RWTH Aachen

Department of Computer Science
Technical Report

Quantitative Timed Analysis of Interactive Markov Chains

Dennis Guck, Tingting Han,
Joost-Pieter Katoen and Martin R. Neuhäußer

ISSN 0935–3232 · Aachener Informatik-Berichte · AIB-2012-09

RWTH Aachen · Department of Computer Science · May 2012

Quantitative Timed Analysis of Interactive Markov Chains

Dennis Guck[1], Tingting Han[2],
Joost-Pieter Katoen[1], and Martin R. Neuhäußer[3]

[1] RWTH Aachen University, Germany
[2] University of Oxford, UK
[3] Saarland University, Germany

Abstract. This paper presents new algorithms and accompanying tool support for analyzing interactive Markov chains (IMCs), a stochastic timed $1\frac{1}{2}$-player game in which delays are exponentially distributed. IMCs are compositional and act as semantic model for engineering formalisms such as AADL and dynamic fault trees. We provide algorithms for determining the extremal expected time of reaching a set of states, and the long-run average of time spent in a set of states. The prototypical tool IMCA supports these algorithms as well as the synthesis of ε-optimal piecewise constant timed policies for timed reachability objectives. Two case studies show the feasibility and scalability of the algorithms.

1 Introduction

Continuous-time Markov chains (CTMCs) are perhaps the most well-studied stochastic model in performance evaluation and naturally reflect the random real-time behavior of stoichiometric equations in systems biology. LTSs (labeled transition systems) are one of the main operational models for concurrency and are equipped with a plethora of behavioral equivalences like bisimulation and trace equivalences. A natural mixture of CTMCs and LTSs yields so-called *interactive Markov chains* (IMCs), originally proposed as a semantic model of stochastic process algebras [19,20]. As a state may have several outgoing action-transitions, IMCs are in fact stochastic real-time $1\frac{1}{2}$-player games, also called continuous-time probabilistic automata by Knast in the 1960's [22].

IMC usage. The simplicity of IMCs and their compositional nature —they are closed under CSP-like parallel composition and restriction— make them attractive to act as a semantic backbone of several formalisms. IMCs were developed for stochastic process algebras [19]. Dynamic fault trees are used in reliability engineering for safety analysis purposes and specify the causal relationship between failure occurrences. If failures occur according to an exponential distribution, which is quite a common assumption in reliability analysis, dynamic fault trees are in fact IMCs [4]. The same holds for the standardized Architectural Analysis and Design Language (AADL) in which nominal system behavior is extended with probabilistic error models. IMCs turn out to be a natural semantic model for AADL [5]; the use of this connection in the aerospace domain has recently been shown in [29]. In addition, IMCs are used for stochastic extensions of Statemate [3], and for modeling and analysing industrial GALS hardware designs [12].

IMC analysis. The main usage of IMCs so far has been the compositional generation and minimization of models. Its analysis has mainly been restricted to "fully probabilistic" IMCs which induce CTMCs and are therefore amenable to standard Markov chain analysis or, alternatively, model checking [1]. CTMCs can sometimes be obtained from IMCs by applying weak bisimulation minimization; however, if this does not suffice, semantic restrictions on the IMC level are imposed to ensure full probabilism. The CADP toolbox [11] supports the compositional generation, minimization, and standard CTMC analysis of IMCs. In this paper, we focus on the *quantitative timed analysis* of arbitrary IMCs, in particular of those, that are non-deterministic and can be seen as stochastic real-time $1\frac{1}{2}$-player games. We provide algorithms for the expected time analysis and long-run average fraction of time analysis of IMCs and show how both cases can be reduced to stochastic shortest path (SSP) problems [2,15]. This complements recent work on the approximate time-bounded reachability analysis of IMCs [30]. Our algorithms are presented in detail and proven correct. Prototypical tool support for these analyses is presented that includes an implementation of [30]. The feasibility and scalability of our algorithms are illustrated on two examples: A dependable workstation cluster [18] and a Google file system [10]. Our IMCA tool is a useful backend for the CADP toolbox, as well as for analysis tools for dynamic fault trees and AADL error models.

Related work. Untimed quantitative reachability analysis of IMCs has been handled in [11]; timed reachability in [30]. Other related work is on continuous-time Markov decision processes (CTMDPs). A numerical algorithm for time-bounded expected accumulated rewards in CTMDPs is given in [8] and used as building brick for a CSL model checker in [7]. Algorithms for timed reachability in CTMDPs can be found in, e.g. [6,26]. Long-run averages in stochastic decision processes using observer automata ("experiments") have been treated in [14], whereas the usage of SSP problems for verification originates from [15]. Finally, [27] considers discrete-time Markov decision processes (MDPs) with ratio cost functions; we exploit such objectives for long-run average analysis.

Organization of the paper. Section 2 introduces IMCs. Section 3 and 4 are devoted to the reduction of computing the optimal expected time reachability and long-run average objectives to stochastic shortest path problems. Our tool IMCA and the results of two case studies are presented in Section 5. Section 6 concludes the paper. This version is an update of [17] including additional proofs in the appendix.

2 Interactive Markov chains

Interactive Markov chains. IMCs are finite transition systems with action-labeled transitions and Markovian transitions which are labeled with a positive real number (ranged over by λ) identifying the rate of an exponential distribution.

Definition 1 (Interactive Markov chain). *An* interactive Markov chain *is a tuple $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ where S is a nonempty, finite set of states with* initial state *$s_0 \in S$, Act is a finite set of actions, and*

- $\longrightarrow \subseteq S \times Act \times S$ *is a set of* action *transitions and*
- $\Longrightarrow \subseteq S \times \mathbb{R}_{>0} \times S$ *is a set of* Markovian *transitions.*

We abbreviate $(s, \alpha, s') \in \longrightarrow$ by $s \xrightarrow{\alpha} s'$ and $(s, \lambda, s') \in \Longrightarrow$ by $s \overset{\lambda}{\Longrightarrow} s'$. IMCs are closed under parallel composition [19] by synchronizing on action transitions in a TCSP-like manner. As our main interest is in the analysis of IMCs, we focus on so-called *closed* IMCs [21], i.e. IMCs that are not subject to any further synchronization. W.l.o.g. we assume that in closed IMCs all outgoing action transition of state s are uniquely labeled, thereby naming the state's nondeterministic choices. In the rest of this paper, we only consider closed IMCs. For simplicity, we assume that IMCs do not contain deadlock states, i.e. in any state either an action or a Markovian transition emanates.

Definition 2 (Maximal progress). *In any closed IMC, action transitions take precedence over Markovian transitions.*

The rationale behind the maximal progress assumption is that in closed IMCs, action transitions are not subject to interaction and thus can happen immediately, whereas the probability for a Markovian transition to happen immediately is zero. Accordingly, we assume that each state s has either only outgoing action transitions or only outgoing Markovian transitions. Such states are called interactive and Markovian, respectively; we use $IS \subseteq S$ and $MS \subseteq S$ to denote the sets of interactive and Markovian states. Let $Act(s) = \{\, \alpha \in Act \mid \exists s' \in S.\, s \xrightarrow{\alpha} s' \,\}$ be the set of enabled actions in s, if $s \in IS$ and $Act(s) = \{\perp\}$ if $s \in MS$. In Markovian states, we use the special symbol $\perp$ to denote purely stochastic behavior without any nondeterministic choices.

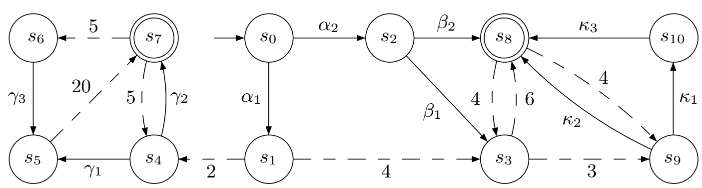

Fig. 1. An example IMC.

Example 1. Fig. 1 depicts an IMC $\mathcal{I}$, where solid and dashed lines represent action and Markovian transitions, respectively. The set of Markovian states is $MS = \{s_1, s_3, s_5, s_7, s_8\}$; IS contains all other states. Nondeterminism between action transitions appears in states s_0, s_2, s_4, and s_9.

A sub-IMC of an IMC $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$, is a pair (S', K) where $S' \subseteq S$ and K is a function that assigns each $s \in S'$ a set $\varnothing \neq K(s) \subseteq Act(s)$ of actions such that for all $\alpha \in K(s)$, $s \xrightarrow{\alpha} s'$ or $s \overset{\lambda}{\Longrightarrow} s'$ imply $s' \in S'$. An *end component* is a sub-IMC whose underlying graph is strongly connected; it is *maximal* w.r.t. K if it is not contained in any other end component (S'', K).

Example 2. In Fig. 1, the sub-IMC (S', K) with state space $S' = \{s_4, s_5, s_6, s_7\}$ and $K(s) = Act(s)$ for all $s \in S'$ is a maximal end component.

IMC semantics. An IMC without action transitions is a CTMC; if $\Longrightarrow$ is empty, then it is an LTS. We briefly explain the semantics of Markovian transitions. Roughly speaking, the meaning of $s \overset{\lambda}{\Longrightarrow} s'$ is that the IMC can switch from state s to s' within d time units with probability $1 - e^{-\lambda d}$. The positive real value λ thus uniquely identifies a negative exponential distribution. For $s \in MS$, let $\mathbf{R}(s,s') = \sum\{\lambda \mid s \overset{\lambda}{\Longrightarrow} s'\}$ be the *rate* to move from state s to state s'. If $\mathbf{R}(s,s') > 0$ for more than one state s', a competition between the transitions of s exists, known as the race condition. The probability to move from such state s to a particular state s' within d time units, i.e. $s \Longrightarrow s'$ wins the race, is

$$\frac{\mathbf{R}(s,s')}{E(s)} \cdot \left(1 - e^{-E(s)d}\right), \tag{1}$$

where $E(s) = \sum_{s' \in S} \mathbf{R}(s,s')$ is the *exit rate* of state s. Intuitively, (1) states that after a delay of at most d time units (second term), the IMC moves probabilistically to a direct successor state s' with discrete branching probability $\mathbf{P}(s,s') = \frac{\mathbf{R}(s,s')}{E(s)}$.

Paths and schedulers. An infinite path π in an IMC is an infinite sequence:

$$\pi \;=\; s_0 \xrightarrow{\sigma_0,t_0} s_1 \xrightarrow{\sigma_1,t_1} s_2 \xrightarrow{\sigma_2,t_2} \ldots$$

with $s_i \in S$, $\sigma_i \in Act$ or $\sigma_i = \perp$, and $t_i \in \mathbb{R}_{\geq 0}$. The occurrence of action α in state s_i in π is denoted $s_i \xrightarrow{\alpha,0} s_{i+1}$; the occurrence of a Markovian transition after t time units delay in s_i is denoted $s_i \xrightarrow{\perp,t} s_{i+1}$. For $t \in \mathbb{R}_{\geq 0}$, let $\pi@t$ denote the set of states that π occupies at time t. Note that $\pi@t$ is in general not a single state, but rather a set of states, as an IMC may exhibit immediate transitions and thus may occupy various states at the same time instant. Let *Paths* and *Paths** denote the sets of infinite and finite paths, respectively.

Nondeterminism appears when there is more than one action transition enabled in a state. The corresponding choice is resolved using *schedulers*. A scheduler (ranged over by D) is a measurable function which yields for each finite path ending in some state s a probability distribution over the set of enabled actions in s. For details, see [30]. A stationary deterministic scheduler is a mapping $D : IS \to Act$. The usual cylinder set construction yields a σ-algebra $\mathfrak{F}_{Paths}$ of subsets of *Paths*; given a scheduler D and an initial state s, $\mathfrak{F}_{Paths}$ can be equipped with a probability measure [30], denoted $\Pr_{s,D}$.

Zenoness. The time elapsed along an infinite path $\pi = s_0 \xrightarrow{\sigma_0,t_0} s_1 \xrightarrow{\sigma_1,t_1} \ldots$ up to state n is $\sum_{i=0}^{n-1} t_i$. Path π is non-Zeno whenever $\sum_{i=0}^{\infty} t_i$ diverges to infinity; accordingly, an IMC $\mathcal{I}$ with initial state s_0 is non-Zeno if for all schedulers D, $\Pr_{s_0,D}\{\pi \in Paths \mid \sum_{i=0}^{\infty} t_i = \infty\} = 1$. As the probability of a Zeno path in a finite CTMC —thus only containing Markovian transitions— is zero [1], IMC $\mathcal{I}$ is non-Zeno if and only if no strongly connected component with states $T \subseteq IS$ is reachable from s_0. In the rest of this paper, we assume IMCs to be non-Zeno.

Stochastic shortest path problems. The (non-negative) SSP problem considers the minimum expected cost for reaching a set of goal states in a discrete-time Markov decision process (MDP).

Definition 3 (MDP). *$\mathcal{M} = (S, Act, \mathbf{P}, s_0)$ is a* Markov decision process*, where S, Act and s_0 are as before and $\mathbf{P} : S \times Act \times S \to [0,1]$ is a* transition probability function *such that for all $s \in S$ and $\alpha \in Act$, $\sum_{s' \in S} \mathbf{P}(s,\alpha,s') \in \{0,1\}$.*

Definition 4 (SSP problem). *A* non-negative stochastic shortest path problem *(SSP problem) is a tuple* $\mathcal{P} = (S, Act, \mathbf{P}, s_0, G, c, g)$*, where* $(S, Act, \mathbf{P}, s_0)$ *is an MDP,* $G \subseteq S$ *is a set of* goal *states,* $c : S \setminus G \times Act \to \mathbb{R}_{\geq 0}$ *is a* cost function *and* $g : G \to \mathbb{R}_{\geq 0}$ *is a* terminal cost function.

The infinite sequence $\pi = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \ldots$ is a path in the MDP if $s_i \in S$ and $\mathbf{P}(s_i, \alpha_i, s_{i+1}) > 0$ for all $i \geqslant 0$. Let k be the smallest index such that $s_k \in G$. The accumulated cost along π of reaching G, denoted $C_G(\pi)$, is $\sum_{j=0}^{k-1} c(s_j, \alpha_j) + g(s_k)$. The minimum expected cost reachability of G starting from s in the SSP $\mathcal{P}$, denoted $cR^{\min}(s, \Diamond\, G)$, is defined as

$$cR^{\min}(s, \Diamond\, G) = \inf_D \mathbb{E}_{s,D}(C_G) = \inf_D \sum_{\pi \in Paths_{abs}} C_G(\pi) \cdot \Pr^{abs}_{s,D}(\pi),$$

where $Paths_{abs}$ denotes the set of (time-abstract) infinite paths in the MDP and $\Pr^{abs}_{s,D}$ the probability measure on sets of MDP paths that is induced by scheduler D and initial state s. The quantity $cR^{\min}(s, \Diamond\, G)$ can be obtained [2,13] by solving the following linear programming problem with variables $\{x_s\}_{s \in S \setminus G}$: maximize $\sum_{s \in S \setminus G} x_s$ subject to the following constraints for each $s \in S \setminus G$ and $\alpha \in Act$:

$$x_s \leqslant c(s, \alpha) + \sum_{s' \in S \setminus G} \mathbf{P}(s, \alpha, s') \cdot x_{s'} + \sum_{s' \in G} \mathbf{P}(s, \alpha, s') \cdot g(s').$$

3 Expected time analysis

Expected time objectives. Let $\mathcal{I}$ be an IMC with state space S and $G \subseteq S$ a set of goal states. Define the (extended) random variable $V_G : Paths \to \mathbb{R}^{\infty}_{\geq 0}$ as the elapsed time before first visiting some state in G, i.e. for infinite path $\pi = s_0 \xrightarrow{\sigma_0, t_0} s_1 \xrightarrow{\sigma_1, t_1} \cdots$, let $V_G(\pi) = \min\{t \in \mathbb{R}_{\geq 0} \mid G \cap \pi@t \neq \varnothing\}$ where $\min(\varnothing) = +\infty$. The minimal expected time to reach G from $s \in S$ is given by

$$eT^{\min}(s, \Diamond\, G) \;=\; \inf_D \mathbb{E}_{s,D}(V_G) \;=\; \inf_D \int_{Paths} V_G(\pi) \; \Pr_{s,D}(d\pi).$$

Note that by definition of V_G, only the amount of time before entering the first G-state is relevant. Hence, we may turn all G-states into absorbing Markovian states without affecting the expected time reachability. Accordingly, we assume for the remainder of this section that for all $s \in G$ and some $\lambda > 0$, $s \overset{\lambda}{\Longrightarrow} s$ is the only outgoing transition of state s.

Theorem 1. *The function* $eT^{\min}$ *is a fixpoint of the Bellman operator*

$$[L(v)](s) = \begin{cases} \dfrac{1}{E(s)} + \displaystyle\sum_{s' \in S} \mathbf{P}(s, s') \cdot v(s') & \text{if } s \in MS \setminus G \\ \displaystyle\min_{s \xrightarrow{\alpha} s'} v(s') & \text{if } s \in IS \setminus G \\ 0 & \text{if } s \in G. \end{cases}$$

Intuitively, Thm. 1 justifies to add the expected sojourn times in all Markovian states before visiting a G-state. Any non-determinism in interactive states (which are, by definition, left instantaneously) is resolved by minimizing the expected reachability time from the reachable one-step successor states.

Computing expected time probabilities. The characterization of $eT^{\min}(s, \Diamond G)$ in Thm. 1 allows us to reduce the problem of computing the minimum expected time reachability in an IMC to a non-negative SSP problem [2,15].

Definition 5 (SSP for minimum expected time reachability). *The SSP of IMC $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ for the expected time reachability of $G \subseteq S$ is $\mathcal{P}_{eT^{\min}}(\mathcal{I}) = (S, Act \cup \{\perp\}, \mathbf{P}, s_0, G, c, g)$ where $g(s) = 0$ for all $s \in G$ and*

$$\mathbf{P}(s, \sigma, s') = \begin{cases} \frac{\mathbf{R}(s,s')}{E(s)} & \textit{if } s \in MS \wedge \sigma = \perp \\ 1 & \textit{if } s \in IS \wedge s \xrightarrow{\sigma} s' \\ 0 & \textit{otherwise, and} \end{cases}$$

$$c(s, \sigma) = \begin{cases} \frac{1}{E(s)} & \textit{if } s \in MS \setminus G \wedge \sigma = \perp \\ 0 & \textit{otherwise.} \end{cases}$$

Intuitively, action transitions are assigned a Dirac distribution, whereas the probabilistic behavior of a Markovian state is as explained before. The reward of a Markovian state is its mean residence time. Terminal costs are set to zero.

Theorem 2 (Correctness of the reduction). *For IMC $\mathcal{I}$ and its induced SSP $\mathcal{P}_{eT^{\min}}(\mathcal{I})$ it holds:*

$$eT^{\min}(s, \Diamond G) = cR^{\min}(s, \Diamond G)$$

where $cR^{\min}(s, \Diamond G)$ denotes the minimal cost reachability of G in SSP $\mathcal{P}_{eT^{\min}}(\mathcal{I})$.

Proof. According to [2,15], $cR^{\min}(s, \Diamond G)$ is the unique fixpoint of the Bellman operator L' defined as:

$$\left[L'(v)\right](s) = \min_{\alpha \in Act(s)} c(s, \alpha) + \sum_{s' \in S \setminus G} \mathbf{P}(s, \alpha, s') \cdot v(s') + \sum_{s' \in G} \mathbf{P}(s, \alpha, s') \cdot g(s').$$

We prove that the Bellman operator L from Thm. 1 equals L' for SSP $\mathcal{P}_{eT^{\min}}(\mathcal{I})$. By definition, it holds that $g(s) = 0$ for all $s \in S$. Thus

$$\left[L'(v)\right](s) = \min_{\alpha \in Act(s)} c(s, \alpha) + \sum_{s' \in S \setminus G} \mathbf{P}(s, \alpha, s') \cdot v(s').$$

For $s \in MS$, $Act(s) = \{\perp\}$; if $s \in G$, then $c(s, \perp) = 0$ and $\mathbf{P}(s, \perp, s) = 1$ imply $L'(v)(s) = 0$. For $s \in IS$ and $\alpha \in Act(s)$, there exists a unique $s' \in S$ such that $\mathbf{P}(s, \alpha, s') = 1$. Thus we can rewrite L' as follows:

$$\left[L'(v)\right](s) = \begin{cases} c(s, \perp) + \sum_{s' \in S \setminus G} \mathbf{P}(s, \perp, s') \cdot v(s') & \text{if } s \in MS \setminus G \\ \min_{s \xrightarrow{\alpha} s'} c(s, \alpha) + v(s') & \text{if } s \in IS \setminus G \\ 0 & \text{if } s \in G. \end{cases} \tag{2}$$

By observing that $c(s, \perp) = \frac{1}{E(s)}$ if $s \in MS \setminus G$ and $c(s, \sigma) = 0$, otherwise, we can rewrite L' in (2) to yield the Bellman operator L as defined in Thm. 1. $\square$

Observe from the fixpoint characterization of $eT^{\min}(s, \Diamond G)$ in Thm. 1 that in interactive states—and only those may exhibit nondeterminism—it suffices to choose the successor state that minimizes $v(s')$. In addition, by Thm. 2, the Bellman operator L from Thm. 1 yields the minimal cost reachability in SSP $\mathcal{P}_{eT^{\min}}(\mathcal{I})$. These two observations and the fact that stationary deterministic policies suffice to attain the minimum expected cost of an SSP [2,15] yields:

Corollary 1. *There is a stationary deterministic scheduler yielding* $eT^{\min}(s, \Diamond G)$.

The uniqueness of the minimum expected cost of an SSP [2,15] now yields:

Corollary 2. $eT^{\min}(s, \Diamond G)$ *is the unique fixpoint of* L *(see Thm. 1).*

The uniqueness result enables the usage of standard solution techniques such as value iteration and linear programming to compute $eT^{\min}(s, \Diamond G)$.

4 Long-run average analysis

Long-run average objectives. Let $\mathcal{I}$ be an IMC with state space S and $G \subseteq S$ a set of goal states. We use $\mathbf{I}_G$ as an indicator with $\mathbf{I}_G(s) = 1$ if $s \in G$ and 0, otherwise. Following the ideas of [14,23], the fraction of time spent in G on an infinite path π in $\mathcal{I}$ up to time bound $t \in \mathbb{R}_{\geq 0}$ is given by the random variable (r. v.) $A_{G,t}(\pi) \;=\; \frac{1}{t}\int_0^t \mathbf{I}_G(\pi@u)\; du$. Taking the limit $t \to \infty$, we obtain the r. v.

$$A_G(\pi) \;=\; \lim_{t\to\infty} A_{G,t}(\pi) \;=\; \lim_{t\to\infty} \frac{1}{t}\int_0^t \mathbf{I}_G(\pi@u)\; du.$$

The expectation of A_G for scheduler D and initial state s yields the corresponding long-run average time spent in G:

$$\mathrm{LRA}^D(s,G) = \mathbb{E}_{s,D}(A_G) = \int_{Paths} A_G(\pi)\,\mathrm{Pr}_{s,D}(d\pi).$$

The minimum long-run average time spent in G starting from state s is then:

$$\mathrm{LRA}^{\min}(s,G) \;=\; \inf_D\; \mathrm{LRA}^D(s,G) \;=\; \inf_D \mathbb{E}_{s,D}(A_G).$$

For the long-run average analysis, we may assume w.l.o.g. that $G \subseteq MS$, as the long-run average time spent in any interactive state is always 0. This claim follows directly from the fact that interactive states are instantaneous, i.e. their sojourn time is 0 by definition. Note that in contrast to the expected time analysis, G-states cannot be made absorbing in the long-run average analysis.

Theorem 3. *There is a stationary deterministic scheduler yielding* $\mathrm{LRA}^{\min}(s,G)$.

In the remainder of this section, we discuss in detail how to compute the minimum long-run average fraction of time to be in G in an IMC $\mathcal{I}$ with initial state s_0. The general idea is the following three-step procedure:

1. Determine the maximal end components $\{\mathcal{I}_1, \ldots, \mathcal{I}_k\}$ of IMC $\mathcal{I}$.
2. Determine $\mathrm{LRA}^{\min}(G)$ in maximal end component $\mathcal{I}_j$ for all $j \in \{1, \ldots, k\}$.
3. Reduce the computation of $\mathrm{LRA}^{\min}(s_0, G)$ in IMC $\mathcal{I}$ to an SSP problem.

The first phase can be performed by a graph-based algorithm [13] which has recently been improved in [9], whereas the last two phases boil down to solving linear programming problems. In the next subsection, we show that determining the LRA in an end component of an IMC can be reduced to a long-run ratio objective in an MDP equipped with two cost functions. Then, we show the reduction of our original problem to an SSP problem.

4.1 Long-run averages in unichain IMCs

In this subsection, we consider computing long-run averages in *unichain* IMCs, i.e. IMCs that under any stationary deterministic scheduler yield a strongly connected graph structure.

Long-run ratio objectives in MDPs. Let $\mathcal{M} = (S, Act, \mathbf{P}, s_0)$ be an MDP. Assume w.l.o.g. that for each state s there exists $\alpha \in Act$ such that $\mathbf{P}(s, \alpha, s') > 0$. Let $c_1, c_2 : S \times (Act \cup \{\perp\}) \to \mathbb{R}_{\geqslant 0}$ be cost functions. The operational interpretation is that a cost $c_1(s, \alpha)$ is incurred when selecting action α in state s, and similar for c_2. Our interest is the ratio between c_1 and c_2 along a path. The *long-run ratio* $\mathcal{R}$ between the accumulated costs c_1 and c_2 along the infinite path $\pi = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$ in the MDP $\mathcal{M}$ is defined by[4]:

$$\mathcal{R}(\pi) \;=\; \lim_{n\to\infty} \frac{\sum_{i=0}^{n-1} c_1(s_i, \alpha_i)}{\sum_{j=0}^{n-1} c_2(s_j, \alpha_j)}.$$

The minimum long-run ratio objective for state s of MDP $\mathcal{M}$ is defined by:

$$R^{\min}(s) \;=\; \inf_D \mathbb{E}_{s,D}(\mathcal{R}) \;=\; \inf_D \sum_{\pi \in Paths_{abs}} \mathcal{R}(\pi) \cdot \Pr^{abs}_{s,D}(\pi).$$

From [13], it follows that $R^{\min}(s)$ can be obtained by solving the following linear programming problem with real variables k and x_s for each $s \in S$: Maximize k subject to the following constraints for each $s \in S$ and $\alpha \in Act$:

$$x_s \;\leqslant\; c_1(s, \alpha) - k \cdot c_2(s, \alpha) + \sum_{s' \in S} \mathbf{P}(s, \alpha, s') \cdot x_{s'}.$$

Reducing LRA objectives in unichain IMCs to long-run ratio objectives in MDPs. We consider the transformation of an IMC into an MDP with 2 cost functions.

Definition 6. *Let $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ be an IMC and $G \subseteq S$ a set of goal states. The induced MDP is $\mathcal{M}(\mathcal{I}) = (S, Act \cup \{\perp\}, \mathbf{P}, s_0)$ with cost functions c_1 and c_2, where*

$$\mathbf{P}(s, \sigma, s') = \begin{cases} \frac{\mathbf{R}(s,s')}{E(s)} & \text{if } s \in MS \wedge \sigma = \perp \\ 1 & \text{if } s \in IS \wedge s \xrightarrow{\sigma} s' \\ 0 & \text{otherwise,} \end{cases}$$

$$c_1(s, \sigma) = \begin{cases} \frac{1}{E(s)} & \text{if } s \in MS \cap G \wedge \sigma = \perp \\ 0 & \text{otherwise,} \end{cases} \qquad c_2(s, \sigma) = \begin{cases} \frac{1}{E(s)} & \text{if } s \in MS \wedge \sigma = \perp \\ 0 & \text{otherwise.} \end{cases}$$

Observe that cost function c_2 keeps track of the average residence time in state s whereas c_1 only does so for states in G. The following result shows that the long-run average fraction of time spent in G-states in the IMC $\mathcal{I}$ and the long-run ratio objective $R^{\min}$ in the induced MDP $\mathcal{M}(\mathcal{I})$ coincide.

Theorem 4. *For unichain IMC $\mathcal{I}$, $LRA^{\min}(s, G)$ equals $R^{\min}(s)$ in MDP $\mathcal{M}(\mathcal{I})$.*

[4] In our setting, $\mathcal{R}(\pi)$ is well-defined as the cost functions c_1 and c_2 are obtained from non-Zeno IMCs, as explained below. This entails that for any infinite path π, $c_2(s_j, \alpha_j) > 0$ for some index j.

Proof. Let $\mathcal{I}$ be a unichain IMC with state space S and $G \subseteq S$. Consider a stationary deterministic scheduler D on $\mathcal{I}$. As $\mathcal{I}$ is unichain, D induces an ergodic CTMC $(S, \mathbf{R}, s_0)$, where $\mathbf{R}(s, s') = \sum\{\lambda \mid s \stackrel{\lambda}{\Longrightarrow} s'\}$, and $\mathbf{R}(s, s') = \infty$ if $s \in IS$ and $s \xrightarrow{D(s)} s'$.[5] The proof now proceeds in three steps.

$\langle 1 \rangle$ According to the ergodic theorem for CTMCs [24], almost surely:

$$\mathbb{E}_{s_i}\Big(\lim_{t\to\infty} \frac{1}{t}\int_0^t \mathbf{I}_{\{s_i\}}(X_u)\,du\Big) \;=\; \frac{1}{z_i \cdot E(s_i)}.$$

Here, random variable X_t denotes the state of the CTMC at time t and $z_i = \mathbb{E}_i(T_i)$ is the expected return time to state s_i where random variable T_i is the return time to s_i when starting from s_i. We assume $\frac{1}{\infty} = 0$. Thus, in the long run almost all paths will stay in s_i for $\frac{1}{z_i \cdot E(s_i)}$ fraction of time.

$\langle 2 \rangle$ Let μ_i be the probability to stay in s_i in the long run in the embedded discrete-time Markov chain $(S, \mathbf{P}', s_0)$ of CTMC $(S, \mathbf{R}, s_0)$. Thus $\boldsymbol{\mu} \cdot \mathbf{P}' = \boldsymbol{\mu}$ where $\boldsymbol{\mu}$ is the vector containing μ_i for all states $s_i \in S$. Given the probability μ_i of staying in state s_i, the expected return time to s_i is

$$z_i = \frac{\sum_{s_j \in S} \mu_j \cdot E(s_j)^{-1}}{\mu_i}.$$

$\langle 3 \rangle$ Gathering the above results now yields:

$$\begin{aligned}
\textsc{Lra}^D(s, G) &= \mathbb{E}_{s,D}\Big(\lim_{t\to\infty} \frac{1}{t}\int_0^t \mathbf{I}_G(X_u)\,du\Big) = \mathbb{E}_{s,D}\Big(\lim_{t\to\infty} \frac{1}{t}\int_0^t \sum_{s_i \in G} \mathbf{I}_{\{s_i\}}(X_u)\,du\Big) \\
&= \sum_{s_i \in G} \mathbb{E}_{s,D}\Big(\lim_{t\to\infty} \frac{1}{t}\int_0^t \mathbf{I}_{\{s_i\}}(X_u)\,du\Big) \stackrel{\langle 1 \rangle}{=} \sum_{s_i \in G} \frac{1}{z_i \cdot E(s_i)} \\
&\stackrel{\langle 2 \rangle}{=} \sum_{s_i \in G} \frac{\mu_i}{\sum_{s_j \in S} \mu_j E(s_j)^{-1}} \cdot \frac{1}{E(s_i)} = \frac{\sum_{s_i \in G} \mu_i E(s_i)^{-1}}{\sum_{s_j \in S} \mu_j E(s_j)^{-1}} \\
&= \frac{\sum_{s_i \in S} \mathbf{I}_G(s_i) \cdot \mu_i E(s_i)^{-1}}{\sum_{s_j \in S} \mu_j E(s_j)^{-1}} = \frac{\sum_{s_i \in S} \mu_i \cdot (\mathbf{I}_G(s_i) \cdot E(s_i)^{-1})}{\sum_{s_j \in S} \mu_j \cdot E(s_j)^{-1}} \\
&\stackrel{(\star)}{=} \frac{\sum_{s_i \in S} \mu_i \cdot c_1(s_i, D(s_i))}{\sum_{s_j \in S} \mu_j \cdot c_2(s_j, D(s_j))} \stackrel{(\star\star)}{=} \mathbb{E}_{s,D}(\mathcal{R})
\end{aligned}$$

Step $(\star)$ is due to the definition of c_1, c_2. Step $(\star\star)$ has been proven in [13].

By definition, there is a one-to-one correspondence between the schedulers of $\mathcal{I}$ and its MDP $\mathcal{M}(\mathcal{I})$. Together with the above results, this yields that $\textsc{Lra}^{\min} = \inf_D \textsc{Lra}^D(s)$ in IMC $\mathcal{I}$ equals $R^{\min}(s) = \inf_D \mathbb{E}_{s,D}(\mathcal{R})$ in MDP $\mathcal{M}(\mathcal{I})$. $\square$

To summarize, computing the minimum long-run average fraction of time that is spent in some goal state in $G \subseteq S$ in unichain IMC $\mathcal{I}$ equals the minimum long-run ratio objective in an MDP with two cost functions. The latter can be obtained by solving an LP problem. Observe that for any two states s, s' in a unichain IMC, $\textsc{Lra}^{\min}(s, G)$ and $\textsc{Lra}^{\min}(s', G)$ coincide. In the sequel, we therefore omit the state and simply write $\textsc{Lra}^{\min}(G)$ when considering unichain IMCs. In the next subsection, we consider IMCs that are not unichains.

[5] Strictly speaking, ∞ is not characterizing a negative exponential distribution and is used here to model an instantaneous transition. The results applied to CTMCs in this proof are not affected by this slight extension of rates.

4.2 Reduction to a stochastic shortest path problem

Let $\mathcal{I}$ be an IMC with initial state s_0 and maximal end components $\{\mathcal{I}_1, \ldots, \mathcal{I}_k\}$ for $k > 0$ where IMC $\mathcal{I}_j$ has state space S_j. Note that being a maximal end component implies that each $\mathcal{I}_j$ is also a unichain IMC. Using this decomposition of $\mathcal{I}$ into maximal end components, we obtain the following result:

Lemma 1. *Let $\mathcal{I} = (S, Act, \rightarrow, \Longrightarrow, s_0)$ be an IMC, $G \subseteq S$ a set of goal states and $\{\mathcal{I}_1, \ldots, \mathcal{I}_k\}$ the set of maximal end components in $\mathcal{I}$ with state spaces $S_1, \ldots, S_k \subseteq S$. Then*

$$\textsc{Lra}^{\min}(s_0, G) = \inf_D \sum_{j=1}^{k} \textsc{Lra}_j^{\min}(G) \cdot \Pr^D(s_0 \models \Diamond S_j),$$

where $\Pr^D(s_0 \models \Diamond S_j)$ is the probability to eventually reach some state in S_j from s_0 under scheduler D and $\textsc{Lra}_j^{\min}(G)$ is the long-run average fraction of time spent in $G \cap S_j$ in unichain IMC $\mathcal{I}_j$.

We finally show that the problem of computing minimal LRA is reducible to a non-negative SSP problem [2,15]. This is done as follows. In IMC $\mathcal{I}$, each maximal end component $\mathcal{I}_j$ is replaced by a new state u_j. Formally, let $U = \{u_1, \ldots, u_k\}$ be a set of fresh states such that $U \cap S = \varnothing$.

Definition 7 (SSP for long run average). *Let $\mathcal{I}$, S, $G \subseteq S$, $\mathcal{I}_j$ and S_j be as before. The SSP induced by $\mathcal{I}$ for the long-run average fraction of time spent in G is the tuple $\mathcal{P}_{LRA^{\min}}(\mathcal{I}) = \left(S \setminus \bigcup_{i=1}^{k} S_i \cup U, Act \cup \{\perp\}, \mathbf{P}', s_0, U, c, g\right)$, where*

$$\mathbf{P}'(s, \sigma, s') = \begin{cases} \mathbf{P}(s, \sigma, s'), & \text{if } s, s' \in S \setminus \bigcup_{i=1}^{k} S_i \\ \sum_{s' \in S_j} \mathbf{P}(s, \sigma, s') & \text{if } s \in S \setminus \bigcup_{i=1}^{k} S_i \wedge s' = u_j, u_j \in U \\ 1 & \text{if } s = s' = u_i \in U \wedge \sigma = \perp \\ 0 & \text{otherwise.} \end{cases}$$

Here, $\mathbf{P}$ is defined as in Def. 6. Furthermore, $g(u_i) = \textsc{Lra}_i^{\min}(G)$ for $u_i \in U$ and $c(s, \sigma) = 0$ for all s and $\sigma \in Act \cup \{\perp\}$.

The state space of the SSP consists of all states in the IMC $\mathcal{I}$ where each maximal end component $\mathcal{I}_j$ is replaced by a single state u_j which is equipped with a $\perp$-labeled self-loop. The terminal costs of the new states u_i are set to $\textsc{Lra}_i^{\min}(G)$. The transition probabilities are defined as in the transformation of an IMC into an MDP, see Def. 6, except that for transitions to u_j the cumulative probability to move to one of the states in S_j is taken. Note that as interactive transitions are uniquely labeled (as we consider closed IMCs), $\mathbf{P}'$ is indeed a probability function. The following theorem states the correctness of the reduction.

Theorem 5 (Correctness of the reduction). *For IMC $\mathcal{I}$ and its induced SSP $\mathcal{P}_{LRA^{\min}}(\mathcal{I})$ it holds:*

$$\textsc{Lra}^{\min}(s, G) = cR^{\min}(s, \Diamond U)$$

where $cR^{\min}(s, \Diamond U)$ is the minimal cost reachability of U in SSP $\mathcal{P}_{LRA^{\min}}(\mathcal{I})$.

Example 3. Consider the IMC $\mathcal{I}$ in Fig. 1 and its maximal end components $\mathcal{I}_1$ and $\mathcal{I}_2$ with state spaces $S_1 = \{s_4, s_5, s_6, s_7\}$ and $S_2 = \{s_3, s_8, s_9, s_{10}\}$, respectively. Let $G = \{s_7, s_8\}$ be the set of goal states. For the underlying MDP $\mathcal{M}(\mathcal{I})$, we have $\mathbf{P}(s_4, \gamma_1, s_5) = 1$, $c_1(s_4, \gamma_1) = c_2(s_4, \gamma_1) = 0$, $\mathbf{P}(s_7, \perp, s_4) = \frac{1}{2}$, $c_1(s_7, \perp) = c_2(s_7, \perp) = \frac{1}{10}$, and $\mathbf{P}(s_5, \perp, s_7) = 1$ with $c_1(s_5, \perp) = 0$ and $c_2(s_5, \perp) = \frac{1}{20}$. Solving the linear programming problems for each of the maximal end components $\mathcal{I}_1$ and $\mathcal{I}_2$, we obtain $\text{LRA}_1^{\min}(G) = \frac{2}{3}$, $\text{LRA}_1^{\max}(G) = \frac{4}{5}$, and $\text{LRA}_2^{\max}(G) = \text{LRA}_2^{\min}(G) = \frac{9}{13}$. The SSP $\mathcal{P}_{LRA^{\min}}(\mathcal{I})$ for the complete IMC $\mathcal{I}$ is obtained by replacing $\mathcal{I}_1$ and $\mathcal{I}_2$ with fresh states u_1 and u_2 where $g(u_1) = \frac{2}{3}$ and $g(u_2) = \frac{9}{13}$. We have $\mathbf{P}'(s_1, \perp, u_1) = \frac{1}{3}$, $\mathbf{P}'(s_2, \beta_2, u_2) = 1$, etc. Finally, by solving the linear programming problem for $\mathcal{P}_{LRA^{\min}}(\mathcal{I})$, we obtain $\text{LRA}^{\min}(s_0, G) = \frac{80}{117}$ by choosing α_1 in state s_0 and γ_1 in state s_4. Dually, $\text{LRA}^{\max}(s_0, G) = \frac{142}{195}$ is obtained by choosing α_1 in state s_0 and γ_2 in state s_4.

5 Case studies

5.1 Tool support

What is IMCA*?* IMCA (Interactive Markov Chain Analyzer) is a tool for the *quantitative* analysis of IMCs. In particular, it supports the verification of IMCs against (a) timed reachability objectives, (b) reachability objectives, (c) expected time objectives, (d) expected step objectives, and (e) long-run average objectives. In addition, it supports the minimization of IMCs with respect to strong bisimulation. IMCA synthesizes ε-optimal piecewise constant timed policies for (a) timed reachability objectives using the approach of [30], and optimal positional policies for the objectives (b)–(e). Measures (c) and (e) are determined using the approach explained in this paper. IMCA supports the plotting of piecewise constant policies (on a per state basis) and incorporates a plot functionality for timed reachability which allows to plot the timed reachability probabilities for a state over a given time interval.

Input format. IMCA has a simple input format that facilitates its usage as a back-end tool for other tools that generate IMCs from high-level model specifications such as AADL, DFTs, PRISM reactive modules, and so on. It supports the `bcg`-format, such that it accepts state spaces generated (and possibly minimized) using the CADP toolbox [11]; CADP supports a LOTOS-variant for the compositional modeling of IMCs and compositional minimization of IMCs.

Implementation Details. A schematic overview of the IMCA tool is given in Fig. 2. The tool is written in `C++`, consists of about 6,000 lines of code, and

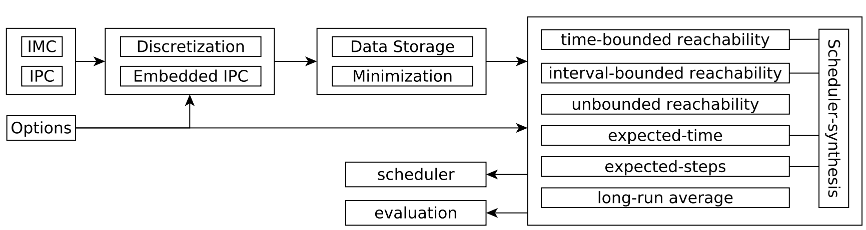

Fig. 2. Tool functionality of IMCA.

N	# states	# transitions	$\lvert G\rvert$	$eT^{\max}(s, \Diamond G)$ time (s)	$\Pr^{\max}(s, \Diamond G)$ time (s)	$\mathrm{LRA}^{\max}(s, G)$ time (s)
1	111	320	74	0.0009	0.0061	0.0046
4	819	2996	347	0.0547	0.0305	0.1137
8	2771	10708	1019	0.6803	0.3911	1.3341
16	10131	40340	3419	10.1439	5.3423	20.0278
32	38675	156436	12443	292.7389	94.0289	455.4387
52	100275	408116	31643	3187.1171	1807.7994	OOM

Table 1. Computation times for the workstation cluster.

exploits the GNU Multiple Precision Arithmetic Library[6] and the Multiple Precision Floating-Point Reliable Library[7] so as to deal with the small probabilities that occur during discretization for (a). Other included libraries are QT 4.6 and SoPlex 1.6.0 [28].The latter supports several efficient algorithms to solve LP problems; by default it uses simplex on an LP problem and its dual.

5.2 Case studies

We study the practical feasibility of IMCA's algorithms for expected time reachability and long-run averages on two case studies: A dependable workstation cluster [18] and a Google file system [10]. The experiments were conducted on a single core of a 2.8 GHz Intel Core i7 processor with 4GB RAM running Linux.

Workstation cluster. In this benchmark, two clusters of workstations are connected via a backbone network. In each cluster, the workstations are connected via a switch. All components can fail. Our model for the workstation cluster benchmark is basically as used in all of its studies so far, except that the inspection transitions in the GSPN (Generalized Stochastic Petri Net) model of [18] are immediate rather than —as in all current studies so far— stochastic transitions with a very high rate. Accordingly, whenever the repair unit is available and different components have failed, the choice which component to repair next is nondeterministic (rather than probabilistic). This yields an IMC with the same size as the Markov chain of [18]. Table 1 shows the computation times for the maximum expected reachability times where the set G of goal states depends on the number N of operational workstations. More precisely, G is the set of states in which none of the operational left (or right) workstations connected via an operational switch and backbone is available. For the sake of comparison, the next column indicates the computation times for unbounded reachability probabilities for the same goal set. The last column of Table 1 lists the results for the long-run average analysis; the model consists of a single end component.

Google file system. The model of [10] focuses on a replicated file system as used as part of the Google search engine. In the Google file system model, files are divided into chunks of equal size. Several copies of each chunk reside at several chunk servers. The location of the chunk copies is administered by a single master server. If a user of the file system wants to access a certain chunk of a file, it asks the master for the location. Data transfer then takes place directly between a chunk server and the user. The model features three parameters: The number

[6] `http://gmplib.org/`.

[7] `http://www.mpfr.org/`.

M	# states	# transitions	$\lvert G\rvert$	$eT^{\min}(s, \diamond G)$ time (s)	$\Pr^{\min}(s, \diamond G)$ time (s)	$\text{LRA}^{\min}(s, G)$ time (s)
10	1796	6544	408	1.6568	0.1584	0.1411
20	7176	27586	1713	2.6724	2.5669	14.9804
30	16156	63356	3918	11.3836	14.2459	35.0654
40	28736	113928	7023	31.1416	48.8603	236.5308
60	64696	202106	15933	142.2179	315.8246	OOM

Table 2. Computation times for Google file system ($S = 5000$ and $N = 100000$).

M of chunk servers, the number S of chunks a chunk server may store, and the total number N of chunks. In our setting, $S = 5000$ and $N = 100000$, whereas M varies. The set G of goal states characterizes the set of states that offer at least service level one. We consider a variant of the GSPN model in [10] in which the probability of a hardware or a software failure in the chunk server is unknown. This aspect was not addressed in [10]. Table 2 summarizes the computation times for the analysis of the nondeterministic Google file system model.

6 Conclusions

We presented novel algorithms, prototypical tool support in Imca, and two case studies for the analysis of expected time and long run average objectives of IMCs. We have shown that both objectives can be reduced to stochastic shortest path problems. As IMCs are the semantic backbone of engineering formalisms such as AADL error models [5], dynamic fault trees [4] and GALS hardware designs [12], our contribution enlarges the analysis capabilities for dependability and reliability. The support of the compressed `bcg`-format allows for the direct usage of our tool and algorithms as back-end to tools like CADP [11] and CORAL [4]. The tool and case studies are publicly available at `http://moves.rwth-aachen.de/imca`. Future work will focus on the generalization of the presented algorithms to Markov automata [16], and experimentation with symbolic data structures such as multi-terminal BDDs by, e.g. exploiting Prism for the MDP analysis.

Acknowledgment. This research was supported by the EU FP7 MoVeS and MEALS projects, the ERC advanced grant VERIWARE, the DFG research center AVACS (SFB/TR 14) and the DFG/NWO ROCKS programme. We thank Silvio de Carolis for the `bcg`-interface and Ernst Moritz Hahn for his help on the Google file system.

References

1. Baier, C., Haverkort, B. R., Hermanns, H., Katoen, J.-P.: Model-checking algorithms for continuous-time Markov chains. *IEEE TSE* **29** (2003) 524–541
2. Bertsekas, D. P., Tsitsiklis, J. N.: An analysis of stochastic shortest path problems. *Mathematics of Operations Research* **16** (1991) 580–595
3. Böde, E., Herbstritt, M., Hermanns, H., Johr, S., Peikenkamp, T., Pulungan, R., Rakow, J., Wimmer, R., Becker, B.: Compositional dependability evaluation for STATEMATE. *IEEE TSE* **35** (2009) 274–292
4. Boudali, H., Crouzen, P., Stoelinga, M.: A rigorous, compositional, and extensible framework for dynamic fault tree analysis. *IEEE TSDC* **7** (2009) 128–143
5. Bozzano, M., Cimatti, A., Katoen, J.-P., Nguyen, V., Noll, T., Roveri, M.: Safety, dependability and performance analysis of extended AADL models. *The Computer Journal* **54** (2011) 754–775

6. Brázdil, T., Forejt, V., Krcál, J., Kretínský, J., Kucera, A.: Continuous-time stochastic games with time-bounded reachability. In: *FSTTCS. LIPIcs*, Vol. 4. Schloss Dagstuhl (2009) 61–72
7. Buchholz, P., Hahn, E. M., Hermanns, H., Zhang, L.: Model checking algorithms for CT-MDPs. In: *CAV. LNCS*, Vol. 6806. Springer (2011) 225–242
8. Buchholz, P., Schulz, I.: Numerical analysis of continuous time Markov decision processes over finite horizons. *Computers & OR* **38** (2011) 651–659
9. Chatterjee, K., Henzinger, M.: Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In: *Symp. on Discrete Algorithms (SODA)*. SIAM (2011) 1318–1336
10. Cloth, L., Haverkort, B. R.: Model checking for survivability. In: *QEST*. IEEE Computer Society (2005) 145–154
11. Coste, N., Garavel, H., Hermanns, H., Lang, F., Mateescu, R., Serwe, W.: Ten years of performance evaluation for concurrent systems using CADP. In: *ISoLA. LNCS*, Vol. 6416. Springer (2010) 128–142
12. Coste, N., Hermanns, H., Lantreibecq, E., Serwe, W.: Towards performance prediction of compositional models in industrial GALS designs. In: *CAV. LNCS*, Vol. 5643. Springer (2009) 204–218
13. de Alfaro, L.: *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University (1997)
14. de Alfaro, L.: How to specify and verify the long-run average behavior of probabilistic systems. In: *LICS*. IEEE CS Press (1998) 454–465
15. de Alfaro, L.: Computing minimum and maximum reachability times in probabilistic systems. In: *CONCUR. LNCS*, Vol. 1664. Springer (1999) 66–81
16. Eisentraut, C., Hermanns, H., Zhang, L.: On probabilistic automata in continuous time. In: *LICS*. IEEE Computer Society (2010) 342–351
17. Guck, D., Han, T., Katoen, J.-P., Neuhäußer, M. R.: Quantitative timed analysis of interactive Markov chains. In: *NASA Formal Methods*. LNCS 7226. Springer (2012) 8–23
18. Haverkort, B. R., Hermanns, H., Katoen, J.-P.: On the use of model checking techniques for dependability evaluation. In: *SRDS*. IEEE CS (2000) 228–237
19. Hermanns, H.: *Interactive Markov Chains and the Quest for Quantified Quality*. *LNCS*, Vol. 2428. Springer (2002)
20. Hermanns, H., Katoen, J.-P.: The how and why of interactive Markov chains. In: *FMCO. LNCS*, Vol. 6286. Springer (2009) 311–337
21. Johr, S.: *Model Checking Compositional Markov Systems*. PhD thesis, Saarland University (2007)
22. Knast, R.: Continuous-time probabilistic automata. *Information and Control* **15** (1969) 335–352
23. López, G., Hermanns, H., Katoen, J.-P.: Beyond memoryless distributions: Model checking semi-Markov chains. In: *PAPM-PROBMIV*. LNCS 2165. Springer (2001) 57–70
24. Norris, J.: *Markov Chains*. Cambridge University Press (1997)
25. Puterman, M. L.: *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons (1994)
26. Rabe, M. N., Schewe, S.: Finite optimal control for time-bounded reachability in CTMDPs and continuous-time Markov games. *Acta Inf.* **48** (2011) 291–315
27. von Essen, C., Jobstmann, B.: Synthesizing systems with optimal average-case behavior for ratio objectives. In: *iWIGP. EPTCS*, Vol. 50. (2011) 17–32
28. Wunderling, R.: *Paralleler und objektorientierter Simplex-Algorithmus*. PhD thesis, Technische Universität Berlin (1996) http://www.zib.de/Publications/abstracts/TR-96-09/.
29. Yushtein, Y., Bozzano, M., Cimatti, A., Katoen, J.-P., Nguyen, V. Y., Noll, T., Olive, X., Roveri, M.: System-software co-engineering: Dependability and safety perspective. In: *SMC-IT*. IEEE Computer Society (2011) 18–25
30. Zhang, L., Neuhäußer, M. R.: Model checking interactive Markov chains. In: *TACAS. LNCS*, Vol. 6015. Springer (2010) 53–68

A Proof of Theorem 1

Theorem 1. The function $eT^{\min}$ is a fixpoint of the Bellman operator

$$[L(v)](s) = \begin{cases} \dfrac{1}{E(s)} + \displaystyle\sum_{s' \in S} \mathbf{P}(s,s') \cdot v(s') & \text{if } s \in \mathit{MS} \setminus G \\ \displaystyle\min_{s \xrightarrow{\alpha} s'} v(s') & \text{if } s \in \mathit{IS} \setminus G \\ 0 & \text{if } s \in G. \end{cases}$$

Proof. We show $L\left(eT^{\min}(s, \Diamond G)\right) = eT^{\min}(s, \Diamond G)$ for all $s \in S$. Distinguish three cases: $s \in \mathit{MS} \setminus G$, $s \in \mathit{IS} \setminus G$ and $s \in G$. If $s \in \mathit{MS} \setminus G$, we derive

$$\begin{aligned} eT^{\min}(s, \Diamond G) &= \inf_D \mathbb{E}_{s,D}(V_G) = \inf_D \int_{\mathit{Paths}} V_G(\pi) \Pr_{s,D}(d\pi) \\ &= \inf_D \int_0^\infty t \cdot E(s) e^{-E(s)t} + \sum_{s' \in S} \mathbf{P}(s,s') \cdot \mathbb{E}_{s',D\left(s \xrightarrow{t,\perp} \cdot\right)}(V_G)\, dt \\ &= \int_0^\infty t \cdot E(s) e^{-E(s)t} + \sum_{s' \in S} \mathbf{P}(s,s') \cdot \inf_D \mathbb{E}_{s',D\left(s \xrightarrow{t,\perp} \cdot\right)}(V_G)\, dt \\ &= \int_0^\infty t \cdot E(s) e^{-E(s)t} + \sum_{s' \in S} \mathbf{P}(s,s') \cdot \inf_D \mathbb{E}_{s',D}(V_G)\, dt \\ &= \int_0^\infty t \cdot E(s) e^{-E(s)t}\, dt + \sum_{s' \in S} \mathbf{P}(s,s') \cdot eT^{\min}(s', \Diamond G) \\ &= \frac{1}{E(s)} + \sum_{s' \in S} \mathbf{P}(s,s') \cdot eT^{\min}(s', \Diamond G) = L\left(eT^{\min}(s, \Diamond G)\right). \end{aligned}$$

If $s \in \mathit{IS} \setminus G$, we derive

$$\begin{aligned} eT^{\min}(s, \Diamond G) &= \inf_D \mathbb{E}_{s,D}(V_G) = \inf_D \int_{\mathit{Paths}} V_G(\pi) \Pr_{s,D}(d\pi) \\ &= \inf_D \sum_{s \xrightarrow{\alpha} s'} D(s)(\alpha) \cdot \mathbb{E}_{s',D\left(s \xrightarrow{\alpha,0} \cdot\right)}(V_G). \end{aligned}$$

As IMC $\mathcal{I}$ is closed, each transition can be assumed to be uniquely labeled, i.e. each action α uniquely determines a successor state s' with $s \xrightarrow{\alpha} s'$. Let

$$\alpha = \arg\min_{s \xrightarrow{\alpha} s'} \inf_D \mathbb{E}_{s',D}(V_G).$$

Then all optimal schedulers must choose α with probability 1, i.e. $D(s)(\alpha) = 1$ and $D(s)(\sigma) = 0$ for all $\sigma \neq \alpha$. Hence, we obtain

$$\begin{aligned} eT^{\min}(s, \Diamond G) &= \inf_D \min_{s \xrightarrow{\alpha} s'} \mathbb{E}_{s',D\left(s \xrightarrow{\alpha,0} \cdot\right)}(V_G) = \min_{s \xrightarrow{\alpha} s'} \inf_D \mathbb{E}_{s',D\left(s \xrightarrow{\alpha,0} \cdot\right)}(V_G) \\ &= \min_{s \xrightarrow{\alpha} s'} \inf_D \mathbb{E}_{s',D}(V_G) = \min_{s \xrightarrow{\alpha} s'} eT^{\min}(s', \Diamond G) = L\left(eT^{\min}(s, \Diamond G)\right). \end{aligned}$$

Finally, assume $s \in G$. Then

$$eT^{\min}(s, \Diamond G) = \inf_D \int_{\mathit{Paths}} V_G(\pi) \Pr_{s,D}(d\pi) = 0 = L\left(eT^{\min}(s, \Diamond G)\right). \qquad \square$$

B Proof of Theorem 3

Theorem 3. There is a stationary deterministic scheduler yielding $\textsc{Lra}^{\min}(s, G)$.

Proof (Sketch). Transform IMC $\mathcal{I}$ into a continuous-time MDP (CTMDP) $\mathcal{C}_{\mathcal{I}}$ as in [21]. There is a one-to-one correspondence between schedulers on IMC $\mathcal{I}$ and schedulers of CTMDP $\mathcal{C}_{\mathcal{I}}$ such that the probability measure of corresponding sets of infinite paths in $\mathcal{I}$ and $\mathcal{C}_{\mathcal{I}}$ coincide [21]. Thus, under a given scheduler, the LRA in IMC $\mathcal{I}$ equals the LRA in CTMDP $\mathcal{C}_{\mathcal{I}}$. It follows that $\textsc{Lra}^{\min}(s, G)$ in $\mathcal{I}$ equals $\textsc{Lra}^{\min}(s, G)$ in $\mathcal{C}_{\mathcal{I}}$. The theorem now follows from the fact that $\textsc{Lra}^{\min}(s, G)$ in CTMDPs are attained under stationary deterministic schedulers, cf. [25]. $\square$

C Proof of Lemma 1

Lemma 1. $\textsc{Lra}^{\min}(s_0, G) \;=\; \inf_D \sum_{j=1}^{k} \textsc{Lra}_j^{\min}(G) \cdot \Pr^D(s_0 \models \Diamond S_j)$ where $\Pr^D(s_0 \models \Diamond S_j)$ is the probability to eventually reach some state in S_j from s_0 under scheduler D and $\textsc{Lra}_j^{\min}(G)$ is the long-run average fraction of time spent in G in unichain IMC $\mathcal{I}_j$.

Proof (Sketch). Under all deterministic stationary schedulers, each infinite path π in (finite) IMC $\mathcal{I}$ can be partitioned into two fragments: $\pi_{s_0 s} = s_0 s_1 \ldots s$ and $\pi_s^\omega = s \ldots s \ldots$, where each state on $\pi_{s_0 s}$ (except s) do not belong to any $\mathcal{I}_j$ and all states on π_s^ω belong to $\mathcal{I}_i$, say. The minimal LRA will be obtained when the LRA in each IMC $\mathcal{I}_j$ is minimal and the reachability probability to each IMC $\mathcal{I}_j$ is minimal. A scheduler that attains $\textsc{Lra}^{\min}(s_0, G)$ thus acts according to a scheduler that minimizes the probability to reach $\mathcal{I}_j$ and then acts as a scheduler that minimizes the LRA within $\mathcal{I}_j$. $\square$

D Proof of Theorem 5

Theorem 5 (Correctness of the reduction). For IMC $\mathcal{I}$ and its induced SSP $\mathcal{P}_{LRA^{\min}}(\mathcal{I})$ it holds:

$$\textsc{Lra}^{\min}(s, G) = cR^{\min}(s, \Diamond U)$$

where $cR^{\min}(s, \Diamond U)$ denotes the minimal cost reachability of U in SSP $\mathcal{P}_{LRA^{\min}}(\mathcal{I})$.

Proof. This follows straightforwardly from:

$$\begin{aligned} cR^{\min}(s, \Diamond U) &= \inf_D \mathbb{E}_{s,D}\big\{g(X_{T_U})\big\} = \inf_D \sum_{i=1}^{k} g(X_{T_{u_i}}) \cdot \Pr^D(s \models \Diamond u_i) \\ &= \inf_D \sum_{i=1}^{k} \textsc{Lra}_i^{\min}(G) \cdot \Pr^D(s \models \Diamond u_i) = \textsc{Lra}^{\min}(s, G). \qquad \square \end{aligned}$$

Aachener Informatik-Berichte

This list contains all technical reports published during the past three years. A complete list of reports dating back to 1987 is available from:

http://aib.informatik.rwth-aachen.de/

To obtain copies please consult the above URL or send your request to:

Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen,
Email: `biblio@informatik.rwth-aachen.de`

2009-01 * Fachgruppe Informatik: Jahresbericht 2009
2009-02 Taolue Chen, Tingting Han, Joost-Pieter Katoen, Alexandru Mereacre: Quantitative Model Checking of Continuous-Time Markov Chains Against Timed Automata Specifications
2009-03 Alexander Nyßen: Model-Based Construction of Embedded Real-Time Software - A Methodology for Small Devices
2009-05 George B. Mertzios, Ignasi Sau, Shmuel Zaks: A New Intersection Model and Improved Algorithms for Tolerance Graphs
2009-06 George B. Mertzios, Ignasi Sau, Shmuel Zaks: The Recognition of Tolerance and Bounded Tolerance Graphs is NP-complete
2009-07 Joachim Kneis, Alexander Langer, Peter Rossmanith: Derandomizing Non-uniform Color-Coding I
2009-08 Joachim Kneis, Alexander Langer: Satellites and Mirrors for Solving Independent Set on Sparse Graphs
2009-09 Michael Nett: Implementation of an Automated Proof for an Algorithm Solving the Maximum Independent Set Problem
2009-10 Felix Reidl, Fernando Sánchez Villaamil: Automatic Verification of the Correctness of the Upper Bound of a Maximum Independent Set Algorithm
2009-11 Kyriaki Ioannidou, George B. Mertzios, Stavros D. Nikolopoulos: The Longest Path Problem is Polynomial on Interval Graphs
2009-12 Martin Neuhäußer, Lijun Zhang: Time-Bounded Reachability in Continuous-Time Markov Decision Processes
2009-13 Martin Zimmermann: Time-optimal Winning Strategies for Poset Games
2009-14 Ralf Huuck, Gerwin Klein, Bastian Schlich (eds.): Doctoral Symposium on Systems Software Verification (DS SSV'09)
2009-15 Joost-Pieter Katoen, Daniel Klink, Martin Neuhäußer: Compositional Abstraction for Stochastic Systems
2009-16 George B. Mertzios, Derek G. Corneil: Vertex Splitting and the Recognition of Trapezoid Graphs
2009-17 Carsten Kern: Learning Communicating and Nondeterministic Automata
2009-18 Paul Hänsch, Michaela Slaats, Wolfgang Thomas: Parametrized Regular Infinite Games and Higher-Order Pushdown Strategies
2010-01 * Fachgruppe Informatik: Jahresbericht 2010
2010-02 Daniel Neider, Christof Löding: Learning Visibly One-Counter Automata in Polynomial Time

2010-03 Holger Krahn: MontiCore: Agile Entwicklung von domänenspezifischen Sprachen im Software-Engineering

2010-04 René Wörzberger: Management dynamischer Geschäftsprozesse auf Basis statischer Prozessmanagementsysteme

2010-05 Daniel Retkowitz: Softwareunterstützung für adaptive eHome-Systeme

2010-06 Taolue Chen, Tingting Han, Joost-Pieter Katoen, Alexandru Mereacre: Computing maximum reachability probabilities in Markovian timed automata

2010-07 George B. Mertzios: A New Intersection Model for Multitolerance Graphs, Hierarchy, and Efficient Algorithms

2010-08 Carsten Otto, Marc Brockschmidt, Christian von Essen, Jürgen Giesl: Automated Termination Analysis of Java Bytecode by Term Rewriting

2010-09 George B. Mertzios, Shmuel Zaks: The Structure of the Intersection of Tolerance and Cocomparability Graphs

2010-10 Peter Schneider-Kamp, Jürgen Giesl, Thomas Ströder, Alexander Serebrenik, René Thiemann: Automated Termination Analysis for Logic Programs with Cut

2010-11 Martin Zimmermann: Parametric LTL Games

2010-12 Thomas Ströder, Peter Schneider-Kamp, Jürgen Giesl: Dependency Triples for Improving Termination Analysis of Logic Programs with Cut

2010-13 Ashraf Armoush: Design Patterns for Safety-Critical Embedded Systems

2010-14 Michael Codish, Carsten Fuhs, Jürgen Giesl, Peter Schneider-Kamp: Lazy Abstraction for Size-Change Termination

2010-15 Marc Brockschmidt, Carsten Otto, Christian von Essen, Jürgen Giesl: Termination Graphs for Java Bytecode

2010-16 Christian Berger: Automating Acceptance Tests for Sensor- and Actuator-based Systems on the Example of Autonomous Vehicles

2010-17 Hans Grönniger: Systemmodell-basierte Definition objektbasierter Modellierungssprachen mit semantischen Variationspunkten

2010-18 Ibrahim Armaç: Personalisierte eHomes: Mobilität, Privatsphäre und Sicherheit

2010-19 Felix Reidl: Experimental Evaluation of an Independent Set Algorithm

2010-20 Wladimir Fridman, Christof Löding, Martin Zimmermann: Degrees of Lookahead in Context-free Infinite Games

2011-01 * Fachgruppe Informatik: Jahresbericht 2011

2011-02 Marc Brockschmidt, Carsten Otto, Jürgen Giesl: Modular Termination Proofs of Recursive Java Bytecode Programs by Term Rewriting

2011-03 Lars Noschinski, Fabian Emmes, Jürgen Giesl: A Dependency Pair Framework for Innermost Complexity Analysis of Term Rewrite Systems

2011-04 Christina Jansen, Jonathan Heinen, Joost-Pieter Katoen, Thomas Noll: A Local Greibach Normal Form for Hyperedge Replacement Grammars

2011-06 Johannes Lotz, Klaus Leppkes, and Uwe Naumann: dco/c Derivative Code by Overloading in C++

2011-07 Shahar Maoz, Jan Oliver Ringert, Bernhard Rumpe: An Operational Semantics for Activity Diagrams using SMV

2011-08 Thomas Ströder, Fabian Emmes, Peter Schneider-Kamp, Jürgen Giesl, Carsten Fuhs: A Linear Operational Semantics for Termination and Complexity Analysis of ISO Prolog

2011-09 Markus Beckers, Johannes Lotz, Viktor Mosenkis, Uwe Naumann (Editors): Fifth SIAM Workshop on Combinatorial Scientific Computing

2011-10 Markus Beckers, Viktor Mosenkis, Michael Maier, Uwe Naumann: Adjoint Subgradient Calculation for McCormick Relaxations

2011-11 Nils Jansen, Erika Ábrahám, Jens Katelaan, Ralf Wimmer, Joost-Pieter Katoen, Bernd Becker: Hierarchical Counterexamples for Discrete-Time Markov Chains

2011-12 Ingo Felscher, Wolfgang Thomas: On Compositional Failure Detection in Structured Transition Systems

2011-13 Michael Förster, Uwe Naumann, Jean Utke: Toward Adjoint OpenMP

2011-14 Daniel Neider, Roman Rabinovich, Martin Zimmermann: Solving Muller Games via Safety Games

2011-16 Niloofar Safiran, Uwe Naumann: Toward Adjoint OpenFOAM

2011-18 Kamal Barakat: Introducing Timers to pi-Calculus

2011-19 Marc Brockschmidt, Thomas Ströder, Carsten Otto, Jürgen Giesl: Automated Detection of Non-Termination and NullPointerExceptions for Java Bytecode

2011-24 Callum Corbett, Uwe Naumann, Alexander Mitsos: Demonstration of a Branch-and-Bound Algorithm for Global Optimization using McCormick Relaxations

2011-25 Callum Corbett, Michael Maier, Markus Beckers, Uwe Naumann, Amin Ghobeity, Alexander Mitsos: Compiler-Generated Subgradient Code for McCormick Relaxations

2011-26 Hongfei Fu: The Complexity of Deciding a Behavioural Pseudometric on Probabilistic Automata

2012-01 * Fachgruppe Informatik: Annual Report 2012

2012-02 Thomas Heer: Controlling Development Processes

2012-03 Arne Haber, Jan Oliver Ringert, Bernhard Rumpe: MontiArc - Architectural Modeling of Interactive Distributed and Cyber-Physical Systems

2012-04 Marcus Gelderie: Strategy Machines and their Complexity

2012-05 Thomas Ströder, Fabian Emmes, Jürgen Giesl, Peter Schneider-Kamp, and Carsten Fuhs: Automated Complexity Analysis for Prolog by Term Rewriting

2012-06 Marc Brockschmidt, Richard Musiol, Carsten Otto, Jürgen Giesl: Automated Termination Proofs for Java Programs with Cyclic Data

2012-08 Hongfei Fu: Computing Game Metrics on Markov Decision Processes

* These reports are only available as a printed version.

Please contact `biblio@informatik.rwth-aachen.de` to obtain copies.