|$, where Σ is the set of actions. Thus, in almost all cases, the large reduction in the number of intervals far outweighs the extra cost of using double ε-nets. Our worst case running times for triple and quadruple ε-nets are not so attractive: triple ε-nets require an extra $|L| \cdot |\Sigma|^2$ factor over double ε-nets, where L is the set of locations,

Table 1 The number of intervals needed by our algorithms for precisions $10^{-7}, 10^{-9}$, and 10^{-11}.

Technique	Error	$\pi = 10^{-7}$	$\pi = 10^{-9}$	$\pi = 10^{-11}$
Current techniques	$O(\varepsilon^2)$	$1,000,000,000$	$100,000,000,000$	$10,000,000,000,000$
Double ε-nets	$O(\varepsilon^3)$	$81,650$	$816,497$	$8,164,966$
Triple ε-nets	$O(\varepsilon^4)$	$3,219$	$14,939$	$69,337$
Quadruple ε-nets	$O(\varepsilon^5)$	605	$1,911$	$6,043$

and quadruple ε-nets require yet another $|L| \cdot |\Sigma|^2$ factor over triple ε-nets. However, these worst case running times only occur when the choice of optimal action changes frequently, and we speculate that the cost of using these algorithms in practice is much lower than our theoretical worst case bounds. Our experimental results with triple ε-nets support this claim.

An added advantage of our techniques is that they can be applied to continuous-time Markov games as well as to CTMDPs. Buckholz and Schulz restrict their analysis to CTMDPs. Moreover, previous works on CTMGs have been restricted to simplified settings, such as the time-abstract setting [4]. Therefore, to the best of our knowledge, we present the first practically implementable approximation algorithms for the time-dependent time-bounded reachability problem in CTMGs. Each of our approximations also provide memoryless strategies for both players that achieve similar error bounds.

2 Preliminaries

▶ **Definition 1.** A continuous-time Markov game (or simply Markov game) is a tuple $(L, L_r, L_s, \Sigma, \mathbf{R}, \mathbf{P}, \nu)$, consisting of a finite set L of locations, which is partitioned into locations L_r (controlled by a *reachability* player) and L_s (controlled by a *safety* player), a finite set Σ of actions, a rate matrix $\mathbf{R} : (L \times \Sigma \times L) \to \mathbb{Q}_{\geqslant 0}$, a discrete transition matrix $\mathbf{P} : (L \times \Sigma \times L) \to \mathbb{Q} \cap [0, 1]$, and an initial distribution $\nu \in \mathit{Dist}(L)$.

We require that the following side-conditions hold: For all locations $l \in L$, there must be an action $a \in \Sigma$ such that $\mathbf{R}(l, a, L) := \sum_{l' \in L} \mathbf{R}(l, a, l') > 0$, which we call *enabled*. We denote the set of enabled actions in l by $\Sigma(l)$. For a location l and actions $a \in \Sigma(l)$, we require for all locations l' that $\mathbf{P}(l, a, l') = \frac{\mathbf{R}(l,a,l')}{\mathbf{R}(l,a,L)}$, and we require $\mathbf{P}(l, a, l') = 0$ for non-enabled actions. We define the *size* $|\mathcal{M}|$ of a Markov game as the number of non-zero rates in the rate matrix $\mathbf{R}$.

A Markov game is called *uniform* with uniformisation rate λ, if $\mathbf{R}(l, a, L) = \lambda$ holds for all locations l and enabled actions $a \in \Sigma(l)$. We further call a Markov game *normed*, if its uniformisation rate is 1. Note that for normed Markov games we have $\mathbf{R} = \mathbf{P}$. We will present our results for normed Markov games only. The following lemma states that our algorithms for normed Markov games can be applied to solve Markov games that are not normed.

▶ **Lemma 2.** *We can adapt an $O(f(\mathcal{M}))$ time algorithm for normed Markov games to solve an arbitrary Markov game in time $O(f(\mathcal{M}) + |L|)$.*

We are particularly interested in Markov games with a single player, which are continuous-time Markov decision processes (CTMDPs). In CTMDPs all positions belong to the reachability player ($L = L_r$), or to the safety player ($L = L_s$), depending on whether we analyse the *maximum* or *minimum* reachability probability problem.

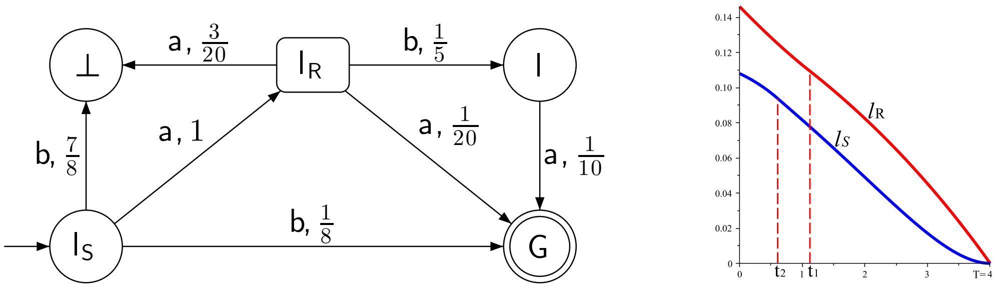

Figure 1 Left: a normed Markov game. Right: the function f within $[0, 4]$ for l_R and l_S.

As a running example, we will use the normed Markov game shown in the left half of Figure 1. Locations belonging to the safety player are drawn as circles, and locations belonging to the reachability player are drawn as rectangles. The self-loops of the normed Markov game are not drawn, but rates assigned to the self loops can be derived from the other rates: for example, we have $\mathbf{R}(l_R, a, l_R) = 0.8$. The locations G and $\perp$ are absorbing, and there is only a single enabled action for l. It therefore does not matter which player owns l, G, and $\perp$.

2.1 Schedulers and Strategies

We consider Markov games in a time interval $[0, T]$ with $T \in \mathbb{R}_{\geq 0}$. The non-determinism in the system needs to be resolved by a pair of strategies for the two players which together form a *scheduler* for the whole system. Formally, a strategy is a function in $\mathit{Paths}_{r/s} \times [0, T] \to \Sigma$, where Paths_r and Paths_s are the sets of finite paths $l_0 \xrightarrow{a_0, t_0} l_1 \ldots \xrightarrow{a_{n-1}, t_{n-1}} l_n$ with $l_n \in L_r$ and $l_n \in L_s$, respectively. We use $\mathcal{S}_r$ and $\mathcal{S}_s$ to denote the strategies of reachability player and the strategies of safety player, respectively, and we use Π_r and Π_s to denote the set of all strategies for the reachability and safety players, respectively. (For technical reasons one has to restrict the schedulers to those which are measurable. This restriction, however, is of no practical relevance. In particular, simple piecewise constant timed-positional strategies $L \times [0, T] \to \Sigma$ suffice for optimal scheduling [14, 13, 2], and all schedulers that occur in this paper are from the particularly tame class of cylindrical schedulers [14].)

If we fix a pair $(\mathcal{S}_r, \mathcal{S}_s)$ of strategies, we obtain a deterministic stochastic process, which is in fact a time inhomogeneous Markov chain, and we denote it by $\mathcal{M}_{\mathcal{S}_{r,s}}$. For $t \leq T$, we use $Pr_{\mathcal{S}_{r+s}}(t)$ to denote the transient distribution at time t over S under the scheduler $(\mathcal{S}_r, \mathcal{S}_s)$.

Given a Markov game $\mathcal{M}$, a goal region $G \subseteq L$, and a time bound $T \in \mathbb{R}_{\geq 0}$, we are interested in the *optimal* probability of being in a goal state at time T (and the corresponding pair of optimal strategies). This is given by:

$$\sup_{\mathcal{S}_r \in \Pi_r} \inf_{\mathcal{S}_s \in \Pi_s} \sum_{l \in G} Pr_{\mathcal{S}_{r+s}}(l, T),$$

where $Pr_{\mathcal{S}_{r+s}}(l, T) := Pr_{\mathcal{S}_{r+s}}(T)(l)$. It is commonly referred to as the *maximum* time-bounded reachability probability problem in the case of CTMDPs with a reachability player only. For $t \leq T$, we define $f : L \times \mathbb{R}_{\geq 0} \to [0, 1]$, to be the optimal probability to be in the goal region at the time bound T, assuming that we start in location l and that t time units have passed already. By definition, it holds then that $f(l, T) = 1$ if $l \in G$ and $f(l, T) = 0$ if $l \notin G$. Optimising the vector of values $f(\cdot, 0)$ then yields the optimal value and its optimal piecewise deterministic strategy.

Let us return to the example shown in Figure 1. The right half of the Figure shows the optimal reachability probabilities, as given by f, for the locations l_R and l_S when the time bound $T = 4$. The points $t_1 \approx 1.123$ and $t_2 \approx 0.609$ represent the times at which the optimal strategies change their decisions. Before t_1 it is optimal for the reachability player to use action b at l_R, but afterwards the optimal choice is action a. Similarly, the safety player uses action b before t_2, and switches to a afterwards.

2.2 Characterisation of f

We define a matrix $\mathbf{Q}$ such that $\mathbf{Q}(l,a,l') = \mathbf{R}(l,a,l')$ if $l' \neq l$ and $\mathbf{Q}(l,a,l) = -\sum_{l' \neq l} \mathbf{R}(l,a,l')$. The optimal function f can be characterised as a set of differential equations [2], see also [11, 10]. For each $l \in L$ we define $f(l,T) = 1$ if $l \in G$, and 0 if $l \notin G$. Otherwise, for $t < T$, we define:

$$-\dot{f}(l,t) = \underset{a \in \Sigma(l)}{\mathsf{opt}} \sum_{l' \in L} \mathbf{Q}(l,a,l') \cdot f(l',t), \tag{1}$$

where $\mathsf{opt} \in \{\max, \min\}$ is max for reachability player locations and min for safety player locations. We will use the opt-notation throughout this paper.

Using the matrix $\mathbf{R}$, Equation (1) can be rewritten to:

$$-\dot{f}(l,t) = \underset{a \in \Sigma(l)}{\mathsf{opt}} \sum_{l' \in L} \mathbf{R}(l,a,l') \cdot (f(l',t) - f(l,t)) \tag{2}$$

For uniform Markov games, we simply have $\mathbf{Q}(l,a,l) = \mathbf{R}(l,a,l) - \lambda$, with $\lambda = 1$ for normed Markov games. This also provides an intuition for the fact that uniformisation does not alter the reachability probability: the rate $\mathbf{R}(l,a,l)$ does not appear in (1).

3 Approximating Optimal Control for Normed Markov Games

In this section we describe ε-nets, which are a technique for approximating optimal values and strategies in a normed continuous-time Markov game. Thus, throughout the whole section, we fix a normed Markov game $\mathcal{M} = (L, L_r, L_s, \Sigma, \mathbf{R}, \mathbf{P}, \nu)$.

Our approach to approximating optimal control within the Markov game is to break time into intervals of length ε, and to approximate optimal control separately in each of the $\lceil \frac{T}{\varepsilon} \rceil$ distinct intervals. Optimal time-bounded reachability probabilities are then computed iteratively for each interval, starting with the final interval and working backwards in time. The error made by the approximation in each interval is called the *step error*. In Section 3.1 we show that if the step error in each interval is bounded, then the *global error* made by our approximations is also bounded.

Our results begin with a simple approximation that finds the optimal action at the start of each interval, and assumes that this action is optimal for the duration of the interval. We refer to this as the *single* ε-net technique, and we will discuss this approximation in Section 3.2. While it only gives a simple linear function as an approximation, this technique gives error bounds of $O(\varepsilon^2)$, which is comparable to existing techniques.

However, single ε-nets are only a starting point for our results. Our main observation is that, if we have a piecewise polynomial approximation of degree c that achieves an error bound of $O(\varepsilon^k)$, then we can compute a piecewise polynomial approximation of degree $c+1$ that achieves an error bound of $O(\varepsilon^{k+1})$. Thus, starting with single ε-nets, we can construct double ε-nets, triple ε-nets, and quadruple ε-nets, with each of these approximations becoming increasingly more precise. The construction of these approximations will be discussed in Sections 3.3 and 3.4.

In addition to providing an approximation of the time-bounded reachability probabilities, our techniques also provide memoryless strategies for both players. For each level of ε-net, we will define two approximations: the function p_1 is the approximation for the time-bounded reachability probability given by single ε-nets, and the function g_1 gives the reachability probability obtained by following the memoryless strategy that is derived from p_1. This notation generalises to deeper levels of ε-nets: the functions p_2 and g_2 are produced by double ε-nets, and so on.

We will use $\mathcal{E}(k,\varepsilon)$ to denote the difference between p_k and f. In other words, $\mathcal{E}(k,\varepsilon)$ gives the difference between the approximation p_k and the true optimal reachability probabilities. We will use $\mathcal{E}_s(k,\varepsilon)$ to denote the difference between g_k and f. We defer formal definition of these measures to subsequent sections. Our objective in the following subsections is to show that the step errors $\mathcal{E}(k,\varepsilon)$ and $\mathcal{E}_s(k,\varepsilon)$ are in $O(\varepsilon^{k+1})$, with small constants.

3.1 Step Error and Global Error

In subsequent sections we will prove bounds on the *ε-step* error made by our approximations. This is the error that is made in a single interval of length ε. However, in order for our approximations to be valid, they must provide a bound on the *global* error, which is the error made by our approximations over every ε interval. In this section, we prove that, if the ε-step error of an approximation is bounded, then the global error of the approximation is bounded by the sum of these errors.

We define $f : [0,T] \to [0,1]^{|L|}$ as the vector valued function $f(t) \mapsto \bigotimes_{l\in L} f(l,t)$ that maps each point of time to a vector of reachability probabilities, with one entry for each location. Given two such vectors $f(t)$ and $p(t)$, we define the maximum norm $\|f(t)-p(t)\| = \max\{|f(l,t)-p(l,t)| \mid l \in L\}$, which gives the largest difference between $f(l,t)$ and $p(l,t)$.

We also introduce notation that will allow us to define the values at the start of an ε interval. For each interval $[t-\varepsilon, t]$, we define $f_x^t : [t-\varepsilon, t] \to [0,1]^{|L|}$ to be the function obtained from the differential equations (1) when the values at the time t are given by the vector $x \in [0,1]^{|L|}$. More formally, if $\tau = t$ then we define $f_x^t(\tau) = x$, and if $t-\varepsilon \le \tau < t$ and $l \in L$ then we define:

$$-\dot{f}_x^t(l,\tau) = \underset{a\in\Sigma(l)}{\mathsf{opt}} \sum_{l'\in L} \mathbf{Q}(l,a,l') f_x^t(l',\tau). \tag{3}$$

The following lemma states that if the ε-step error is bounded for every interval, then the global error is simply the sum of these errors.

▶ **Lemma 3.** *Let p be an approximation of f that satisfies $\|f(t)-p(t)\| \le \mu$ for some time point $t \in [0,T]$. If $\|f_{p(t)}^t(t-\varepsilon) - p(t-\varepsilon)\| \le \nu$ then we have $\|f(t-\varepsilon) - p(t-\varepsilon)\| \le \mu + \nu$.*

3.2 Single ε-Nets

In single ε-nets, we compute the gradient of the function f at the end of each interval, and we assume that this gradient remains constant throughout the interval. This yields a *linear* approximation function p_1, which achieves a local error of ε^2.

We now define the function p_1. For initialisation, we define $p_1(l,T) = 1$ if $l \in G$ and $p_1(l,T) = 0$ otherwise. Then, if p_1 is defined for the interval $[t,T]$, we will use the following procedure to extend it to the interval $[t-\varepsilon, T]$. We first determine the optimising enabled actions for each location for $f^t_{p_1(t)}$ at time t. That is, we choose, for all $l \in L$, an action:

$$a^t_l \in \underset{a \in \Sigma(l)}{\arg\mathsf{opt}} \sum_{l' \in L} \mathbf{Q}(l, a, l') \cdot p_1(l', t). \tag{4}$$

We then fix $c^t_l = \sum_{l' \in L} \mathbf{Q}(l, a^t_l, l') \cdot p_1(l', t)$ as the descent of $p_1(l, \cdot)$ in the interval $[t - \varepsilon, t]$. Therefore, for every $\tau \in [0, \varepsilon]$ and every $l \in L$ we have:

$$-\dot{p}_1(l, t - \tau) = c^t_l \quad \text{and} \quad p_1(l, t - \tau) = p_1(l, t) + \tau \cdot c^t_l.$$

Let us return to our running example. We will apply the approximation p_1 to the example shown in Figure 1. We will set $\varepsilon = 0.1$, and focus on the interval $[1.1, 1.2]$ with initial values $p_1(G, 1.2) = 1$, $p_1(l, 1.2) = 0.244$, $p_1(l_R, 1.2) = 0.107$, $p_1(l_S, 1.2) = 0.075$, $p_1(\bot, 1.2) = 0$. These are close to the true values at time 1.2. Note that the point t_1, which is the time at which the reachability player switches the action played at l_R, is contained in the interval $[1.1, 1.2]$. Applying Equation (4) with these values allows us to show that the maximising action at l_R is a, and the minimising action at l_S is also a. As a result, we obtain the approximation $p_1(l_R, t - \tau) = 0.0286\tau + 0.107$ and $p_1(l_S, t - \tau) = 0.032\tau + 0.075$.

We now prove error bounds for p_1. Recall that $\mathcal{E}(1, \tau)$ denotes the difference between f and p_1 after τ time units. We can now formally define this error, and prove the following bounds.

▶ **Lemma 4.** *If $\varepsilon \le 1$, then $\mathcal{E}(1, \varepsilon) := \|f^t_{p_1(t)}(t - \varepsilon) - p_1(t - \varepsilon)\| \le \varepsilon^2$.*

The approximation p_1 can also be used to construct strategies for the two players with similar error bounds. We will describe the construction for the reachability player. The construction for the safety player can be derived analogously.

The strategy for the reachability player is to play the action chosen by p_1 during the entire interval $[t - \varepsilon, t]$. We will define a system of differential equations $g_1(l, \tau)$ that describe the outcome when the reachability fixes this strategy, and when the safety player plays an optimal counter strategy. For each location l, we define $g_1(l, t) = f^t_{p_1(t)}(l, t)$, and we define $g_1(l, \tau)$, for each $\tau \in [t - \varepsilon, t]$, as:

$$-\dot{g}_1(l, \tau) = \sum_{l' \in L} \mathbf{Q}(l, a^t_l, l') \cdot g_1(l', \tau) \qquad \text{if } l \in L_r, \tag{5}$$

$$-\dot{g}_1(l, \tau) = \min_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{Q}(l, a, l') \cdot g_1(l', \tau) \qquad \text{if } l \in L_s. \tag{6}$$

We can prove the following bounds for $\mathcal{E}_s(1, \varepsilon)$, which is the difference between g_1 and $f^t_{p_1(t)}$ on an interval of length ε.

▶ **Lemma 5.** *We have $\mathcal{E}_s(1, \varepsilon) := \|g_1(t - \varepsilon) - f^t_{p_1(t)}(t - \varepsilon)\| \le 2 \cdot \varepsilon^2$.*

Lemma 4 gives the ε-step error for p_1, and we can apply Lemma 3 to show that the global error is bounded by $\varepsilon^2 \cdot \frac{T}{\varepsilon} = \varepsilon T$. If π is the required precision, then we can choose $\varepsilon = \frac{\pi}{T}$ to produce an algorithm that terminates after $\frac{T}{\varepsilon} \approx \frac{T^2}{\pi}$ many steps. Hence, we obtain the following known result.

▶ **Theorem 6.** *For a normed Markov game $\mathcal{M}$ of size $|\mathcal{M}|$, we can compute a π-optimal strategy and determine the quality of $\mathcal{M}$ up to precision π in time $O(|\mathcal{M}| \cdot T \cdot \frac{T}{\pi})$.*

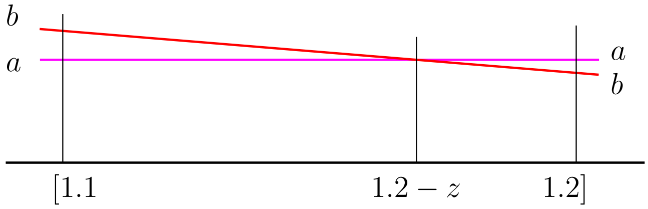

Figure 2 This figure shows how $-\dot{p}_2$ is computed on the interval $[1.1, 1.2]$ for the location l_R. The function is given by the upper envelope of the two functions: it agrees with the quality of a on the interval $[1.2 - z, 1.2]$ and with the quality of b on the interval $[1.1, 1.2 - z]$.

3.3 Double ε-Nets

In this section we show that only a small amount of additional computation effort needs to be expended in order to dramatically improve over the precision obtained by single ε-nets. This will allow us to use much larger values of ε while still retaining our desired precision.

In single ε-nets, we computed the gradient of f at the start of each interval and assumed that the gradient remained constant for the duration of that interval. This gave us the approximation p_1. The key idea behind double ε-nets is that we can use the approximation p_1 to approximate the gradient of f throughout the interval.

We define the approximation p_2 as follows: we have $p_2(l, T) = 1$ if $l \in G$ and 0 otherwise, and if $p_2(l, \tau)$ is defined for every $l \in L$ and every $\tau \in [t, T]$, then we define $p_2(l, \tau)$ for every $\tau \in [t - \varepsilon, t]$ as:

$$-\dot{p}_2(l, \tau) = \operatorname*{opt}_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{R}(l, a, l') \cdot (p_1(l', \tau) - p_1(l, \tau)) \quad \forall l \in L. \tag{7}$$

By comparing Equations (7) and (2), we can see that double ε-nets uses p_1 as an approximation for f during the interval $[t - \varepsilon, t]$. Furthermore, in contrast to p_1, note that the approximation p_2 can change it's choice of optimal action during the interval. The ability to change the choice of action during an interval is the key property that allows us to prove stronger error bounds than previous work.

▶ **Lemma 7.** If $\varepsilon \le 1$ then $\mathcal{E}(2, \varepsilon) := \|p_2(\tau) - f^t_{p_2(t)}(\tau)\| \le \frac{2}{3}\varepsilon^3$.

Let us apply the approximation p_2 to the example shown in Figure 1. We will again use the interval $[1.1, 1.2]$, and we will use initial values that were used when we applied single ε-nets to the example in Section 3.2. We will focus on the location l_R. From the previous section, we know that $p_1(l_R, t - \tau) = 0.0286\tau + 0.107$, and for the actions a and b we have:

- $\sum_{l' \in L} \mathbf{R}(l_R, a, l') p_1(l', t - \tau) = \frac{1}{20} + \frac{4}{5} p_1(l_R, t - \tau)$,
- $\sum_{l' \in L} \mathbf{R}(l_R, b, l') p_1(l', t - \tau) = \frac{1}{5} p_1(l, t - \tau) + \frac{4}{5} p_1(l_R, t - \tau)$.

These functions are shown in Figure 2. To obtain the approximation p_2, we must take the maximum of these two functions. Since p_1 is a linear function, we know that these two functions have exactly one crossing point, and it can be determined that this point occurs when $p_1(l, t - \tau) = 0.25$, which happens at $\tau = z := \frac{5}{63}$. Since $z \le 0.1 = \varepsilon$, we know that the lines intersect within the interval $[1.1, 1.2]$. Consequently, we get the following piecewise quadratic function for p_2:

- When $0 \le \tau \le z$, we use the action a and obtain $-\dot{p}_2(l_R, t - \tau) = -0.00572\tau + 0.0286$, which implies that $p_2(l_R, t - \tau) = -0.00286\tau^2 + 0.0286\tau + 0.107$.
- When $z < \tau \le 0.1$ we use action b and obtain $-\dot{p}_2(l_R, t - \tau) = 0.0094\tau + 0.0274$, which implies that $p_2(l_R, t - \tau) = 0.0047\tau^2 + 0.0274\tau + 0.107047619$.

As with single ε-nets, we can provide a strategy that obtains similar error bounds. Once again, we will consider only the reachability player, because the proof can easily be generalised for the safety player. In much the same way as we did for g_1, we will define a system of differential equations $g_2(l,\tau)$ that describe the outcome when the reachability player plays according to p_2, and the safety player plays an optimal counter strategy. For each location l, we define $g_2(l,t) = f^t_{p_2(t)}(l,t)$. If a^τ_l denotes the action that maximises Equation (7) at the time point $\tau \in [t-\varepsilon, t]$, then we define $g_2(l,\tau)$, as:

$$-\dot{g_2}(l,\tau) = \sum_{l' \in L} \mathbf{Q}(l, a^\tau_l, l') \cdot g_2(l',\tau) \qquad \text{if } l \in L_r, \tag{8}$$

$$-\dot{g_2}(l,\tau) = \min_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{Q}(l, a, l') \cdot g_2(l',\tau) \qquad \text{if } l \in L_s. \tag{9}$$

The following lemma proves that difference between g_2 and $f^t_{p_2(t)}$ has similar bounds to those shown in Lemma 7

▶ **Lemma 8.** *If $\varepsilon \le 1$ then we have $\mathcal{E}_s(2,\varepsilon) := \|g_2(t-\varepsilon) - f^t_{p_2(t)}(t-\varepsilon)\| \le 2 \cdot \varepsilon^3$.*

Computing the approximation p_2 for an interval $[t-\varepsilon, t]$ is not expensive. The fact that p_1 is linear implies that each action can be used for at most one subinterval of $[t-\varepsilon, t]$. Therefore, there are less than $|\Sigma|$ points at which the strategy changes, which implies that p_2 is a piecewise quadratic function with at most $|\Sigma|$ pieces. It is possible to design an algorithm that uses sorting to compute these switching points, achieving the following complexity.

▶ **Lemma 9.** Computing p_2 for an interval $[t-\varepsilon, t]$ takes $O(|\mathcal{M}| + |L| \cdot |\Sigma| \cdot \log|\Sigma|)$ time.

Since the ε-step error for double ε-nets is bounded by ε^3, we can apply Lemma 3 to conclude that the global error is bounded by $\varepsilon^3 \cdot \frac{T}{\varepsilon} = \varepsilon^2 T$. Therefore, if we want to compute f with a precision of π, we should choose $\varepsilon \approx \sqrt{\frac{\pi}{T}}$, which gives $\frac{T}{\varepsilon} \approx \frac{T^{1.5}}{\sqrt{\pi}}$ distinct intervals.

▶ **Theorem 10.** For a normed Markov game $\mathcal{M}$ we can approximate the time-bounded reachability, construct π optimal memoryless strategies for both players, and determine the quality of these strategies with precision π in time $O(|\mathcal{M}| \cdot T \cdot \sqrt{\frac{T}{\pi}} + |L| \cdot T \cdot \sqrt{\frac{T}{\pi}} \cdot |\Sigma| \log|\Sigma|)$.

3.4 Triple ε-Nets and Beyond

The techniques used to construct the approximation p_2 from the approximation p_1 can be generalised. This is because the only property of p_1 that is used in the proof of Lemma 7 is the fact that it is a piecewise polynomial function that approximates f. Therefore, we can inductively define a sequence of approximations p_k as follows:

$$-\dot{p}_k(l,\tau) = \mathop{\mathsf{opt}}_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{R}(l,a,l') \cdot (p_{k-1}(l',\tau) - p_{k-1}(l,\tau)) \tag{10}$$

We can repeat the arguments from the previous sections to obtain the following error bounds:

▶ **Lemma 11.** For every $k > 2$, if we have $\mathcal{E}(k,\varepsilon) \le c \cdot \varepsilon^{k+1}$, then we have $\mathcal{E}(k+1,\varepsilon) \le \frac{2}{k+2} \cdot c \cdot \varepsilon^{k+2}$. Moreover, if we additionally have that $\mathcal{E}_s(k,\varepsilon) \le d \cdot \varepsilon^{k+1}$, then we also have that $\mathcal{E}_s(k+1,\varepsilon) \le \frac{8c+3d}{k+2} \cdot \varepsilon^{k+2}$.

Computing the accuracies explicitly for the first four levels of ε-nets gives:

k	1	2	3	4	…
$\mathcal{E}(k,\varepsilon)$	ε^2	$\frac{2}{3}\varepsilon^3$	$\frac{1}{3}\varepsilon^4$	$\frac{2}{15}\varepsilon^5$	…
$\mathcal{E}_s(k,\varepsilon)$	$2\varepsilon^2$	$2\varepsilon^3$	$\frac{17}{6}\varepsilon^4$	$\frac{67}{30}\varepsilon^5$	…

We can also compute, for a given precision π, the value of ε that should be used in order to achieve an accuracy of π with ε-nets of level k.

▶ **Lemma 12.** To obtain a precision π with an ε-net of level k, we choose $\varepsilon \approx \sqrt[k]{\frac{\pi}{T}}$, resulting in $\frac{T}{\varepsilon} \approx T\sqrt[k]{\frac{T}{\pi}}$ steps.

Unfortunately, the cost of computing ε-nets of level k becomes increasingly prohibitive as k increases. To see why, we first give a property of the functions p_k. Recall that p_2 is a piecewise quadratic function. It is not too difficult to see how this generalises to the approximations p_k.

▶ **Lemma 13.** *The approximation p_k is piecewise polynomial with degree less than or equal to k.*

Although these functions are well-behaved in the sense that they are always piecewise polynomial, the number of pieces can grow exponentially in the worst case. The following lemma describes this bound.

▶ **Lemma 14.** If p_{k-1} has c pieces in the interval $[t-\varepsilon, t]$, then p_k has at most $\frac{1}{2}\cdot c\cdot k\cdot |L|\cdot|\Sigma|^2$ pieces in the interval $[t-\varepsilon, t]$.

The upper bound given above is quite coarse, and we would be surprised if it were found to be tight. Moreover, we do not believe that the number of pieces will grow anywhere close to this bound in practice. This is because it is rare, in our experience, for optimal strategies to change their decision many times within a small time interval.

However, there is a more significant issue that makes ε-nets become impractical as k increases. In order to compute the approximation p_k, we must be able to compute the roots of polynomials with degree $k-1$. Since we can only efficiently compute the roots of quadratic functions, and efficiently approximate the roots of cubic functions, only the approximations p_3 and p_4 are realistically useful.

Once again it is possible to provide a smart algorithm that uses sorting in order to find the switching points in the functions p_3 and p_4, which gives the following bounds on the cost of computing them.

▶ **Theorem 15.** For a normed Markov $\mathcal{M}$ we can construct π optimal memoryless strategies for both players and determine the quality of these strategies with precision π in time $O(|L|^2\cdot\sqrt[3]{\frac{T}{\pi}}\cdot T\cdot|\Sigma|^4\log|\Sigma|)$ when using triple ε-nets, and in time $O(|L|^3\cdot\sqrt[4]{\frac{T}{\pi}}\cdot T\cdot|\Sigma|^6\log|\Sigma|)$ when using quadruple ε-nets.

It is not clear if triple and quadruple ε-nets will only be of theoretical interest, or if they will be useful in practice. It should be noted that the worst case complexity bounds given by Theorem 15 arise from the upper bound on the number of switching points given in Lemma 14. Thus, if the number of switching points that occur in practical examples is small, these techniques may become more attractive. Our experiments in the following section give some evidence that this may be true.

4 Experimental Results and Conclusion

In order to test the practicability of our algorithms, we have implemented both double and triple-ε nets. We evaluated these algorithms on two sets of examples. Firstly, we tested our algorithms on the Erlang-example (see Figure 3) presented in [4] and [15]. We chose

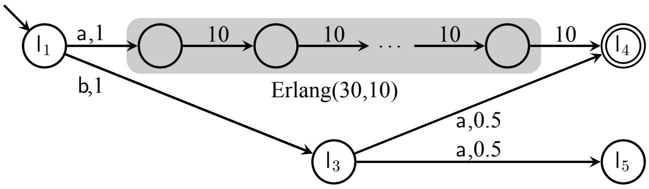

Figure 3 A CTMDP offering the choice between a long chain of fast transition and a slower path that looses some probability mass in l_5.

Table 2 Experimental evalutation of our algorithms.

	Erlang model			Game model	
precision \ method	MRMC [4]	Double-nets	Triple-nets	Double-nets	Triple-nets
10^{-4}	0.05 s	0.04 s	0.01 s	0.29 s	0.06 s
10^{-5}	0.20 s	0.10 s	0.02 s	0.93 s	0.13 s
10^{-6}	1.32 s	0.32 s	0.03 s	2.94 s	0.28 s
10^{-7}	8 s	0.98 s	0.06 s	9.35 s	0.60 s
10^{-8}	475 s	3.11 s	0.12 s	29.21 s	1.29 s
10^{-9}	—	9.91 s	0.27 s	94 s	2.78 s
10^{-10}	—	31.24 s	0.58 s	299 s	6.05 s

to consider the same parameters used by those papers: we consider maximal probability to reach location l_4 from l_1 within 7 time units. Since this example is a CTMDP, we were able to compare our results with the Markov Reward Model Checker (MRMC) [4] implementation, which includes an implementation of the techniques proposed by Buckholz and Schulz.

We also tested our algorithms on continuous-time Markov games, where we used the model depicted in Figure 4, consisting of two chains of locations $l_1, l_2, \ldots, l_{100}$ and $l'_1, l'_2, \ldots, l'_{100}$ that are controlled by the maximising player and the minimising player, respectively. This example is designed to produce a large number of switching points. In every location l_i of the maximising player, there is the choice between the short but slow route along the chain of maximising locations, and the slightly longer route which uses the minimising player's locations. If very little time remains, the maximising player prefers to take the slower actions, as fewer transitions are required to reach the goal using these actions. The maximiser also prefers these actions when a large amount of time remains. However, between these two extremes, there is a time interval in which it is advantageous for the maximising player to take the action with rate 3. A similar situation occurs for the minimising player, and this leads to a large number of points where the players change their strategy.

The results of our experiments are shown in Table 2. The MRMC implementation was unable to provide results for precisions beyond $1.86 \cdot 10^{-9}$. For the Erlang examples we found that, as the desired precision increases, our algorithms draw further ahead of the current techniques. The most interesting outcome of these experiments is the validation of triple ε-nets for practical use. While the worst case theoretical bounds arising from Lemma 14 indicated that the cost of computing the approximation for each interval may become prohibitive, these results show that the worst case does not seem to play a role in practice. In fact, we found that the number of switching points summed over all intervals and locations never exceeded 2 in this example.

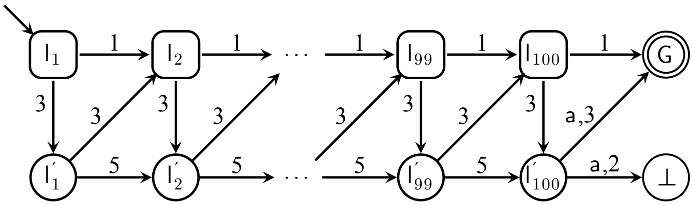

Figure 4 A CTMG with many switching points.

Our results on Markov games demonstrate that our algorithms are capable of solving non-trivially sized games in practice. Once again we find that triple ε-nets provide a substantial performance increase over double ε-nets, and that the worst case bounds given by Lemma 14 do not seem occur. Double ε-nets found 297 points where the strategy changed during an interval, and triple ε-nets found 684 such points. Hence, the $|L||\Sigma|^2$ factor given in Lemma 14 does not seem to arise here.

References

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

2 R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

3 M. Bozzano, A. Cimatti, M. Roveri, J.-P. Katoen, V. Y. Nguyen, and T. Noll. Verification and performance evaluation of AADL models. In *ESEC/SIGSOFT FSE*, pages 285–286, 2009.

4 P. Buchholz, E. M. Hahn, H. Hermanns, and L. Zhang. Model checking algorithms for CTMDPs. In *Proc. of CAV*, pages 225–242, 2011.

5 P. Buchholz and I. Schulz. Numerical analysis of continuous time Markov decision processes over finite horizons. *Computers and Operations Research*, 38(3):651–659, 2011.

6 T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Computing maximum reachability probabilities in Markovian timed automata. Technical report, RWTH Aachen, 2010.

7 N. Coste, H. Hermanns, E. Lantreibecq, and W. Serwe. Towards performance prediction of compositional models in industrial gals designs. In *Proc. of CAV*, pages 204–218, 2009.

8 H. Garavel, R. Mateescu, F. Lang, and W. Serwe. CADP 2006: A toolbox for the construction and analysis of distributed processes. In *Proc. of CAV*, pages 158–163, 2007.

9 T. A. Henzinger, M. Mateescu, and V. Wolf. Sliding window abstraction for infinite Markov chains. In *Proc. of CAV*, pages 337–352, 2009.

10 A. Martin-Löfs. Optimal control of a continuous-time Markov chain with periodic transition probabilities. *Operations Research*, 15(5):872–881, 1967.

11 B. L. Miller. Finite state continuous time Markov decision processes with a finite planning horizon. *SIAM Journal on Control*, 6(2):266–280, 1968.

12 M. R. Neuhäußer, M. Stoelinga, and J.-P. Katoen. Delayed nondeterminism in continuous-time Markov decision processes. In *Proc. of FOSSACS*, pages 364–379, 2009.

13 M. R. Neuhäußer and L. Zhang. Time-bounded reachability probabilities in continuous-time Markov decision processes. In *Proc. of QEST*, pages 209–218, 2010.

14 M. Rabe and S. Schewe. Finite optimal control for time-bounded reachability in continuous-time Markov games and CTMDPs. *Acta Informatica*, pages 291–315, 2011.

15 L. Zhang and M. R. Neuhäußer. Model checking interactive Markov chains. In *Proc. of TACAS*, pages 53–68, 2010.