Cost vs. Time in Stochastic Games and Markov Automata*

Hassan Hatefi[1], Bettina Braitling[2], Ralf Wimmer[2],
Luis María Ferrer Fioriti[1], Holger Hermanns[1], and Bernd Becker[2]

[1] Saarland University, Saarbrücken, Germany
{hhatefi | ferrer | hermanns}@cs.uni-saarland.de
[2] Albert-Ludwigs-Universität Freiburg, Freiburg im Breisgau, Germany
{braitlin | wimmer | becker}@informatik.uni-freiburg.de

Abstract. Costs and rewards are important tools for analysing quantitative aspects of models like energy consumption and costs of maintenance and repair. Under the assumption of transient costs, this paper considers the computation of expected cost-bounded rewards and cost-bounded reachability for Markov automata and stochastic games. We give a transformation of this class of properties to expected time-bounded rewards and time-bounded reachability, which can be computed by available algorithms. We prove the correctness of the transformation and show its effectiveness on a number of case studies.

1 Introduction

Markov automata (MA) [13] constitute a compositional modelling formalism for concurrent stochastic systems. They generalise discrete-time Markov chains (DTMCs), Markov decision processes (MDPs), probabilistic automata (PA [28]), continuous-time Markov chains (CTMCs), and interactive Markov chains (IMCs [22]). Markov automata form the semantic foundation of, among others, dynamic fault trees [6], stochastic activity networks, and generalised stochastic Petri nets (GSPNs) [12]. Compositional modelling for MA [31] is supported by the MAMA tool set [17, 18], also providing access to effective model analysis via the IMCA tool [16]. That analysis follows the principles of model checking [5]. Concretely speaking, algorithms for model checking time-bounded reachability and continuous stochastic logic (CSL) [21], as well as long-run average and expected reachability times [17, 18] are supported.

Apart from timing-related properties, there is an immensely large spectrum of potential applications that ask for integration of cost-related modelling and analysis. Costs, or dually rewards, are especially convenient to reflect economical

* This work was partly supported by the German Research Council (DFG) as part of the Transregional Collaborative Research Center AVACS (SFB/TR 14), by the EU 7th Framework Programme under grant agreement no. 295261 (MEALS) and 318490 (SENSATION), by the CDZ project CAP (GZ 1023), and by the CAS/SAFEA International Partnership Program for Creative Research Teams.

implications, power consumption, wear and abrasion, or other quantitative information. Therefore MA have lately been extended to MRA, Markov reward automata. In MRA, states and transitions can be equipped with rewards or costs, accumulated as time advances and as transitions are taken. Algorithms for computing the long-run average reward, for the expected cumulative reward until reaching a goal, and for the expected cumulative reward until a certain time bound are known and implemented [19]. Effective abstraction and refinement strategies for MRA have also been introduced [7], working on stochastic reward game abstractions of MRA.

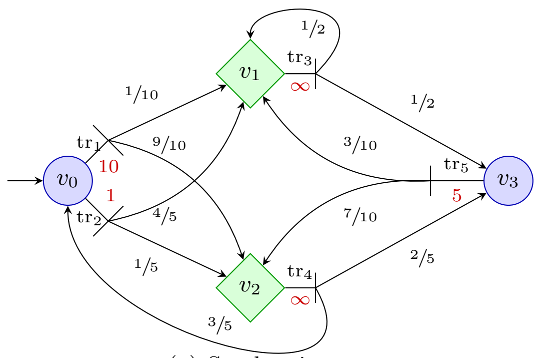
In this paper, we turn our attention to properties that relate multiple dimensions of cost or rewards. In particular, we enable the computation of expected cumulative rewards until exceeding a cost bound, both for Markov reward automata and stochastic reward games. This can, for instance, answer questions of central importance for energy-harvesting battery-powered missions: *Under a given initial budget, what is the maximum probability of the battery running dry, or how many tasks can maximally be expected to be carried out by the battery?*

To answer such questions we give a fixed point characterisation of expected cost-bounded rewards and a transformation for stochastic games from cost- to time-bounded rewards. This transformation supports arbitrary non-negative transient costs. Markov automata are closed under this transformation. After the transformation, arbitrary algorithms for expected time-bounded rewards like [19, 7] can be applied to compute expected cost-bounded rewards.

In order to develop our contribution, we take inspiration from various sources, especially from the domain of continuous-time Markov decision processes (CTMDPs). This encompasses works on necessary and sufficient criteria for optimality with respect to time-bounded rewards [24], and algorithms to compute optimal time-bounded rewards using uniformisation [10]. Instantaneous transition rewards have been added to the CTMC setting as well [11].

Our work is strongly influenced by the study of the duality between time and costs in CTMDPs under time-abstract strategies [4], built up on the earlier work in the setting of CTMCs [3]. We extend it in various dimensions: Our technique supports zero-cost states, where previously only strictly positive costs were allowed. We optimise over time-dependent strategies, which are a superclass of time-abstract ones. We extend the setting to expected reward analysis on two-player games with discrete and continuous locations, which is also an improvement over [14, 15]. And finally our analysis technique works for any kind of models, not only uniform ones.

Structure of the paper. In the following section, we introduce the necessary foundations. Sec. 3 describes the fixed point characterisation of optimal expected cost-bounded rewards and the transformation from cost to time bounds. We report on experimental results in Sec. 4 and conclude the paper in Sec. 5. An extended version of this paper with proofs of the main propositions is available at [20].



(a) Stochastic game

	tr_1	tr_2	tr_3	tr_4	tr_5	v_0	v_1	v_2	v_3
c	5	0	0	0	3				
ρ_t	1	3	0	0	1				
ρ_i	4	1	1	5	2				
ρ_f						1	0	2	3

(b) Costs and rewards

Fig. 1. An example of a stochastic game with costs and rewards

2 Foundations

Let V be a finite (or countably infinite) set. A probability distribution over V is a function $\mu : V \to [0,1]$ such that $\sum_{v \in V} \mu(v) = 1$. We denote the set of probability distributions over V by $\mathrm{Distr}(V)$. The real numbers are denoted by $\mathbb{R}$, $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers, and $\mathbb{R}^\infty_{\geq 0} := \mathbb{R}_{\geq 0} \cup \{\infty\}$. Accordingly $\mathbb{R}_{>0}$, $\mathbb{R}^\infty_{>0}$ etc. are used.

Definition 1 (Stochastic game). *A* stochastic (continuous-time two-player) game *(SG) is a tuple $\mathcal{G} = \big(V, (V_1, V_2), v_\mathrm{init}, T\big)$ such that $V = V_1 \uplus V_2$ is the finite set of states, $v_\mathrm{init} \in V$ is the initial state, and $T \subseteq V \times \mathbb{R}^\infty_{>0} \times \mathrm{Distr}(V)$ is the transition relation.*

V_1 and V_2 are the states of player 1 and player 2, respectively; we also denote them as V_1- and V_2-states. Transitions $(v, \lambda, \mu) \in T$ with rate $\lambda < \infty$ are called *Markovian*, transitions with infinite rate *probabilistic*. We denote the set of Markovian and probabilistic transitions by T_M and T_P, respectively. We use $T_\mathrm{M}(v)$ and $T_\mathrm{P}(v)$ to refer to the set of Markovian and probabilistic transitions available at state v. Then, $T(v) = T_\mathrm{M}(v) \uplus T_\mathrm{P}(v)$ is the set of all available transitions of v. We assume that $T(v) \neq \emptyset$ for all $v \in V$.

The game starts in state v_init. If the current state is $v \in V_1$, then it is player 1's turn, otherwise player 2's. The current player chooses a transition $(v, \lambda, \mu) \in T(v)$ for leaving state v. The rate $\theta_\mathrm{rate}\big((v, \lambda, \mu)\big) = \lambda \in \mathbb{R}^\infty_{\geq 0}$ determines how long we stay at v, whereas $\theta_\mathrm{distr}\big((v, \lambda, \mu)\big) = \mu \in \mathrm{Distr}(V)$ gives us the distribution which leads to the successor states. If $\lambda = \infty$, the transition is taken instantaneously. Otherwise, λ is taken as the parameter of an exponential distribution. In this case, the probability that a transition to state $v' \in V$ happens within $t \geq 0$ time units, is given by $\mu(v') \cdot (1 - \mathrm{e}^{-\lambda \cdot t})$. For conciseness, we write λ_tr instead of $\theta_\mathrm{rate}(\mathrm{tr})$ and μ_tr instead of $\theta_\mathrm{distr}(\mathrm{tr})$ for $\mathrm{tr} \in T$.

Example 1. Fig. 1(a) shows an example of a stochastic game. It consists of two player 1 states (drawn as circles) and two player 2 states (drawn as diamonds).

The exit rates of the transitions $\mathrm{tr}_1, \ldots, \mathrm{tr}_5$ are written in red. The game starts in v_0. Player 1 chooses one of the outgoing transitions $\{\mathrm{tr}_1, \mathrm{tr}_2\}$, say tr_1. The probability to stay in v_0 for at most t time units is then given by $1 - \mathrm{e}^{-10 \cdot t}$. When the transition fires, we move to v_1 with probability 0.1 and to v_2 with probability 0.9; say v_1 is the successor state. There it is player 2's turn. As only one outgoing transition is available, namely tr_3, and its exit rate is ∞, it is left immediately, either to v_1, again, or to v_3, both with probability 0.5. □

Markov automata (MA) [13] are a special type of stochastic games with a single player and without a nondeterministic choice between different Markovian transitions at one state. The reason for this restriction is that Markov automata are designed to be a compositional formalism, i. e. the MA for a system consisting of several components can be constructed from the MA of the individual components.

Definition 2 (Markov automaton). *A* Markov automaton *(MA) is a stochastic game* $\mathcal{M} = \big(V, (V, \emptyset), v_{\mathrm{init}}, T\big)$ *such that* $|T_{\mathrm{M}}(v)| \leq 1$ *holds for all* $v \in V$*. We simply write* $\mathcal{M} = (V, v_{\mathrm{init}}, T)$ *for a Markov automaton* $\mathcal{M}$*.*

In this paper we only consider *closed* Markov automata which are not subject to further composition operations. In this case, it is standard for Markov automata to make an *urgency assumption*: Since nothing prevents probabilistic transitions from happening instantaneously and the probability that a Markovian transition is taken without delay is zero, probabilistic transitions take precedence over Markovian transitions. Therefore we assume for MA that Markovian transitions have been removed from all states which also exhibit an outgoing probabilistic transition.

Paths through stochastic games. The dynamics of an SG is specified by paths. An infinite path $\pi \in (V \times \mathbb{R}_{\geq 0} \times T)^\omega$ is an infinite sequence of states, sojourn times, and transitions. A finite path is such a sequence which is finite and ends in a state, i. e. $\pi \in (V \times \mathbb{R}_{\geq 0} \times T)^\star \times V$. We usually write $v \xrightarrow{t,\mathrm{tr}}$ instead of $(v, t, \mathrm{tr}) \in (V \times \mathbb{R}_{\geq 0} \times T)$. We use $\mathrm{Paths}^{\mathrm{fin}}$ and $\mathrm{Paths}^{\mathrm{inf}}$ to denote the set of finite and infinite paths, respectively. The length $|\pi|$ of a path π is ∞ if π is infinite, and equal to the number of transitions on π if π is finite. The last state of a finite path π is denoted by $\mathrm{last}(\pi)$. Given a finite or infinite path $\pi = v_0 \xrightarrow{t_0,\mathrm{tr}_0} v_1 \xrightarrow{t_1,\mathrm{tr}_1} \cdots$ and $0 \leq i < |\pi|$, v_i is the $(i+1)$-th state of π, denoted by $\pi[i]$; t_i is the time of staying at v_i, denoted by $\mathrm{time}(\pi[i])$; and $\mathrm{trans}(\pi[i]) = \mathrm{tr}_i$ is the executed transition at v_i. Note that v_i is left instantaneously, i. e. $\mathrm{time}(\pi[i]) = 0$, if $\mathrm{trans}(\pi[i])$ has an infinite rate. For $0 \leq i \leq j \leq |\pi|$, the sub-path $v_i \xrightarrow{t_i,\mathrm{tr}_i} \cdots v_j$ is denoted by $\pi[i \cdot\cdot j]$.

Strategies. The nondeterminism that may occur at a state is resolved by functions, which are called *strategies* (or policies or schedulers). Each player follows her own strategy in order to accomplish her goal. A strategy of player i $(i = 1, 2)$ is a function $\sigma_i : V_i \times \mathbb{R}_{\geq 0} \to T$ such that $\sigma_i(v, t) \in T(v)$ for all $v \in V$ and $t \in \mathbb{R}_{\geq 0}$.

This strategy class is called *early total-time dependent positional deterministic* (ETTPD), since it uses the total time which has passed since the start of the system and the current state to make its choice, and returns a fixed outgoing transition. Early (in contrast to late) [26] means that the decision which transition to take has to be made when entering a state and may not be changed while residing in the state. ETTPD strategies can be easily extended to the more general *early total-cost dependent positional deterministic* (ETCPD) strategies, where the role of time is taken by costs. There are yet more general classes of early strategies whose decision may depend, e. g. on the whole history since the start of the system, and they may return a probability distribution over the available transitions instead of a fixed transition. However, one can show for the property classes we consider in this paper, that the supremum (and infimum) over ETCPD strategies coincides with the supremum (infimum, respectively) over this more general strategy class [25, 14, 15]. We denote the set of all ETCPD strategies of player i that are measurable in cost by Σ_i.

Probability measure. Given strategies σ_1, σ_2 for both players and a state $v \in V$, a probability space on the set of infinite paths starting in v can be constructed. The set of measurable events is thereby the σ-algebra that is induced by a standard cylinder set construction [2] together with a unique probability measure $\mathrm{Pr}_{v,\sigma_1,\sigma_2}$ on the events. $\mathrm{Pr}_{v,\sigma_1,\sigma_2}(\Pi)$ is the probability of the set of paths Π, starting from state v, given that player 1 and player 2 play with strategies σ_1 and σ_2, respectively. Both the σ-algebra and the probability measure are constructed by extending the existing techniques used for MA and IMCs. We omit the details here; for more information see, e. g. [21, 25, 23].

Zenoness. It may happen that an SG contains an end component [5, Def. 10.117] consisting of probabilistic transitions only. Such an end component leads to the existence of sets of infinite paths π with finite sojourn times and non-zero probability, i. e. $\lim_{n\to\infty} \sum_{i=0}^{n} \mathrm{time}(\pi[i]) < \infty$. This phenomenon is known as *Zenoness*. Since such behaviour has to be considered unrealistic, we assume that the SGs under consideration are non-Zeno, i. e. that they do not contain such end components. Formally, an SG is non-Zeno iff

$$\mathrm{Pr}_{v,\sigma_1,\sigma_2}\big(\{\pi \in \mathrm{Paths}^{\mathrm{inf}} : \lim_{n\to\infty} \sum_{i=0}^{n} \mathrm{time}(\pi[i]) < \infty\}\big) = 0$$

holds for all states $v \in V$ and all strategies $\sigma_1 \in \Sigma_1$ and $\sigma_2 \in \Sigma_2$.

For more on strategies and on SGs in general we refer to [29, 8].

Costs and rewards. We now extend stochastic games by costs and rewards to analyse properties like "What is the maximal reward one can earn when the accumulated cost is bounded by b?"

Definition 3 (Cost and reward structures). *Let $\mathcal{G}$ be a stochastic game as above. A* cost function $c : T \to \mathbb{R}_{\geq 0}$ *assigns a non-negative cost rate to* *each transition. A* reward structure ρ *is a triple* $\rho = (\rho_\mathrm{t}, \rho_\mathrm{i}, \rho_\mathrm{f})$ *of functions* $\rho_\mathrm{t}, \rho_\mathrm{i} : T \to \mathbb{R}_{\geq 0}$*, and* $\rho_\mathrm{f} : V \to \mathbb{R}_{\geq 0}$*;* ρ_t *is the transient reward rate,* ρ_i *the instantaneous reward, and* ρ_f *the final reward.*

For a transition $\mathrm{tr} = (v, \lambda, \mu) \in T$, costs and transient rewards are granted per time unit, i. e. residing in v for t time units before taking transition tr causes a cost of $t \cdot c(\mathrm{tr})$, and a transient reward of $t \cdot \rho_\mathrm{t}(\mathrm{tr})$ is granted. In contrast, the instantaneous reward $\rho_\mathrm{i}(\mathrm{tr})$ is granted for taking the transition tr. The final reward is granted for the state reached when the maximal cost has been spent. This allows, e. g. to consider cost-bounded reachability probabilities as a special case of expected cost-bounded rewards (for more details, see below).

Please note that we do not consider instantaneous costs in this paper. They would render the transformation in Sec. 3 impossible, since there is no instantaneous time. In principle, adapting the analysis algorithm for time-bounded rewards [19, 7] to cost bounds should be possible. That algorithm is based on discretising the time interval, yielding a discrete-time probabilistic game. However, analysing cost-bounded properties for discrete-time models is expensive, even more so as we have to support non-integer costs [1].

Cost and reward of paths. Given a finite path $\pi^\mathrm{fin} = v_0 \xrightarrow{t_0,\mathrm{tr}_0} v_1 \xrightarrow{t_1,\mathrm{tr}_1} \cdots v_{n-1} \xrightarrow{t_{n-1},\mathrm{tr}_{n-1}} v_n$, its cost is defined as $\mathrm{cost}(\pi^\mathrm{fin}) := \sum_{i=0}^{n-1} c(\mathrm{tr}_i) \cdot t_i$. The cost can be extended for an infinite path $\pi = v_0 \xrightarrow{t_0,\mathrm{tr}_0} v_1 \xrightarrow{t_1,\mathrm{tr}_1} \cdots$ by $\mathrm{cost}(\pi) := \lim_{n\to\infty} \mathrm{cost}(\pi[0 \cdot\cdot n])$. The cumulative reward of a finite and an infinite path can be defined in a similar way, i. e. $\mathrm{crew}(\pi^\mathrm{fin}) := \sum_{i=0}^{n-1} \Big(\rho_\mathrm{t}(\mathrm{tr}_i) \cdot t_i + \rho_\mathrm{i}(\mathrm{tr}_i)\Big)$ and $\mathrm{crew}(\pi) := \lim_{n\to\infty} \mathrm{crew}(\pi[0 \cdot\cdot n])$. Furthermore we define the *cost-bounded reward* of π by

$$\mathrm{cbr}^{\mathcal{G}}_{\rho,c}(\pi, b) := \begin{cases} \mathrm{crew}(\pi), & \text{if } \mathrm{cost}(\pi) \leq b, \\ \mathrm{crew}(\pi[0 \cdot\cdot n^*]) + \frac{b - \mathrm{cost}(\pi[0 \cdot\cdot n^*])}{c(\mathrm{tr}_{n^*})} \cdot \rho_\mathrm{t}(\mathrm{tr}_{n^*}) & \\ \qquad + \rho_\mathrm{f}(\pi[n^*]), & \text{otherwise,} \end{cases}$$

where $n^* \in \mathbb{N}$ is the index of the state along path π such that $\mathrm{cost}(\pi[0 \cdot\cdot n^*]) \leq b$ and $\mathrm{cost}(\pi[0 \cdot\cdot n^* + 1]) > b$. More precisely, the cost exceeds b after residing $\frac{b - \mathrm{cost}(\pi[0 \cdot\cdot n^*])}{c(\mathrm{tr}_{n^*})}$ time units in the n^*-th state of the path, and thereby the state is subject to the final reward. Note that such an index exists, provided that $\mathrm{cost}(\pi) > b$.

Example 2. Consider again the stochastic game in Fig. 1(a). We extend it by the cost function and reward structure shown in Fig. 1(b). Now consider the path $\pi = v_0 \xrightarrow{3,\mathrm{tr}_1} v_1 \xrightarrow{0,\mathrm{tr}_3} v_3 \xrightarrow{2,\mathrm{tr}_5} v_2 \xrightarrow{0,\mathrm{tr}_4} v_0 \to \cdots$ and assume the cost bound $b = 20$. The cost incurring in v_0 before taking tr_1 is $5 \cdot 3 = 15$. Since tr_3 is probabilistic, no cost incurs in v_1. In v_3 we have costs $3 \cdot 2 = 6$. Therefore the cost bound is reached while staying in v_3, after $1/3 \cdot (20 - 15) = 5/3$ time units. We then have $n^* = 2$. Since v_3 is the state in which the cost bound is reached, we additionally get its final reward $\rho_\mathrm{f}(v_3) = 3$. The cost-bounded reward for this path is accordingly $\mathrm{cbr}^{\mathcal{G}}_{\rho,c}(\pi, 20) = (3 \cdot 1 + 4) + (0 \cdot 0 + 1) + (5/3 \cdot 1) + 3 = 12\,2/3$. □

Given strategies $\sigma_1 \in \Sigma_1$ and $\sigma_2 \in \Sigma_2$ we can define the *expected cost-bounded reward* (ECR) as the expectation of cbr:

$$\mathbb{E}\mathrm{cbr}^{\sigma_1,\sigma_2}_{\mathcal{G},\rho,c}(v,b) := \int\limits_{\pi \in \mathrm{Paths}^{\mathrm{inf}}(v)} \mathrm{cbr}^{\mathcal{G}}_{\rho,c}(\pi,b)\,\mathrm{dPr}_{v,\sigma_1,\sigma_2}(\pi)\,.$$

The two players can independently try to maximise or minimise the reward earned until the cost bound is reached. Hence, for $\mathrm{opt}_1, \mathrm{opt}_2 \in \{\inf, \sup\}$ we define the *optimal* expected cost-bounded reward by

$$\mathbb{E}\mathrm{cbr}^{\mathrm{opt}_1,\mathrm{opt}_2}_{\mathcal{G},\rho,c}(v,b) := \mathop{\mathrm{opt}_1}_{\sigma_1\in\Sigma_1} \mathop{\mathrm{opt}_2}_{\sigma_2\in\Sigma_2} \mathbb{E}\mathrm{cbr}^{\sigma_1,\sigma_2}_{\mathcal{G},\rho,c}(v,b)\,.$$

Two important classes of properties can be considered as special cases of expected cost-bounded rewards:

For *time-bounded rewards*, denoted by random variable tbr, the time is limited during which reward is collected. This corresponds to using the constant **1**-function as cost. We therefor define $\mathbb{E}\mathrm{tbr}^{\sigma_1,\sigma_2}_{\mathcal{G},\rho}(v,b) := \mathbb{E}\mathrm{cbr}^{\sigma_1,\sigma_2}_{\mathcal{G},\rho,\mathbf{1}}(v,b)$.

The second class encompasses *cost-bounded reachability probabilities*, i. e. questions like "What is the maximal probability to reach a set $V_{\mathrm{goal}} \subseteq V$ of states with cost $\leq b$?". We first make the states in V_{goal} absorbing and add a Markovian self-loop $\mathrm{tr}_v = (v, \lambda, \{v \mapsto 1\})$ with arbitrary finite rate $0 < \lambda < \infty$ to each state $v \in V_{\mathrm{goal}}$ and define the final reward by $\rho_{\mathrm{f}}(v) = 1$ if $v \in V_{\mathrm{goal}}$, and $\rho_{\mathrm{f}}(v) = 0$ otherwise. The transient and instantaneous rewards are constantly 0. Then the expected reward until cost b is reached corresponds to the probability of reaching V_{goal} with costs $\leq b$.

Algorithms to compute optimal expected time-bounded rewards are available both for Markov automata [19] and stochastic games [7]. To the best of our knowledge, up to now there are no algorithms available to compute the optimal expected cost-bounded rewards for MA and SG.

3 Transformation of Stochastic Games

In this section, we first give a fixed point characterisation of expected cost-bounded rewards for stochastic games and prove its correctness. Similar to time-bounded properties [7], this fixed point characterisation is not amenable to an efficient solution. Therefore we transform the stochastic game so that the optimal expected cost-bounded reward coincides with the optimal expected time-bounded reward in the transformed game. This allows us to apply arbitrary algorithms like [19, 7] for expected time-bounded rewards to compute optimal expected cost-bounded rewards.

Theorem 1 (Fixed point characterisation). *Let $\mathcal{G}$ be a stochastic game with cost function c and reward structure $\rho = (\rho_{\mathrm{t}}, \rho_{\mathrm{i}}, \rho_{\mathrm{f}})$. Let $b \in \mathbb{R}_{\geq 0}$ be a cost bound, $\mathrm{opt}_1, \mathrm{opt}_2 \in \{\inf, \sup\}$, and $\mathrm{opt}_{[v]} = \mathrm{opt}_i$ if $v \in V_i$. Then, $\mathbb{E}\mathrm{cbr}^{\mathrm{opt}_1,\mathrm{opt}_2}_{\mathcal{G},\rho,c}(v,b)$ is* *the least fixed point of the higher-order operator* $\Omega_{\mathrm{opt}_1,\mathrm{opt}_2} : (V \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}) \to (V \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0})$*, such that*

$$\Omega_{\mathrm{opt}_1,\mathrm{opt}_2}(F)(v,b) =$$
$$\operatorname*{opt}_{\substack{[v]\\ \mathrm{tr}\in T(v)}} \begin{cases} \displaystyle\int_0^{b/c(\mathrm{tr})} \lambda_{\mathrm{tr}} \cdot \mathrm{e}^{-\lambda_{\mathrm{tr}}\cdot t} \cdot \sum_{v'\in V} \mu_{\mathrm{tr}}(v') \cdot F\big(v', b - c(\mathrm{tr})\cdot t\big)\,\mathrm{d}t & \\ \quad + \Big(\frac{\rho_{\mathrm{t}}(\mathrm{tr})}{\lambda_{\mathrm{tr}}} + \rho_{\mathrm{i}}(\mathrm{tr})\Big) \cdot \Big(1 - \mathrm{e}^{-\frac{\lambda_{\mathrm{tr}}\cdot b}{c(\mathrm{tr})}}\Big) + \rho_{\mathrm{f}}(v) \cdot \mathrm{e}^{-\frac{\lambda_{\mathrm{tr}}\cdot b}{c(\mathrm{tr})}}, & \\ & \textit{if } \mathrm{tr} \in T_{\mathrm{M}}(v) \wedge c(\mathrm{tr}) > 0 \wedge b > 0, \\ \frac{\rho_{\mathrm{t}}(\mathrm{tr})}{\lambda_{\mathrm{tr}}} + \rho_{\mathrm{i}}(\mathrm{tr}) + \displaystyle\sum_{v'\in V} \mu_{\mathrm{tr}}(v') \cdot F(v', b), & \textit{if } \mathrm{tr} \in T_{\mathrm{M}}(v) \wedge c(\mathrm{tr}) = 0, \\ \rho_{\mathrm{i}}(\mathrm{tr}) + \displaystyle\sum_{v'\in V} \mu_{\mathrm{tr}}(v') \cdot F(v', b), & \textit{if } \mathrm{tr} \in T_{\mathrm{P}}(v), \\ \rho_{\mathrm{f}}(v), & \textit{otherwise.} \end{cases}$$

"Least" means in this context that $\forall v \in V, b \in \mathbb{R}_{\geq 0} : \mathrm{Ecbr}^{\mathrm{opt}_1,\mathrm{opt}_2}_{\mathcal{G},\rho,c}(v,b) \leq F(v,b)$, with F being another fixed point of $\Omega_{\mathrm{opt}_1,\mathrm{opt}_2}$.

The fixed point characterisation of expected cost-bounded rewards yields a system of integral equations, which are typically hard to solve. Instead, the following transformation turns cost-bounded rewards into time-bounded rewards. For the latter, not only a fixed point characterisation is available [7], but also a more efficient algorithm, based on discretisation [19, 7].

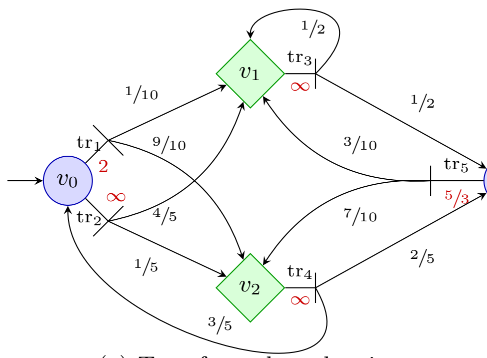
Definition 4 (Cost-to-time transformation). *Let* $\mathcal{G} = \big(V, (V_1, V_2), v_{\mathrm{init}}, T\big)$ *be a stochastic game with cost function* $c : T \to \mathbb{R}_{\geq 0}$ *and reward structure* $\rho = (\rho_{\mathrm{t}}, \rho_{\mathrm{i}}, \rho_{\mathrm{f}})$*. We define the* cost-transformed game $\mathcal{G}^c = \big(V, (V_1, V_2), v_{\mathrm{init}}, T^c\big)$ *with*

$$\begin{aligned} T^c = &\big\{\mathrm{tr} \in T \,\big|\, \lambda_{\mathrm{tr}} = \infty\big\} \\ &\cup \big\{(v, \infty, \mu) \,\big|\, \exists \lambda \in \mathbb{R}_{\geq 0} : \mathrm{tr} = (v, \lambda, \mu) \in T \wedge c(\mathrm{tr}) = 0\big\} \\ &\cup \big\{(v, \lambda/c(\mathrm{tr}), \mu) \,\big|\, \mathrm{tr} = (v, \lambda, \mu) \in T \wedge c(\mathrm{tr}) \neq 0\big\}\,. \end{aligned}$$

and reward structure $\rho^c = (\rho^c_{\mathrm{t}}, \rho^c_{\mathrm{i}}, \rho^c_{\mathrm{f}})$ *such that* $\rho^c_{\mathrm{f}} = \rho_{\mathrm{f}}$*,*

$$\rho^c_{\mathrm{t}}(\mathrm{tr}) = \begin{cases} \rho_{\mathrm{t}}(\mathrm{tr})/c(\mathrm{tr}), & \textit{if } c(\mathrm{tr}) \neq 0, \\ 0, & \textit{if } c(\mathrm{tr}) = 0, \textit{ and} \end{cases}$$
$$\rho^c_{\mathrm{i}}(\mathrm{tr}) = \begin{cases} \rho_{\mathrm{i}}(\mathrm{tr}) + \rho_{\mathrm{t}}(\mathrm{tr})/\lambda_{\mathrm{tr}}, & \textit{if } c(\mathrm{tr}) = 0 \wedge \lambda_{\mathrm{tr}} < \infty, \\ \rho_{\mathrm{i}}(\mathrm{tr}), & \textit{otherwise.} \end{cases}$$

The motivation behind this transformation is as follows: Since we want to transform the cost bound b into a time bound we have to divide b through the cost gained per time unit. This is done by dividing the rate λ of a Markovian transition $\mathrm{tr} \in T_{\mathrm{M}}$ through its cost $c(\mathrm{tr})$. The same has to be done with the transient reward $\rho_{\mathrm{t}}(\mathrm{tr})$. If tr has no cost, i. e. $c(\mathrm{tr}) = 0$, the transition is transformed into a probabilistic transition. The expected transient reward $\rho_{\mathrm{t}}(\mathrm{tr})/\lambda_{\mathrm{tr}}$ has to be added to the instantaneous reward of the transition in this case.



(a) Transformed stochastic game

	tr_1	tr_2	tr_3	tr_4	tr_5	v_0	v_1	v_2	v_3
ρ_{t}^c	$1/5$	0	0	0	$1/3$				
ρ_{i}^c	4	4	1	5	2				
ρ_{f}^c						1	0	2	3

(b) Transformed rewards

Fig. 2. Fig. 1 after transformation

The transformation does not change the structure or size of the SG, and the transformed system is an SG as well. Additionally, Markov automata are closed under this transformation, i. e. if the original SG is actually an MA, so is the transformed system.

Example 3. Consider again the stochastic game in Fig. 1(a) with the costs and rewards in Fig. 1(b). We assume a cost bound of $b = 20$. Then the rewards of the five transitions after transformation are shown in Fig. 2(b). The Markovian transitions tr_1, tr_2, and tr_5 are modified as follows. Transitions tr_3 and tr_4 remain unchanged as they are probabilistic. The expected residence time before taking tr_1 is scaled such that it matches the expected cost in the original game, i. e. the new exit rate becomes $\lambda_{\mathrm{tr}_1}/c(\mathrm{tr}_1) = 10/5 = 2$. The transient reward rate is adjusted accordingly and becomes $\rho_{\mathrm{t}}(\mathrm{tr}_1)/c(\mathrm{tr}_1) = 1/5$. The instantaneous reward does not change. The transition tr_5 is modified in the same way. As the cost of tr_2 is zero, tr_2 becomes probabilistic and the expected reward $\rho_{\mathrm{t}}(\mathrm{tr}_2)/\lambda_{\mathrm{tr}_2}$ earned in v_1 until tr_2 being taken is added to the instantaneous reward of tr_2. The stochastic game after the transformation is shown in Fig. 2(a). □

Theorem 2 (Measure preservation). *Let $\mathcal{G}$ be a stochastic game with reward structure ρ, cost function c, cost bound $b \in \mathbb{R}_{\geq 0}$, $v \in V$, and $\mathrm{opt}_1, \mathrm{opt}_2 \in \{\inf, \sup\}$. Then we have*

$$\mathbb{E}\mathrm{cbr}_{\mathcal{G},\rho,c}^{\mathrm{opt}_1,\mathrm{opt}_2}(v,b) = \mathbb{E}\mathrm{tbr}_{\mathcal{G}^c,\rho^c}^{\mathrm{opt}_1,\mathrm{opt}_2}(v,b)\,.$$

Proof. Here we sketch the proof of the theorem. It is done by showing that the original and the transformed games have indeed the same fixed point characterisation for the respective objectives. For this, on the one hand, we construct the fixed point characterisation of the transformed game using Theorem 1 by assigning the constant cost of one to all Markovian transitions. On the other hand, we reinterpret the representation of the fixed point characterisation of the original model by a series of sound variable substitutions, partly inspired by the transformation. At the end we conclude that both of the fixed point characterisations are the same, and thereby their least fixed points are exactly equal. For more details, see the complete proof in [20]. □

Zero-cost transitions[3] in the original game can introduce Zenoness in the transformed game. It happens if a set of such transitions constitutes an end component in the transformed game. This will be problematic for the analysis, in particular if the end component contains positive rewards. Therefore the strategy that keeps the control of the game inside the end component delivers infinite expected rewards, since staying there gains reward without any cost. Nevertheless the analysis may ignore such a strategy in some cases, for instance in analysis of MA against minimal expected ECR. By any means and for simplicity we exclude such models from our analysis technique.

4 Case Studies and Experimental Results

For our experiments we used the following case studies:
(1) The *Dynamic Power Management System* (DPMS) [27] describes the following scenario: A service requester generates tasks which are stored within a queue until they are handled by a processor. This processor (P) can either be "busy" with processing a job, "idle" while the queue is empty, in a "standby" mode, or in a "sleep" mode. In the latter two modes P is inactive and cannot handle tasks. The change between "busy" and "idle" occurs automatically, depending on whether there are tasks in the queue or not. If P has been "idle" for some time, it is switched into "standby" or "sleep" by a power manager. The power manager is also responsible for switching from these two modes back to "idle". P consumes the least power in "standby" and "sleep" (0.35 W and 0.13 W, respectively), whereas it consumes more power while "idle" (0.95 W) and the most if it is "busy" (2.15 W) [27, 30]. We model the DPMS as an MRA with the costs representing the power consumption of P. The reward corresponds to the number of served tasks. For our experiments we varied the number of different task types (T) and the size of the queue (Q). We explore the expected cost-bounded reward. The model instances are denoted as "DPMS-T-Q".
(2) The *Queueing System* (QS) [21] stores requests of T different types into two queues of size Q each. A server is attached to each queue, which fetches requests from its corresponding queue, and then processes them. One of the servers might insert, with probability 0.1, the already served request into the other queue to be reprocessed by the other server. Power is consumed by both servers when they are processing. We compute the minimum and the maximum number of processed requests under different energy budgets. The model instances are denoted as "QS-T-Q".
(3) The *Polling System* (PS) [17, 32] consists of S station(s) and one server. Each station comes with a queue of size Q, and buffers incoming jobs of T different

[3] Note that the cost of probabilistic transitions is implicitly zero as the delay until taking such transitions is zero.

Table 1. Expected reward in the dynamic power managment system

name	#states	budget = 10 min	budget = 10 max	budget = 20 min	budget = 20 max	budget = 50 min	budget = 50 max
DPMS-2-5	508	0.759	0.859	1.557	1.924	3.910	5.150
DPMS-2-10	1,588	0.759	0.859	1.557	1.924	3.910	5.150
DPMS-2-20	5,548	0.759	0.859	1.557	1.924	3.910	5.150
DPMS-3-5	5,190	0.785	0.883	1.617	1.930	4.129	5.088
DPMS-3-10	29,530	0.785	0.883	1.617	1.930	4.129	5.088
DPMS-3-20	195,810	0.785	0.883	1.617	1.930	4.129	5.088
DPMS-4-5	47,528	0.784	0.877	1.617	1.889	4.143	4.936
DPMS-4-10	492,478	0.784	0.877	1.617	1.889	4.143	4.936

Table 2. Expected reward of the queueing system

name	#states	budget = 1 min	budget = 1 max	budget = 5 min	budget = 5 max	budget = 10 min	budget = 10 max
QS-2-4	46,234	0.249	0.857	1.294	4.078	2.634	7.975
QS-2-5	191,258	0.249	0.857	1.294	4.078	2.634	7.975
QS-2-6	777,754	0.249	0.857	1.294	4.078	2.634	7.975
QS-3-3	117,532	0.125	0.857	0.649	4.078	1.332	7.972
QS-3-4	1,080,865	0.125	0.857	0.649	4.078	1.332	7.972
QS-4-2	42,616	0.125	1.287	0.649	6.127	1.333	12.075
QS-4-3	708,088	0.125	1.287	0.649	6.127	1.333	12.075
QS-6-2	266,974	0.084	1.713	0.433	8.187	0.892	16.201

types. The jobs are then polled and processed by the server. There is a probability of 0.1 for a job to be processed while erroneously remaining in the queue. Each job brings an instantaneous reward when it is completely processed by the server. Whenever processing, the server consumes energy. The model is subject to two kinds of analysis: First we compute the minimum and the maximum probability of encountering the error under some energy budget. The second analysis is on the computation of the minimum and the maximum expected energy bounded reward of the model. The instances of the polling system are denoted as "PS-S-T-Q".

(4) The *Stochastic Job Scheduling* benchmark (SJS) [9] originally stems from economy. In this setting, a number of jobs with different service rates are distributed between processors. Each processor consumes resources, e. g. energy which has to be paid for. The costs in our model represent these expenses. The goal is to have all jobs processed within a certain cost budget. In our experiments we explore the reachability of this goal with homogeneous costs ("all processors have the same costs") and heterogeneous costs ("all processors have different costs"), while varying the number of jobs (M) and the number of processors (N). Since the system degenerates to a CTMC if the service rates are homogeneous, we do not consider this case. The model instances are denoted as "SJS-N-M".

We used SCOOP [31] to create the model files. The transformation from cost to time was done with a python script; the computation time for this

Table 3. Results for the polling system

name	#states	rechability min	rechability max	reward min	reward max
PS-2-2-2	455	0.743	0.773	3.128	3.219
PS-2-2-3	2,055	0.483	0.551	3.980	4.117
PS-2-3-2	2,392	0.995	0.996	1.209	1.253
PS-2-3-3	22,480	0.973	0.983	1.730	1.848
PS-3-2-2	3,577	0.888	0.917	2.549	2.685
PS-3-2-3	34,425	0.665	0.760	3.493	3.732
PS-3-3-2	35,659	1.000	1.000	0.918	0.965
PS-4-2-2	27,783	0.955	0.973	2.166	2.307
PS-4-2-3	570,375	0.793	0.879	3.116	3.403
PS-5-2-2	213,689	0.983	0.992	1.908	2.039

Table 4. Rechability in the stochastic job scheduling benchmark

name	#states	homogeneous costs min	homogeneous costs max	heterogeneous costs min	heterogeneous costs max
SJS-2-4	464	0.241	0.241	0.186	0.243
SJS-2-6	4,144	0.041	0.041	0.021	0.029
SJS-2-8	29,344	0.004	0.004	0.001	0.002
SJS-4-4	3,168	0.241	0.241	0.120	0.610
SJS-4-6	71,644	0.041	0.041	0.013	0.130
SJS-4-8	1,032,272	0.004	0.004	0.001	0.012
SJS-6-4	13,924	0.241	0.241	0.059	0.945
SJS-6-6	685,774	0.041	0.041	0.005	0.374
SJS-8-4	41,552	0.241	0.241	0.033	0.999
SJS-10-4	98,436	0.241	0.241	0.019	1.000

was negligible. We then employed the tool IMCA [16, 17, 19] to determine the minimum and maximum expected cost-bounded reward or the minimum and maximum cost-bounded reachability of the models. It would be possible to use any other analyser for MA, e. g. MeGARA, the prototype from [7].

All experiments were run on an Intel Xeon quad-core processor with 3.3 GHz per core and 64 GB of memory. We set a time limit of 12 hours. The memory consumption was negligible; all experiments needed less than 300 MB.

We will not give detailed time measurements due to space restrictions, nevertheless we want to briefly discuss the computation times. The shortest computations took only fractions of a second, e. g. the computation of the minimum reachability for SJS-2-4 with cost budget 5 took 0.06 seconds, whereas the longer computations needed several hours, e. g. for DPMS-4-10 the computation of the minimum reachability with cost budget 50 took almost 11 hours, which was the longest computation time of all our experiments. In general it can be said that larger systems need more time to analyse than smaller systems. The computation time is also influenced by the size of the cost budget. For example, for cost budget

10 the computation of the minimum reachability for DPMS-4-10 took less than 6 min. This is due to the fact that IMCA uses discretisation [17–19] to compute the values; for a larger bound more discretisation steps are needed. There is also an interesting connection between the costs within the system, its maximum rate, and the computation time: The size of a discretisation step depends on the maximum rate of the transformed system. The higher the maximum rate is, the smaller the discretisation step must be chosen in order to satisfy the given accuracy level. For the computation of cost-bounded rewards, this means that the computation time is strongly influenced by the value of $\max\left\{{}^{\lambda_{\mathrm{tr}}}/_{c(\mathrm{tr})} \,\middle|\, \mathrm{tr} \in T_{\mathrm{M}} : c(\mathrm{tr}) > 0\right\}$. For details on the discretisation, see [19, 7].

Tables 1 to 4 show the results of our experiments. The first two columns of each table contain the name of the respective model instance and its number of states.

In case of DPMS (Table 1) and QS (Table 2) we explore the minimum and maximum expected reward under different cost budgets. For DPMS we used cost budgets of 10, 20, and 50, whereas for QS we used cost budgets of 1, 5, and 10 (see the respective blocks in Table 1 and Table 2). It holds for both DPMS and QS that the expected reward grows with the budget, as does the difference between minimum and maximum reward, as to be expected. Another interesting fact is that the size of the queues in the models – while having a big influence on the size of the system – has practically no impact on the expected reward. It is completely determined by the number of different task types. This observation can be explained as follows: For the processing unit of DPMS (or of QS) it is not important how many jobs exactly can be stored in the queue(s), as long as there *are* jobs in the queue(s).

For PS (Table 3) we studied both minimum and maximum reachability and minimum and maximum expected reward (see the respective blocks in the table) under a cost budget of 5. If we increase the queue size, the minimum and maximum probability for encountering the error decreases, while the expected minimum and maximum reward increases. At the same time we can observe that the reachability increases with the number of stations, e. g. for PS-2-2-2, containing two stations, the maximum probability is 0.773, whereas for PS-5-2-2, containing 5 stations, it is 0.992. This makes sense, since the error is caused by the stations and the probability to encounter the error therefore increases with having more stations.

For SJS (Table 4) we also used a cost budget of 5. Here we studied the minimum and maximum reachability while assuming homogeneous or heterogeneous costs for the different processors of the system (see the respective blocks in Table 4). For homogeneous costs we can observe a similar effect as for DPMS and PS: The number of processors influences the number of states in the system, but has a negligible impact on the reachability. The latter is completely determined by the number of jobs. What's more, the minimum and the maximum reachability are the same in this case. These effects vanish if we assume heterogeneous costs. In this case, the distance between minimum and maximum reachability increases, especially the maximum reachability becomes higher. These observations make sense: In case of a homogeneous system it does not matter, which processor handles which job. However, in a heterogeneous system there is a choice between more and less expensive processors which can handle the jobs, which in turn leads to a higher (lower) maximum (minimum) reachability.

5 Conclusion

We studied the computation of Markov automata and stochastic games against cost-bounded reward objectives. In this regard, we provided a fixed point characterisation for the optimal expected cost-bounded reward. Moreover, we proposed an efficient measure-preserving transformation from cost-bounded to time-bounded objectives. For the latter, an analysis technique based on discretisation with strict error bound exists. Our experiments demonstrate the effectiveness of the approach.

In the future, we plan to improve the efficiency of the proposed approach, e. g. via an abstraction/refinement technique on very large games and automata.

References

1. Andova, S., Hermanns, H., Katoen, J.: Discrete-time rewards model-checked. In: Proc. of FORMATS. LNCS, vol. 2791, pp. 88–104. Springer (2003)
2. Ash, R.B., Doléans-Dade, C.A.: Probability & Measure Theory. Academic Press, 2nd edn. (1999)
3. Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.: On the logical characterisation of performability properties. In: Proc. of ICALP. LNCS, vol. 1853, pp. 780–792. Springer (2000)
4. Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.: Reachability in continuous-time Markov reward decision processes. In: Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]. Texts in Logic and Games, vol. 2, pp. 53–72. Amsterdam University Press (2008)
5. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press (2008)
6. Boudali, H., Crouzen, P., Stoelinga, M.: A rigorous, compositional, and extensible framework for dynamic fault tree analysis. IEEE Trans. Dependable Sec. Comput. 7(2), 128–143 (2010)
7. Braitling, B., Ferrer Fioriti, L.M., Hatefi, H., Wimmer, R., Hermanns, H., Becker, B.: Abstraction-based computation of reward measures for Markov automata. In: Proc. of VMCAI. LNCS, vol. 8931, pp. 172–189. Springer (2015)
8. Brázdil, T., Forejt, V., Krcál, J., Kretínský, J., Kucera, A.: Continuous-time stochastic games with time-bounded reachability. Information and Computation 224, 46–70 (2013)
9. Bruno, J.L., Downey, P.J., Frederickson, G.N.: Sequencing tasks with exponential service times to minimize the expected flow time or makespan. Journal of the ACM 28(1), 100–113 (1981)
10. Buchholz, P., Schulz, I.: Numerical analysis of continuous time Markov decision processes over finite horizons. Computers & Operations Research 38(3), 651–659 (2011)

11. Cloth, L., Katoen, J., Khattri, M., Pulungan, R.: Model checking Markov reward models with impulse rewards. In: Proc. of DSN. pp. 722–731. IEEE CS (2005)
12. Eisentraut, C., Hermanns, H., Katoen, J.P., Zhang, L.: A semantics for every GSPN. In: Proc. of Petri Nets. pp. 90–109. No. 7927 in LNCS, Springer (2013)
13. Eisentraut, C., Hermanns, H., Zhang, L.: On probabilistic automata in continuous time. In: Proc. of LICS. pp. 342–351. IEEE CS (2010)
14. Fu, H.: Maximal cost-bounded reachability probability on continuous-time Markov decision processes. In: Proc. of FOSSACS. pp. 73–87 (2014)
15. Fu, H.: Verifying Probabilistic Systems: New Algorithms and Complexity Results. Ph.D. thesis, RWTH Aachen University (2014)
16. Guck, D., Han, T., Katoen, J.P., Neuhäußer, M.R.: Quantitative timed analysis of interactive Markov chains. In: Proc. of NFM. LNCS, vol. 7226, pp. 8–23. Springer (2012)
17. Guck, D., Hatefi, H., Hermanns, H., Katoen, J.P., Timmer, M.: Modelling, reduction and analysis of Markov automata. In: Proc. of QEST. LNCS, vol. 8054, pp. 55–71. Springer (2013)
18. Guck, D., Hatefi, H., Hermanns, H., Katoen, J., Timmer, M.: Analysis of timed and long-run objectives for Markov automata. Logical Methods in Computer Science 10(3) (2014), http://dx.doi.org/10.2168/LMCS-10(3:17)2014
19. Guck, D., Timmer, M., Hatefi, H., Ruijters, E., Stoelinga, M.: Modelling and analysis of Markov reward automata. In: Proc. of ATVA. LNCS, vol. 8837, pp. 168–184. Springer (2014)
20. Hatefi, H., Braitling, B., Wimmer, R., Ferrer Fioriti, L.M., Hermanns, H., Becker, B.: Cost vs. time in stochastic games and Markov automata (extended version). Reports of SFB/TR 14 AVACS 113, SFB/TR 14 AVACS (2015), http://www.avacs.org
21. Hatefi, H., Hermanns, H.: Model checking algorithms for Markov automata. ECE-ASST 53 (2012)
22. Hermanns, H.: Interactive Markov Chains: The Quest for Quantified Quality, LNCS, vol. 2428. Springer (2002)
23. Johr, S.: Model checking compositional Markov systems. Ph.D. thesis, Saarland University, Germany (2008)
24. Miller, B.L.: Finite state continuous time Markov decision processes with a finite planning horizon. SIAM Journal on Control 6(2), 266–280 (1968)
25. Neuhäußer, M.R.: Model checking nondeterministic and randomly timed systems. Ph.D. thesis, RWTH Aachen University and University of Twente (2010)
26. Neuhäußer, M.R., Zhang, L.: Time-bounded reachability probabilities in continuous-time Markov decision processes. In: Proc. of QEST. pp. 209–218. IEEE CS (2010)
27. Qiu, Q., Qu, Q., Pedram, M.: Stochastic modeling of a power-managed system-construction and optimization. IEEE Transactions on CAD of Integrated Circuits and Systems 20(10), 1200–1217 (2001)
28. Segala, R.: A compositional trace-based semantics for probabilistic automata. In: Proc. of CONCUR. LNCS, vol. 962, pp. 234–248. Springer (1995)
29. Shapley, L.S.: Stochastic games. Proc. of the National Academy of Sciences of the United States of America 39(10), 1095 (1953)
30. Simunic, T., Benini, L., Glynn, P.W., Micheli, G.D.: Dynamic power management for portable systems. In: Proc. of MOBICOM. pp. 11–19 (2000)
31. Timmer, M., Katoen, J.P., van de Pol, J., Stoelinga, M.: Efficient modelling and generation of Markov automata. In: Proc. of CONCUR. LNCS, vol. 7454, pp. 364–379. Springer (2012)
32. Timmer, M., van de Pol, J., Stoelinga, M.: Confluence reduction for Markov automata. In: Proc. of FORMATS. LNCS, vol. 8053, pp. 243–257. Springer (2013)