Improving Time Bounded Reachability Computations in Interactive Markov Chains

Hassan Hatefi, Holger Hermanns

Improving Time Bounded Reachability Computations in Interactive Markov Chains*

Hassan Hatefi[1,2] and Holger Hermanns[1]

[1] Saarland University – Computer Science, Saarbrücken, Germany
[2] Max-Planck-Institut für Informatik, Saarbrücken, Germany
hhatefi@depend.cs.uni-saarland.de, hermanns@cs.uni-saarland.de

Abstract. Interactive Markov Chains (IMCs) are compositional behaviour models extending both Continuous Time Markov Chain (CTMC) and Labeled Transition System (LTS). They are used as semantic models in different engineering contexts ranging from ultramodern satellite designs to industrial system-on-chip manufacturing. Different approximation algorithms have been proposed for model checking of IMC, with time bounded reachability probabilities playing a pivotal role. This paper addresses the accuracy and efficiency of approximating time bounded reachability probabilities in IMC, improving over the state-of-the-art in both efficiency of computation and tightness of approximation. Experimental evidence is provided by applying the new method on a case study.

1 Introduction

Why IMCs? Over the last decade, a formal approach to quantitative performance and dependability evaluation of concurrent systems has gained maturity. At its root are continuous-time Markov chains for which efficient and quantifiably precise solution methods exist [3]. On the specification side, continuous stochastic logic (CSL) [1, 3] enables the specification of a large spectrum of performance and dependability measures. A CTMC can be viewed as a labelled transition system (LTS) whose transitions are delayed according to exponential distributions. Opposed to classical concurrency theory models, CTMCs neither support compositional modelling [23] nor do they allow nondeterminism in the model. Among several formalisms that overcome these limitations [7, 21, 24, 25], interactive Markov chains (IMCs) [22] stand out. IMCs conservatively extend classical concurrency theory with exponentially distributed delays, and this induces several further benefits [8]. In particular, it enables compositional modelling with intermittent weak bisimulation minimisation [21] and allows to augment existing untimed process algebra specifications with random timing [7]. Moreover, the IMC formalism is not restricted to exponential delays but allows to encode arbitrary phase-type distributions such as hyper- and hypoexponentials [28]. Since IMCs smoothly extend classical LTSs, the model has received attention in academic as well as in industrial settings [6, 13, 12, 16].

* This work has been supported by the DFG as part of SFB/TR 14 AVACS, by the DFG/NWO bilateral project ROCKS, and by the European Union FP7-ICT projects MEALS, grant agreement no. 295261, and SENSATION, grant agreement no. 318490.

Why time bounded reachability? The principles of model checking IMCs are by now well understood. One analysis strand, implemented for instance in CADP [17], resorts to CSL model checking of CTMCs. But this is only applicable if the weak bisimulation quotient of the model is indeed a CTMC, which cannot be always guaranteed. This is therefore a partial solution technique, albeit it integrates well with compositional construction and minimisation approaches, and is the one used in industrial applications. The approximate CSL model checking problem for IMCs has been solved by Neuhäusser and Zhang [26, 29]. Most of the solution resorts to untimed model checking [5]. The core innovation lies in the solution of the time bounded model checking problem, that is needed to quantify a *bounded until formula* subject to a (real-valued) time interval. The problem is solved by splitting the time interval into equally sized digitisation steps, each small enough such that with high probability at most one Markov transition occurs in any step.

However, the practical efficiency and accuracy of this approach to evaluate time bounded reachability probabilities turns out substantially inferior to the one known for CTMCs, and this limits applicability to real industrial cases. As a consequence, model checking algorithms for other, less precise, but still highly relevant properties have been coined [19], including expected reachability and long run average properties.

Our contribution. We revisit the approximation of time bounded reachability probabilities so as to arrive at an improved computational approach. For this, we generalise the digitisation approach of Neuhäusser and Zhang [26, 29] by considering the effect of multiple Markov transition firings in a time interval of length δ. We show that this can be exploited by a tighter error bound, and thus a more accurate computation. We put the theoretical improvement into practice by proposing a new algorithm to solve time bounded reachability in IMCs. Empirical results demonstrate that the improved algorithm can gain more than one order of magnitude speedups.

2 Interactive Markov Chain

An Interactive Markov Chain (IMC) is a model that generalises both CTMC and LTS. In this section, we provide the definition of IMC and the necessary concepts relating to it.

Definition 1 (IMC). *An IMC [21] is a tuple* $\mathcal{M} = (S, Act, \longrightarrow, \dashrightarrow, s_0)$*, where*

- S *is a finite set of states,*
- Act *is a set of actions, including* τ*, representing the internal invisible action,*
- $\longrightarrow \subset S \times Act \times S$ *is a set of interactive transitions,*
- $\dashrightarrow \subset S \times \mathbb{R}_{\geq 0} \times S$ *is a set of Markov transitions,*
- s_0 *is the initial state.*

Maximum progress vs. urgency. States of an IMC are partitioned into *interactive*, *Markov* and *hybrid*. Interactive (Markov) states have only interactive (Markov) outgoing transitions, while hybrid states have transitions of both types. Let S_I, S_M and S_H be the set of interactive, Markov and hybrid states respectively. An IMC might have states without any outgoing transition. For the purpose of this paper, any such state is turned into a Markov state by adding a self loop with an arbitrary rate. We distinguish between closed and open IMCs. An open IMC can interact with the environment and in particular, can be composed with other IMCs, e.g. via parallel composition. For such models, a *maximum progress assumption* [21] is imposed which implies that τ-transitions take precedence over Markov transitions whenever both are enabled in a state. In contrast, a closed IMC is not subject to any further communication and composition. In this paper, we assume that the models we are going to analyse are closed, and impose the stronger *urgency assumption* which means that any interactive transition has precedence over Markov transitions, i.e. interactive transitions are taken immediately whenever enabled in a state, leaving no chance for enabled Markov transitions. Consequently, in a closed IMC, hybrid states can be regarded as interactive states.

Branching probabilities. A (probability) distribution μ over a discrete set S is a function $\mu : S \rightarrowtail [0,1]$ such that $\sum_{s \in S} \mu(s) = 1$. If $\mu(s) = 1$ for some $s \in S$, μ is a *Dirac* distribution denoted by μ_s. Let $Dist(S)$ be the set of all distributions over set S. For uniformity of notations, we use a distinguished action $\perp \notin Act$ to indicate Markov transitions and extend the set of actions to $Act_\perp = Act \cup \{\perp\}$. For $s \in S$, we define $Act_\perp(s)$ as the set of *enabled actions* in state s. If s is a Markov state, $Act_\perp(s) = \{\perp\}$, otherwise $Act_\perp(s) = \{\alpha \mid (s, \alpha, s') \in \longrightarrow\}$. The rate between state s and s' is defined by $rate(s, s') = \sum_{(s,\lambda,s') \in \dashrightarrow} \lambda$. Then $E(s) = \sum_{s' \in S} rate(s, s')$ denotes the sum of outgoing rates of state s. Using these concepts, we define the *branching probability* matrix for both interactive and Markov states by

$$\mathbf{P}(s, \alpha, s') = \begin{cases} 1 & s \in S_I \wedge (s, \alpha, s') \in \longrightarrow \\ \frac{rate(s,s')}{E(s)} & s \in S_M \wedge \alpha = \perp \\ 0 & \text{otherwise} \end{cases}$$

Example 1. Let $\mathcal{M}$ be the IMC in Figure 1. s_1 and s_3 are Markov states, while s_2 is an interactive state. Initial state s_0 is a hybrid state, since it has both interactive and Markov transitions enabled. Considering $\mathcal{M}$ as a closed IMC, the urgency assumption allows us to ignore $(s_0, 0.5, s_2) \in \dashrightarrow$ and consider s_0 as an interactive state. Under this assumption, interactive transitions are instantaneously fired after zero time delay. Conversely, the sojourn time in a Markov state s is exponentially distributed with rate $E(s)$. For example, the probability to leave s_1 within δ time unit is $1-e^{-5\delta}$ ($E(s_1) = 2+3 = 5$). At this point, branching probabilities determine the distribution of evolving to next states. For s_1, $\mathbf{P}(s_1, \perp, s_0) = \frac{2}{5}$

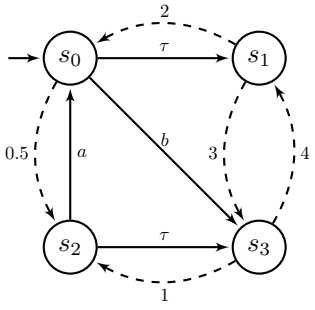

Fig. 1. An exemplary IMC

and $\mathbf{P}(s_1, \bot, s_3) = \frac{3}{5}$, as a result the probabilities to go to s_0 and s_3 after spending at most δ time unit in s_1 are $\frac{2}{5}(1 - e^{-5\delta})$ and $\frac{3}{5}(1 - e^{-5\delta})$ respectively.

Behavioural aspects. Like in other transition systems, an execution in an IMC is described by a path. Formally, a finite path is a finite sequence $\pi = s_0 \xmapsto{t_0,\alpha_0} s_1 \cdots s_{n-1} \xmapsto{t_{n-1},\alpha_{n-1}} s_n$ with $\alpha_i \in Act_\bot, t_i \in \mathbb{R}_{\geq 0}, i = 0 \cdots n-1$. We use $|\pi| = n$ as the length of π and $last(\pi) = s_n$ as the last state of π. Each step of a path π describes how the IMC evolves from one state to the next, with what action and after spending what state sojourn time. For example, when the IMC is in an interactive state $s \in S_I$, it must immediately (in zero time) choose some action $\alpha \in Act_\bot(s)$ and go to state s'. This gives rise to the finite path $s \xmapsto{0,\alpha} s'$. On the other hand, if $s \in S_M$, the IMC can stay for $t > 0$ time units and then choose the next state s' based on the distribution $\mathbf{P}(s, \bot, \cdot)$ by $s \xmapsto{t,\bot} s'$. An infinite path specifies an infinite execution of an IMC. We use $Paths^*$ and $Paths^\omega$ to denote the set of finite and infinite paths, respectively. By dropping the sojourn times from a path, we obtain the time-abstract path. We use subscript ta to refer to the set of time-abstract finite and infinite paths (i.e. $Paths^*_{ta}$ and $Paths^\omega_{ta}$).

Resolving nondeterminism. In states with more than one interactive transitions, the resolution of the transition to take is nondeterministic, just as in the LTS setting. This nondeterminism is resolved by schedulers. The most general scheduler class maps a finite and possibly timed path to a distribution over the set of interactive transitions enabled in the last state of the path:

Definition 2 (Generic Scheduler). *A generic scheduler over IMC $\mathcal{M} = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ is a function, $A : Paths^* \rightarrowtail Dist(\longrightarrow)$, where the support of $A(\pi)$ is a subset of $(\{last(\pi)\} \times Act \times S) \cap \longrightarrow$ and $last(\pi) \in S_I$.*

For a finite path π, a scheduler specifies how to resolve nondeterminism on the last state of π which is an interactive state. It gives a distribution over the set of enabled transitions of $last(\pi)$. We use the term Gen to refer to the set of all generic schedulers. Following the definition of schedules, the probability measure can be uniquely defined over the $\sigma-$algebra on $Paths^\omega$, given scheduler A and initial state s, denoted by $Pr^\omega_{A,s}$ [26].

Non-zenoness. Owing to the presence of immediate state changes, an IMC might exhibit *Zeno behaviour*, where infinitely many interactive transitions are taken in finite or zero time. This is an unrealistic phenomenon, characterised by an infinite path π, where the time spent on π does not diverge, called a Zeno path. To exclude such unrealistic phenomena, we restrict our attention to models where the probability of Zeno behaviour is zero. This means that $\forall A \in Gen,\ \forall s \in S.\ Pr^\omega_{A,s}(\Pi_{<\infty}) = 0$, where $\Pi_{<\infty}$ is the set of all Zeno paths. This condition implies that starting from any interactive states, we must reach the set of Markov states with probability one. In the remainder of this paper, we therefore restrict to such models.

3 Time Bounded Reachability

CSL model checking of time bounded until properties plays a pivotal role in quantitative evaluation of IMCs. It can be reduced to time bounded reachability analysis, by a well-known technique [2] of making target states absorbing. This section reviews the current state-of-the-art [26, 29] of solving time bounded reachability problems in IMC. Section 4 will discuss how can we improve upon that.

Fixed point characterisation. We first discuss the fixed point characterisation for the maximum probability to reach a set of goal states within an interval of time. For this, let $\mathcal{I}$ and $\mathcal{Q}$ be the set of all nonempty nonnegative real intervals with real and rational bounds respectively. For $I \in \mathcal{I}$ and $t \in \mathbb{R}_{\geq 0}$, we define $I \ominus t = \{x - t \mid x \in I \wedge x \geq t\}$. If $I \in \mathcal{Q}$ and $t \in \mathbb{Q}_{\geq 0}$, then $I \ominus t \in \mathcal{Q}$. Given IMC $\mathcal{M}$, a time interval $I \in \mathcal{I}$ and a set of goal states $G \subseteq S$, the set of all paths that reach the goal states within interval I is denoted by $\Diamond^I G$. Let $p^{\mathcal{M}}_{\max}(s, \Diamond^I G)$ be the maximum probability of reaching the goal states within interval I if starting in state s at time 0. In formal terms, it is the supremum ranging over all possible *Gen* schedulers, of the probability measures on the induced paths: $p^{\mathcal{M}}_{\max}(s, \Diamond^I G) = \sup_{A \in Gen} Pr^{\omega}_{A,s}(\Diamond^I G)$. The next lemma recalls a characterisation of $p^{\mathcal{M}}_{\max}(s, \Diamond^I G)$ as a fixed point. That of $p^{\mathcal{M}}_{\min}(s, \Diamond^I G)$ is dealt with similarly.

Lemma 1 (Fixed Point Characterisation for IMCs [26, Theorem 6.1]). *Let $\mathcal{M}$ be an IMC, $G \subseteq S$ be a set of goal states and $I \in \mathcal{I}$ with $\inf I = a$ and $\sup I = b$. $p^{\mathcal{M}}_{\max} : S \times \mathcal{I} \rightarrowtail [0,1]$ is the least fixed point of the higher-order operator $\Omega : (S \times \mathcal{I} \rightarrowtail [0,1]) \rightarrowtail (S \times \mathcal{I} \rightarrowtail [0,1])$, which is:*

1. *For $s \in S_M$*

$$\Omega(F)(s,I) = \begin{cases} \int_0^b E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, \perp, s') F(s', I \ominus t)\, \mathrm{d}t & s \notin G \\ e^{-E(s)a} + \int_0^a E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, \perp, s') F(s', I \ominus t)\, \mathrm{d}t & s \in G \end{cases}$$

2. *For $s \in S_I$*

$$\Omega(F)(s,I) = \begin{cases} 1 & s \in G \wedge 0 \in I \\ \max_{(s,\alpha,s') \in \longrightarrow} F(s', I) & \textit{otherwise} \end{cases}$$

Interactive Probabilistic Chain. The above characterisation provides an integral equation system of the maximum time interval bounded reachability probability. But this system is in general not directly tractable algorithmically [2]. To circumvent this problem, the fixed point characterisation can be approximated by a digitisation [26, 29] approach. Intuitively, the time interval is split into equally sized intervals, which we call digitisation steps. It is assumed that the digitisation constant δ is small enough such that with high probability it carries at most one Markov transition firing. This assumption reduces an IMC to an Interactive Probabilistic Chain (IPC) [12]. An IPC is a digitised version of IMC, obtained by summarising the behaviour of an IMC at equidistant time points.

Definition 3. *An IPC is a tuple $\mathcal{D} = (S, Act, \longrightarrow, \dashrightarrow_d, s_0)$, where S, Act, $\longrightarrow$ and s_0 are as Definition 1 and $\dashrightarrow_d \subset S \times Dist(S)$ is the set of digitised Markov transitions.*

A digitised Markov transition specifies with which probability a state evolves to its successors after taking one time step. The notion of digitised Markov transition resembles the one-step transition matrix in DTMC. The concepts of closed and open models carry over to IPC. As we do not have the notion of continuous time, paths in IPC can be seen as time-abstract paths in IMC, implicitly still counting digitisation steps, and thus discrete time.

Digitisation from IMC to IPC. We now recall the digitisation that turns an IMC into an IPC. Afterwards, we explain how reachability computation in an IMC can be approximated by analysis on IPC, for which there exists a proved error bound.

Definition 4 (Digitisation [26]). *Given IMC $\mathcal{M} = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ and a digitisation constant δ, $\mathcal{M}_\delta = (S, Act, \longrightarrow, \dashrightarrow_\delta, s_0)$ is an IPC constructed from digitisation of $\mathcal{M}$ with respect to digitisation constant δ and $\dashrightarrow_\delta = \{(s, \mu^s) | s \in S_M\}$, where*

$$\mu^s(s') = \begin{cases} (1 - e^{-E(s)\delta})\mathbf{P}(s, \bot, s') & s' \neq s \\ (1 - e^{-E(s)\delta})\mathbf{P}(s, \bot, s') + e^{-E(s)\delta} & s' = s \end{cases}$$

The digitisation in Definition 4 approximates the original model by assuming that at most one Markov transition in $\mathcal{M}$ can fire in each step of length δ. It is specified by distribution μ^s, which contains the probability of having either one or no Markov transition in $\mathcal{M}$ from state s within a time interval of length δ. Using the fixed point characterisation above, it is possible to relate reachability analysis in an IMC to reachability analysis in its associated IPC [26], together with an error bound. We recall the result here:

Theorem 1 (Error Bound [26]). *Given IMC $\mathcal{M} = (S, Act, \longrightarrow, \dashrightarrow, s_0)$, a set of goal states $G \subseteq S$, a time interval $I \in \mathcal{Q}$ such that $a = \inf I$ and $b = \sup I$ with $0 \leq a < b$. and $\lambda = \max_{s \in S_M} E(s)$. Assume digitisation step $\delta > 0$ is selected such that $b = k_b\delta$ and $a = k_a\delta$ for some $k_b, k_a \in \mathbb{N}$. For all $s \in S$ it holds*

$$p^{\mathcal{M}_\delta}_{\max}(s, \Diamond^{(k_a, k_b]} G) - k_a \frac{(\lambda\delta)^2}{2} \leq p^{\mathcal{M}}_{\max}(s, \Diamond^I G) \leq p^{\mathcal{M}_\delta}_{\max}(s, \Diamond^{(k_a, k_b]} G) + k_b \frac{(\lambda\delta)^2}{2} + \lambda\delta$$

For the proof of Theorem 1 see [26, Theorem 6.5].

Time bounded computation in IPC. We briefly review the maximum time bounded reachability computation in IPC [29]. At its core, a modified *value iteration* algorithm is carried out. Given an IPC, a set of goal states and a step interval, the algorithm iteratively proceeds by taking two different phases. In the first phase, reachability probabilities starting from all interactive states are updated. This is done by selecting the maximum from reachability probabilities of Markov states that are reachable from each interactive state. The second phase updates the reachability probabilities from Markov states by taking a digitised time step. The algorithm iterates until the last digitised time step is processed. For more details about the algorithm we refer to [29].

4 Improving Time Bounded Reachability Computation

In this section, we generalise the previously discussed technique for computing maximum time bounded reachability. As before, we approximate the fixed point characterisation of IMC using a digitisation technique. However instead of considering at most one, we consider at most n Markov transition firing(s) in a digitisation step, for n being an arbitrary natural number. This enables us to establish a tighter error bound. Alternatively, an increased n lets us to choose a larger digitisation constant δ, without compromising the original error bound. A larger digitisation constant implies fewer iterations, thus speeding up the overall runtime of the algorithm.

Higher-order approximation. When developing an approximation of n-th order of the maximum reachability probability, we first restrict ourselves to intervals with zero lower bounds.

Definition 5. *Given IMC $\mathcal{M} = (S, Act, \longrightarrow, \dashrightarrow, s_0)$, a set of goal states $G \subseteq S$, an interval $I \in \mathcal{Q}$ such that $\inf I = 0$ and $\sup I = b$. Assume digitisation step $\delta > 0$ is selected such that $b = k_b\delta$ for some $k_b \in \mathbb{N}$. We define $p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^I G) = 1$ if $s \in G$, and for $s \in S \setminus G$:*

$$p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^I G) = \begin{cases} A_{I,n}^n(s, \delta) & s \in S_M \setminus G \\ \max_{(s,\alpha,s') \in \longrightarrow} p_{\max}^{\mathcal{M}_\delta(n)}(s', \Diamond^I G) & s \in S_I \setminus G \end{cases}$$

and for $0 \le k \le n$ and $0 \le \Delta \le \delta$:

$$A_{I,n}^k(s, \Delta) = \begin{cases} \int_0^\Delta E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, \bot, s') A_{I,n}^{k-1}(s', \Delta \\ \qquad -t)\,\mathrm{d}t + e^{-E(s)\Delta} p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^{I \ominus \delta} G) & s \in S_M \setminus G \wedge k > 0 \\ p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^{I \ominus \delta} G) & s \in S_M \setminus G \wedge k = 0 \\ \max_{(s,\alpha,s') \in \longrightarrow} A_{I,n}^k(s, \Delta) & s \in S_I \setminus G \end{cases}$$

Intuitively $A_{I,n}^k(s, \Delta)$ is the maximum probability to reach G from state s inside $I \ominus (\delta - \Delta)$ by having up to k Markov transition(s) in the first Δ time unit and up to n Markov transition(s) in each digitisation step δ afterwards. This approximation represents the behaviour of the original model more faithfully, thus leading to a better error bound. Theorem 2 quantifies the quality of this approximation.

Theorem 2. *Given IMC $\mathcal{M} = (S, Act, \longrightarrow, \dashrightarrow, s_0)$, a set of goal states $G \subseteq S$, an interval $I \in \mathcal{Q}$ with $\inf I = 0$, $\sup I = b$ and $\lambda = \max_{s \in S_M} E(s)$. Assume digitisation step $\delta > 0$ is selected such that $b = k_b\delta$ for some $k_b \in \mathbb{N}$ and $n > 0$ is the order of approximation. For all $s \in S$ it holds*

$$p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^I G) \le p_{\max}^{\mathcal{M}}(s, \Diamond^I G) \le p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^I G) + 1 - e^{-\lambda b} \Big(\sum_{i=0}^{n} \frac{(\lambda\delta)^i}{i!} \Big)^{k_b}$$

The proof of Theorem 2 is tedious, basically following and generalising the proof of [26, Theorem 6.3]. We provide the proof for the case $n = 2$ in the appendix and discuss how it can be extended to the general case. The core insight is, intuitively

speaking, as follows. We can view the error as the probability of more than n Markov transition(s) firing in at least one digitisation step. Due to independence of the number of Markov transition occurrences in digitisation steps, this probability can be upper bounded by k_b independent Poisson processes, all parametrised with the maximum exit rate exhibited in the IMC. In each Poisson process the probability of at most n Markov transition(s) firing in one digitisation step is $e^{-\lambda\delta}\sum_{i=0}^{n}\frac{(\lambda\delta)^i}{i!}$, therefore the probability of a violation of this assumption in at least one digitisation step is $1-e^{-\lambda b}\big(\sum_{i=0}^{n}\frac{(\lambda\delta)^i}{i!}\big)^{k_b}$.

It is worthwhile to note that open and closed intervals of type $(0,b]$ and $[0,b]$ are treated in the same manner based on Theorem 2. They lead to the same fixed point computation of time bounded reachability, in contrast to bounded until [30]. We can directly extend Definition 5 to intervals with non-zero lower bounds and adapt Theorem 2 accordingly.

Theorem 3. *Given IMC $\mathcal{M}=(S,Act,\longrightarrow,\dashrightarrow,s_0)$, a set of goal states $G\subseteq S$, an interval $I\in\mathcal{Q}$ with $\inf I=a>0$, $\sup I=b>a$ and $\lambda=\max_{s\in S_M}E(s)$. Assume digitisation step $\delta>0$ is selected such that $a=k_a\delta$ and $b=k_b\delta$ for some $k_a,k_b\in\mathbb{N}$ and $n>0$ is the order of approximation. For all $s\in S$ it holds*

$$p_{\max}^{\mathcal{M}_\delta(n)}(s,\Diamond^I G)-\Big(1-e^{-\lambda a}\Big(\sum_{i=0}^{n}\frac{(\lambda\delta)^i}{i!}\Big)^{k_a}\Big)\leq p_{\max}^{\mathcal{M}}(s,\Diamond^I G)\leq p_{\max}^{\mathcal{M}_\delta(n)}(s,\Diamond^I G)+\Big(1-e^{-\lambda b}\Big(\sum_{i=0}^{n}\frac{(\lambda\delta)^i}{i!}\Big)^{k_b}\Big)$$

The proof of Theorem 3 combines the one of Theorem 2 and of [26, Theorem 6.4]. It is worth noting that the digitisation error decreases by decreasing digitisation step δ or increasing the order of approximation n. Further, the error vanishes as n goes to infinity or δ goes to zero.

Improved algorithm. In this section we describe how the result of Theorem 2 and 3 can improve the original time bounded reachability approximation [29]. The structure of the algorithm remains unchanged, but is parametrised with natural n. It computes $p_{\max}^{\mathcal{M}_\delta(n)}$ as the approximation of the maximum reachability probability.

Our objective is to compute maximum probability to reach a set of goal states within a given step interval. First we restrict ourselves to the case that the lower bound of the step interval is zero. Afterwards, we extend it to the general case. Let $\mathcal{M}$ be an IMC, $G\subseteq S$ be a set of goal state and $I\in\mathcal{Q}$ be a nonempty interval with $\inf I=0$ and $\sup I=b$. Assume digitisation step $\delta>0$ is selected such that $b=k_b\delta$ for some $k_b\in\mathbb{N}$. We use $p_{\max}^{\mathcal{M}_\delta(n)}(s,\Diamond^I G)$ to denote the approximate maximum probability of reaching the goal states inside I where we only consider up to n Markov transition firing(s) within each digitisation step. Let $Reach^i(s)$ be the set of states that can be reached from s by only using interactive transitions.

The overall algorithm is depicted in Algorithm 1. It proceeds by backwards unfolding the IMC in an iterative manner, starting from the goal states. At the beginning, all goal states are made absorbing: all of their transitions are removed,

Input : $\mathcal{M}$ is the given IMC, $G \subseteq S$ is the set of goal state, I is the interval with $\inf I = 0$ and $\sup I = b$, $\delta > 0$ such that $b = k_b\delta$ for some $k_b \in \mathbb{N}$
Output: Maximum reachability probabilities starting from all states

begin
make all $s \in G$ in $\mathcal{M}$ absorbing ;
foreach $s \in G$ **do** $p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^{[0,0]}G) := 1$;
foreach $s \in S \setminus G$ **do** $p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^{[0,0]}G) := 0$;
foreach $s \in S_I$ **do** $p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^{[0,0]}G) := \max_{s' \in Reach^i(s) \cap S_M} p_{\max}^{\mathcal{M}_\delta(n)}(s', \Diamond^{[0,0]}G)$;
for $j := k_b - 1$ **to** 0 **do**
// m-phase ;
foreach $s \in S_M$ **do** calculate $p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^{I \ominus j\delta}G)$ as in Definition 5 ;
// i^*-phase ;
foreach $s \in S_I$ **do**
$p_{\max}^{\mathcal{M}_\delta(n)}(s, \Diamond^{I \ominus j\delta}G) := \max_{s' \in Reach^i(s) \cap S_M} p_{\max}^{\mathcal{M}_\delta(n)}(s', \Diamond^{I \ominus j\delta}G)$;
end
end

Algorithm 1: Computing maximum step bounded reachability

and replaced by a digitised Markov self loop (a transition to a Dirac distribution over the source state). The initial value of probability vector is set to one for goal states and to zero otherwise. The algorithm then proceeds by intertwining *m-phases* and *i^*-phases* consecutively for k_b steps. In each iteration, i^*-phase and m-phase update reachability probabilities from interactive and Markov states to the set of goal states respectively. After completing i^*-phase and m-phase at the end of an iteration, the elements of $p_{\max}^{\mathcal{M}_\delta(n)}(\cdot, \Diamond^{I \ominus j\delta}G)$ are updated for both interactive and Markov states.

Phases of an iteration. In the following we explain the functioning of i^*-phase and m-phase in more details. An i^*-phase maximises the reachability probabilities starting from interactive states to the set of goal states. By the law of total probability, this can split into two parts: (1) the probability of reaching Markov states from interactive states in zero time and (2) the probability of reaching goal states from Markov states. The latter has been computed by the m-phase directly preceding the i^*-phase under consideration. The former can be computed by a backward search in the interactive reachability graph underlying the IMC [29]. The number of transitions taken does not matter in this case, because they take zero time each. This step thus needs the set of all Markov states that are reachable from each interactive state s via an arbitrary number of interactive transitions. That set, $Reach^i(s) \cap S_M$, can be precomputed prior to the algorithm. From these sets, the i^*-phase selects states with maximum reachability probability. In an *m-phase*, we update the reachability probabilities starting from Markov states by taking at most n Markov transitions. This step is performed by solving the integral equation in Definition 5 for case $s \in S_M \setminus G$. Restricting the number n of Markov transitions in a digitisation step makes the integral equation in Definition 5 tractable, in contrast to Lemma 1. For instance, in the first-order approximation ($n = 1$) it is enough to consider zero or one Markov transition starting from a Markov state. Owing to this

assumption the resulting model ($\mathcal{M}_\delta(1)$) is equivalent to the induced IPC ($\mathcal{M}_\delta$) from the original model with respect to digitisation step δ. For the second-order approximation we need to consider up to two Markov transitions starting from a Markov state.

Example 2. We now discuss by example how i^*- and m-phases are performed for $n = 2$. Assume Figure 2 is a fragment of an IMC $\mathcal{C}$ with a set of goal states G. Given time interval $I = [0, b]$ with $b > 0$ and digitisation step δ, the vector $p_{\max}^{\mathcal{C}_\delta(2)}(\cdot, \Diamond^{I\ominus\delta} G)$ has been computed for all states of $\mathcal{C}$. The aim is to compute $p_{\max}^{\mathcal{C}_\delta(2)}(s_0, \Diamond^I G)$. From Definition 5 we have:

$$p_{\max}^{\mathcal{C}_\delta(2)}(s_0, \Diamond^I G) = A^2_{I,2}(s_0, \Delta) = \int_0^\delta 2e^{-2t} A^1_{I,2}(s_1, \delta - t)\,\mathrm{d}t + e^{-2\delta} p_{\max}^{\mathcal{C}_\delta(2)}(s_0, \Diamond^{I\ominus\delta} G)$$

For s_1 we have $A^1_{I,2}(s_1, \delta{-}t) = \max\{A^1_{I,2}(s_3, \delta{-}t), A^1_{I,2}(s_5, \delta{-}t)\}$, since $Reach^i(s_1) \cap S_M = \{s_3, s_5\}$. From Definition 5 for s_3 and s_5 we have:

$$\begin{aligned} A^1_{I,2}(s_3, \delta - t) &= \int_0^{\delta - t} 3e^{-3t'} A^0_{I,2}(s_4, \delta - t - t')\,\mathrm{d}t' + e^{-3(\delta - t)} p_{\max}^{\mathcal{C}_\delta(2)}(s_3, \Diamond^{I\ominus\delta} G) \\ &= (1 - e^{-3(\delta - t)}) p_{\max}^{\mathcal{C}_\delta(2)}(s_4, \Diamond^{I\ominus\delta} G) + e^{-3(\delta - t)} p_{\max}^{\mathcal{C}_\delta(2)}(s_3, \Diamond^{I\ominus\delta} G) \end{aligned}$$

Similar calculations give:
$A^1_{I,2}(s_5, \delta - t) = (1 - e^{-5(\delta - t)}) p_{\max}^{\mathcal{C}_\delta(2)}(s_6, \Diamond^{I\ominus\delta} G) + e^{-5(\delta - t)} p_{\max}^{\mathcal{C}_\delta(2)}(s_5, \Diamond^{I\ominus\delta} G)$.

Generalisation to intervals with non-zero lower bound. We can generalise time bounded reachability computation just discussed to intervals with non-zero error bound, following a recipe discussed in [2]. Assume we choose interval I such that $\inf I = a > 0$ and $\sup I = b > a$. We break the interval into two parts, first from b down to a and second from a down to zero. Within the first, we are interested in reaching one of the goal states, as a result we make the goal states absorbing. Nevertheless, within the second, it does not matter that the model is in one of the goal states, which consequently leads us to ignore goal states and reintroduce them as before. Accordingly the algorithm proceeds as follows. In the first part ($[0, b - a]$), goal states are made absorbing and reachability probabilities are computed by running Algorithm 1. The result will be used as the initial vector of the next step. Then, goal states are treated as normal states, so we undo absorbing of goal states and set $G = \emptyset$. However other calculations remain the same as before.

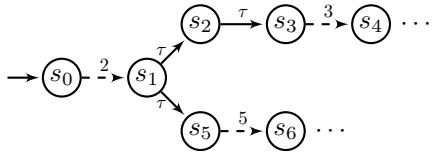

Fig. 2. An exemplary IMC fragment

Complexity and efficiency. The key innovation of this approach lies in both the precision and the efficiency of the computation. Following Theorems 2 and 3, the number of iterations required to guarantee accuracy level ϵ can be calculated by

determining the least k_b such that $1 - e^{-\lambda b}\big(\sum_{i=0}^{n} \frac{(\lambda\delta)^i}{i!}\big)^{k_b} \leq \epsilon$. The inequality however does not have closed-form solution with respect to k_b. Routine calculus allows us to derive that $1 - e^{-\lambda b}\big(\sum_{i=0}^{n} \frac{(\lambda\delta)^i}{i!}\big)^{k_b} \leq k_b \frac{(\lambda\delta)^{n+1}}{(n+1)!}$ which is tight in our setting, since $\lambda\delta$ is very small. Thus, we instead consider inequality $k_b \frac{(\lambda\delta)^{n+1}}{(n+1)!} \leq \epsilon$ which leads to $k_b \geq \lambda b\big(\frac{\lambda b}{(n+1)!\epsilon}\big)^{\frac{1}{n}}$. This shows how the number of iterations required to achieve a predefined accuracy level decreases by increasing the order of approximation n. In other words, using higher-order approximations gives the same error bound in less iterations.

To shed some light on this, we compare the complexity of the original first-order and the second-order instance of the novel approximation. Given accuracy level ϵ and IMC $\mathcal{M}$ as before, assume $N = |S|$ and $M = |\longrightarrow| + |\dashrightarrow|$. The best known complexity for the precomputation of set $\mathit{Reach}^i(\cdot)$ for all interactive states and hence of $\mathit{Reach}^i(\cdot) \cap S_M$ is $\mathcal{O}(N^{2.376})$ [11]. Instantiating the inequality above for $n = 2$ gives $\mathcal{O}\left(\sqrt{\frac{(b\lambda)^3}{\epsilon}}\right)$ as the complexity of the iteration count. Since the size of $\mathit{Reach}^i(s) \cap S_M$ for a given state s is at most N, the complexity of the i^*-phase is $\mathcal{O}(N^2)$. m-phase contains one step reachability computations from Markov states by considering zero, one or two Markov transitions which has the respective complexities $\mathcal{O}(N)$, $\mathcal{O}(MN)$ and $\mathcal{O}(M^2)$. Thus the resulting complexity is $\mathcal{O}\left(N^{2.376} + \left(M^2 + MN + N^2\right)\sqrt{\frac{(b\lambda)^3}{\epsilon}}\right)$, while the complexity of the first-order approximation is $\mathcal{O}\left(N^{2.376} + (M + N^2)\frac{(b\lambda)^2}{\epsilon}\right)$ [29]. We observe that the per iteration complexity of the second-order approximation is higher, but since in almost all cases M is at least N this is a negligible disadvantage. At the same time, the number of iterations (the respective last terms) is much less. Therefore the efficiency of the second-order approximation compares favourably to the original first-order approximation, at least in theory. In the next section we compare the complexity of both algorithms in practice.

5 A Simplified Empirical Evaluation

This section reports on empirical results with an implementation that harvests the theoretical advances established, but is simplified in one dimension: Our current implementation keeps the scheduler decisions constant over each time interval of length δ, even though a timed scheduler may perform slightly better by adjusting the decision during the interval, and not at interval boundaries only. We do not yet have an error bound for the deviation introduced by this simplification. In light of the above discussion, we consider $n = 2$, thus we use a second-order approximation, and compare with the original first-order approximation.

Case study. As a case study we consider a replicated file system as it is used as part of the Google search engine [10]. The IMC specification is available as an example of IMCA tool [18]. The Google File System (GFS) splits files into chunks of equal size maintained by several chunk servers. If a user wants to access a chunk, it asks

a master server which store the address of all chunks. Then the user can directly access the appropriate chunk server to read/write the chunk. The model contains three parameters, N_{cs} is the number of chunk server, C_s is the number of chunks a server may store, and C_t is the total number of chunks.

Evaluation. We set $C_s = 5000$ and $C_t = 100000$ and change the number of chunk servers N_{cs}. The set of goal states G is defined as states in which the master server is up and there is at least one copy of each chunk available. We compute minimum and maximum time bounded reachability with respect to the set of goal states G using both the first- and the second-order approximations on different intervals of time. The former has been implemented in the IMCA tool [18], and our implementation is derived from that. All experiments were conducted on a single core of a 2.5 GHz Intel Core i5 processor with 4GB RAM running on Linux. The computation times of both algorithm under different parameter settings are reported in Table 1.

As stated before, the second-order algorithm takes less iterations for computing reachability to guarantee accuracy ϵ. The computation times reported apparently show a beneficial effect, with the speedup depending on different parameters. Table 1 indicates that the speedup gets higher with increasing λ and with increasing interval upper bounds.

Table 1. Reachability computation time in the Google file system.

					$\mathcal{M}_\delta$ time(s)		$\mathcal{M}_\delta(2)$ time(s)	
N_{cs}	$\lvert S\rvert$	$\lvert G\rvert$	ϵ	I	min	max	min	max
10	1796	408	10^{-3}	$[0, 0.1]$	124.8	115.0	18.6	21.4
			10^{-3}	$[0, 0.4]$	2021.0	1823.6	145.0	165.1
			10^{-4}	$[0, 0.1]$	1308.9	1188.1	56.7	66.0
			10^{-4}	$[0.01, 0.04]$	232.8	214.0	17.1	21.9
20	7176	1713	10^{-4}	$[0, 0.01]$	319.9	308.5	52.2	54.0
			10^{-5}	$[0.005, 0.015]$	5564.9	6413.0	179.4	219.1

6 Conclusions

This paper has presented an improvement of time bounded reachability computations in IMC, based on previous work [29], which has established a digitisation approach for IMC, together with a stable error bound. We have extended this theoretical result by assuming at most n Markov transitions to fire in each digitisation step, where previously $n = 1$ was assumed. In practice, setting $n = 2$ already provides a much tighter error bound, and thus saves considerable computation time. We have demonstrated the effectiveness of the approach in our empirical evaluation with speedups of more than one order of magnitude, albeit for a simplified scheduler scenario.

Lately, model checking of *open* IMC has been studied, where the IMC is considered to be placed in an unknown environment that may delay or influence the IMC behaviour via synchronisation [9]. The approach resorts to the approximation scheme laid out in [29], which we have improved upon in the present paper. Therefore, our improvement directly carries over to the open setting. As a future work, we intend to further generalise the proposed algorithm to Markov Automata [15, 14, 20].

Acknowledgement. We thank Lijun Zhang for helpful discussions and comments, and Dennis Guck for developing and sharing the original implementation of the algorithm and the case study as a part of IMCA.

References

1. Adnan Aziz, Kumud Sanwal, Vigyan Singhal, Robert Brayton: Towards Performance Prediction of Verifying Continuous Time Markov Chains. *CAV*, LNCS 1102:269–276, Springer (1996)
2. Christel Baier, Boudewijn Haverkort, Holger Hermanns, Joost-Pieter Katoen: Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29 (6): 524–541 (2003)
3. Christel Baier, Holger Hermanns, Joost-Pieter Katoen, Boudewijn Haverkort: Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theoretical Computer Science*, 345:2–26 (2005)
4. Christel Baier, Joost-Pieter Katoen: Principles of Model Checking. MIT Press (2008)
5. Andrea Bianco, Luca de Alfaro: Model checking of probabilistic and nondeterministic systems. *FSTTCS*, LNCS 1026:499–513, Springer (1995)
6. Hichem Boudali, Pepijn Crouzen, Boudewijn R. Haverkort, Matthias Kuntz, Mariëlle Stoelinga: Architectural dependability evaluation with Arcade. *DSN*, IEEE 512–521 (2008)
7. Eckard Böde, Marc Herbstritt, Holger Hermanns, Sven Johr, Thomas Peikenkamp, Reza Pulungan, Jan Rakow, Ralf Wimmer, Bernd Becker: Compositional Dependability Evaluation for STATEMATE. *IEEE Transactions on Software Engineering*, 35(2):274–292 (2009)
8. Mario Bravetti, Holger Hermanns, Joost-Pieter Katoen: YMCA — Why Markov Chain Algebra ?. *Electronic Notes in Theoretical Computer Science*, 162:107–112 (2006)
9. Tomáš Brázdil, Holger Hermanns, Jan Krčál, Jan Křetínský and Vojtěch Řehák: Verification of Open Interactive Markov Chains. *FSTTCS*, 474–485 (2012)
10. Lucia Cloth, Boudewijn R. Haverkort: Model Checking for Survivability. *QEST*, IEEE 145–154 (2005)
11. Don Coppersmith, Shmuel Winograd: Matrix Multiplication via Arithmetic Progressions. *STOC*, 1–6 (1987)
12. Nicolas Coste, Holger Hermanns, Etienne Lantreibecq, Wendelin Serwe: Towards Performance Prediction of Compositional Models in Industrial GALS Designs. *CAV*, LNCS 5643:204–218, Springer (2009)
13. Nicolas Coste, Hubert Garavel, Holger Hermanns, Richard Hersemeule, Yvain Thonnart, Meriem Zidouni: Quantitative Evaluation in Embedded System Design: Validation of Multiprocessor Multithreaded Architectures. *DATE*, IEEE 88–89 (2008)
14. Yuxin Deng, Matthew Hennessy: On the Semantics of Markov Automata. *ICALP*, LNCS 6756:307–318, Springer (2011)
15. Christian Eisentraut, Holger Hermanns, Lijun Zhang: On Probabilistic Automata in Continuous Time. *LICS*, 342–351 (2010)
16. Marie-Aude Esteve, Joost-Pieter Katoen, Viet Yen Nguyen, Bart Postma, Yuri Yushtein: Formal correctness, safety, dependability and performance analysis of a satellite. *ICSE*, 1022 – 1031 (2012)
17. Hubert Garavel, Frédéric Lang, Radu Mateescu, Wendelin Serwe: CADP 2010: A Toolbox for the Construction and Analysis of Distributed Processes. *TACAS*, LNCS 6605:372-387, Springer (2011)

18. Dennis Guck: Quantitative Analysis of Markov Automata Master Thesis, RWTH Aachen University (2012)
19. Dennis Guck, Tingting Han, Joost-Pieter Katoen, Martin R. Neuhäusser: Quantitative Timed Analysis of Interactive Markov Chains. *NASA Formal Methods*, 8–23 (2012)
20. Hassan Hatefi, Holger Hermanns: Model Checking Algorithms for Markov Automata. *ECEASST*, 53 (2012)
21. Holger Hermanns: Interactive Markov Chains: The Quest for Quantified Quality. LNCS 2428, Springer (2002)
22. Holger Hermanns, Ulrich Herzog, Joost-Pieter Katoen: Process algebra for performance evaluation. *Theoretical Computer Science*, 274(1-2):43–87 (2002)
23. Holger Hermanns, Joost-Pieter Katoen: Automated compositional Markov chain generation for a plain-old telephone system. *Science of Computer Programming*, 36(1):97–127 (2000)
24. Jane Hillston: A Compositional Approach to Performance Modelling. Cambridge University Press (1996)
25. Hermenegilda Macià, Valentin Valero, Fernando Cuartero, M. Carmen Ruiz: sPBC: A Markovian Extension of Petri Box Calculus with Immediate Multiactions. *Fundamenta Informaticae*, 87(3–4):367–406 (2008)
26. Martin R. Neuhäusser: Model Checking Nondeterministic and Randomly Timed Systems. PhD Thesis, RWTH Aachen University and University of Twente (2010)
27. Boudewijn R. Haverkort, Matthias Kuntz, Anne Remke, S. Roolvink, Mariëlle Stoelinga: Evaluating repair strategies for a water-treatment facility using Arcade. *DSN*, 419–424 (2010)
28. Reza Pulungan: Reduction of Acyclic Phase-Type Representations. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany (2009)
29. Lijun Zhang, Martin R. Neuhäusser: Model Checking Interactive Markov Chains. *TACAS*, LNCS 6015:53–68, Springer (2010)
30. Lijun Zhang, David N. Jansen, Flemming Nielson, Holger Hermanns: Automata-Based CSL Model Checking. *ICALP*, 271–282 (2011)

Appendix

Proof of Theorem 2

We present the proof of Theorem 2 in a restricted setting and afterwards briefly discuss how to extend it to cover the entirety of the theorem. We assume that $I = [0, b]$ and focus on the case $n = 2$. Lemma 1 for $s \in S_M \setminus G$ can be rewritten [26, Section 6.3.1] into

$$p^{\mathcal{M}}_{\max}(s, \Diamond^I G) = \int_0^\delta E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, \bot, s') p^{\mathcal{M}}_{\max}(s', \Diamond^{I \ominus t} G)\, \mathrm{d}t + e^{-E(s)\delta} p^{\mathcal{M}}_{\max}(s, \Diamond^{I \ominus \delta} G) \tag{1}$$

The following holds from Definition 5 for $s \in S_M \setminus G$ and $n = 2$:

$$p^{\mathcal{M}_\delta(2)}_{\max}(s, \Diamond^I G) = A^2_{I,2}(s, \delta) = \int_0^\delta E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, \bot, s') A^1_{I,2}(s', \delta - t)\, \mathrm{d}t + e^{-E(s)\delta} p^{\mathcal{M}_\delta(2)}_{\max}(s, \Diamond^{I \ominus \delta} G) \tag{2}$$

We have to prove that:

$$p^{\mathcal{M}_\delta(2)}_{\max}(s, \Diamond^I G) \leq p^{\mathcal{M}}_{\max}(s, \Diamond^I G) \leq p^{\mathcal{M}_\delta(2)}_{\max}(s, \Diamond^I G) + 1 - e^{-\lambda b}\Big(\sum_{i=0}^{2} \frac{(\lambda\delta)^i}{i!}\Big)^{k_b}$$

In the following, we prove the upper bound of the approximation. For the proof of lower bound see [26, Lemma 6.6].

Proof. The proof is by induction over k_b:
1. $k_b = 1$: We consider two cases:

a. $s \in S_M \setminus G$: Let Π^δ be the set of paths that reach G within δ time unit. In the approximation we measure the set of paths that have at most two Markovian jumps and then reach G. Let this set be denoted by $\Pi^\delta_{\leq 2}$. Since we have $\Pi^\delta = \Pi^\delta_{\leq 2} \cup \Pi^\delta_{>2}$ and $\Pi^\delta_{\leq 2}$ and $\Pi^\delta_{>2}$ are disjoint, we have: $Pr^\omega_{A,s}(\Pi^\delta) - Pr^\omega_{A,s}(\Pi^\delta_{\leq 2}) = Pr^\omega_{A,s}(\Pi^\delta_{>2})$. The probability $Pr^\omega_{A,s}(\Pi^\delta_{>2})$ can be bounded by the probability of more than two arrivals in a Poisson process with the largest exit rate appearing in the IMC within a time interval of length δ. For the Poisson process, this probability is $1 - e^{-\lambda\delta}\big(\sum_{i=0}^{2} \frac{(\lambda\delta)^i}{i!}\big)$.

b. $s \in S_I \setminus G$: This case reduces to case 1.a as follows. We have

$$p^{\mathcal{M}}_{\max}(s, \Diamond^{I\ominus t} G) = \max_{s' \in Reach^i(s) \cap S_M} p^{\mathcal{M}}_{\max}(s', \Diamond^{I\ominus t} G)$$
$$p^{\mathcal{M}_\delta(2)}_{\max}(s, \Diamond^{I\ominus t} G) = \max_{s' \in Reach^i(s) \cap S_M} p^{\mathcal{M}_\delta(2)}_{\max}(s', \Diamond^{I\ominus t} G)$$

From the above equations there exists $s' \in S_M$ such that $p^{\mathcal{M}}_{\max}(s, \Diamond^I G) = p^{\mathcal{M}}_{\max}(s', \Diamond^I G)$. Because s' is a Markov state, the upper bound for s' is deployed to s.

2. $k_b - 1 \rightsquigarrow k_b$: We assume the upper bound holds for $k_b - 1$:

$$p^{\mathcal{M}}_{\max}(s, \Diamond^{I\ominus\delta} G) \leq p^{\mathcal{M}_\delta(2)}_{\max}(s, \Diamond^{I\ominus\delta} G) + 1 - e^{-\lambda(k_b-1)\delta}\Big(\sum_{i=0}^{2} \frac{(\lambda\delta)^i}{i!}\Big)^{k_b-1} \tag{3}$$

Assume $B^i(s,t) = p^{\mathcal{M}}_{\max}(s, \Diamond^{I\ominus t} G) - A^i_{I,2}(s, \delta - t)$ for $0 \leq t \leq \delta,\ i = \{0,1,2\}$ and $C(s) = p^{\mathcal{M}}_{\max}(s, \Diamond^{I\ominus\delta} G) - p^{\mathcal{M}_\delta(2)}_{\max}(s, \Diamond^{I\ominus\delta} G)$. We consider two cases:

a. $s \in S_M \setminus G$: From Eq. 1 and 2 we have:

$$\begin{aligned} B^2(s,0) &= p^{\mathcal{M}}_{\max}(s, \Diamond^I G) - p^{\mathcal{M}_\delta(2)}_{\max}(s, \Diamond^I G) \\ &= \int_0^\delta E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, \perp, s') B^1(s', t)\, \mathrm{d}t + e^{-E(s)\delta} C(s) \end{aligned} \tag{4}$$

We try to find an upper bound for $B^1(s', t)$ for $s' \in S_M$:

$$\begin{aligned} B^1(s',t) &= p^{\mathcal{M}}_{\max}(s', \Diamond^{I\ominus t} G) - A^1_{I,2}(s', \delta - t) \\ &= \int_0^{\delta - t} E(s') e^{-E(s')\tau} \sum_{s'' \in S} \mathbf{P}(s', \perp, s'') B^0(s'', t + \tau)\, \mathrm{d}\tau \\ &\quad + e^{-E(s')(\delta - t)} C(s') \end{aligned} \tag{5}$$

Now we find an upper bound for $B^0(s'', t+\tau)$. For $s'' \in S_M$ we have:

$$\begin{aligned}
B^0(s'', t+\tau) &= p^{\mathcal{M}}_{\max}(s'', \Diamond^{I\ominus(t+\tau)}G) - A^0_{I,2}(s', \delta - t - \tau) \\
&= \int_0^{\delta-t-\tau} E(s'')e^{-E(s'')u} \sum_{v\in S} \mathbf{P}(s'', \bot, v) p^{\mathcal{M}}_{\max}(v, \Diamond^{I\ominus(t+\tau+u)}G)\,\mathrm{d}u \\
&\quad + e^{-E(s')(\delta-t-\tau)} p^{\mathcal{M}}_{\max}(s'', \Diamond^{I\ominus\delta}G) - p^{\mathcal{M}_\delta(2)}_{\max}(s'', \Diamond^{I\ominus\delta}G) \\
&= \int_0^{\delta-t-\tau} E(s'')e^{-E(s'')u} \sum_{v\in S} \mathbf{P}(s'', \bot, v) p^{\mathcal{M}}_{\max}(v, \Diamond^{I\ominus(t+\tau+u)}G)\,\mathrm{d}u \\
&\quad - (1 - e^{-E(s')(\delta-t-\tau)}) p^{\mathcal{M}_\delta(2)}_{\max}(s'', \Diamond^{I\ominus\delta}G) \\
&\quad + e^{-E(s'')(\delta-t-\tau)} C(s'') \qquad (6)
\end{aligned}$$

We know that:

$$\int_0^{\delta-t-\tau} E(s'')e^{-E(s'')u} \sum_{v\in S} \mathbf{P}(s'', \bot, v) p^{\mathcal{M}}_{\max}(v, \Diamond^{I\ominus(t+\tau+u)}G)\,\mathrm{d}u \leq 1 - e^{-E(s'')(\delta-t-\tau)}$$

Plugging the above inequality and 3 into 6 gives:

$$B^0(s'', t+\tau) \leq 1 - e^{-\lambda(k_b\delta - t - \tau)} \Big(\sum_{i=0}^{2} \frac{(\lambda\delta)^i}{i!} \Big)^{k_b - 1} \qquad (7)$$

Plugging 3 and 7 into 5 gives:

$$B^1(s', t) \leq 1 - e^{-\lambda(k_b\delta - t)} \Big(\sum_{i=0}^{2} \frac{(\lambda\delta)^i}{i!} \Big)^{k_b - 1} (1 + \lambda(\delta - t)) \qquad (8)$$

Note that Eq. 7 and 8 are still valid for $s', s'' \in S_I \setminus G$ with the same argument described in 1.b. Finally plugging 3 and 8 into 4 gives:

$$B^2(s, 0) = p^{\mathcal{M}}_{\max}(s, \Diamond^I G) - p^{\mathcal{M}_\delta(2)}_{\max}(s, \Diamond^I G) \leq 1 - e^{-\lambda b} \Big(\sum_{i=0}^{2} \frac{(\lambda\delta)^i}{i!} \Big)^{k_b}$$

b. $s \in S_I \setminus G$: In this case the proof is similar to 1.a.

This proof can directly be extended to intervals with open bounds and to intervals with nonzero lower bounds. Furthermore it can be embedded into an induction on n, thereby showing the theorem for any natural n. We need to skip these cases because of space limitations.