Mean-payoff Optimization in Continuous-time Markov Chains with Parametric Alarms

CHRISTEL BAIER and CLEMENS DUBSLAFF, Faculty of Computer Science, TU Dresden, Germany
L'UBOŠ KORENČIAK, ANTONÍN KUČERA, and VOJTĚCH ŘEHÁK, Faculty of Informatics, Masaryk University, Czech Republic

Continuous-time Markov chains with alarms (ACTMCs) allow for alarm events that can be non-exponentially distributed. Within *parametric* ACTMCs, the parameters of alarm-event distributions are not given explicitly and can be the subject of parameter synthesis. In this line, an algorithm is presented that solves the ε-optimal parameter synthesis problem for parametric ACTMCs with long-run average optimization objectives. The approach provided in this article is based on a reduction of the problem to finding long-run average optimal policies in semi-Markov decision processes (semi-MDPs) and sufficient discretization of the parameter (i.e., action) space. Since the set of actions in the discretized semi-MDP can be very large, a straightforward approach based on an explicit action-space construction fails to solve even simple instances of the problem. The presented algorithm uses an enhanced policy iteration on *symbolic* representations of the action space. Soundness of the algorithm is established for parametric ACTMCs with alarm-event distributions that satisfy four mild assumptions, fulfilled by many kinds of distributions. Exemplifying proofs for the satisfaction of these requirements are provided for Dirac, uniform, exponential, Erlang, and Weibull distributions in particular. An experimental implementation shows that the symbolic technique substantially improves the efficiency of the synthesis algorithm and allows us to solve instances of realistic size.

CCS Concepts: • **Theory of computation** → **Continuous optimization**; • **Mathematics of computing** → *Stochastic processes*; *Distribution functions*; • **Computing methodologies** → *Policy iteration*;

Additional Key Words and Phrases: Parameter synthesis, non-Markovian distributions, semi-Markov decision processes, generalized semi-Markov processes, Markov regenerative processes

ACM Reference format:
Christel Baier, Clemens Dubslaff, L'uboš Korenčiak, Antonín Kučera, and Vojtěch Řehák. 2019. Mean-payoff Optimization in Continuous-time Markov Chains with Parametric Alarms. *ACM Trans. Model. Comput. Simul.* 29, 4, Article 28 (December 2019), 26 pages.
https://doi.org/10.1145/3310225

The authors are partly supported by the Czech Science Foundation, Grant No. 18-11193S, by the DFG through the Collaborative Research Center SFB 912–HAEC, the Excellence Initiative by the German Federal and State Governments (cluster of excellence cfAED), and the DFG-projects BA-1679/11-1 and BA-1679/12-1. Access to Maple license owned by MetaCentrum provided under the program CESNET LM2015042 is greatly appreciated.
Authors' addresses: C. Baier and C. Dubslaff, Faculty of Computer Science, TU Dresden, Nöthnitzer Strasse 46, 01062, Dresden, Saxony, Germany; emails: {christel.baier, clemens.dubslaff}@tu-dresden.de; Ľ. Korenčiak, A. Kučera, V. Řehák, Faculty of Informatics, Masaryk University, Botanická 68a, Brno, 60200, Czech Republic; emails: {korenciak, kucera, rehak}@fi.muni.cz.

1049-3301/2019/12-ART28 $15.00
https://doi.org/10.1145/3310225

1 INTRODUCTION

Mean-payoff is widely accepted as an appropriate concept for measuring long-run average performance of systems with rewards or costs. In this article, we study the problem of synthesizing parameters for possibly *non-exponentially* distributed events in stochastic systems to achieve an ε-optimal mean-payoff. One simple example of such events are *timeouts* that are widely used, e.g., to prevent deadlocks or to ensure some sort of progress in distributed systems. In practice, timeout durations are usually determined in an *ad hoc* manner, requiring a considerable amount of expertise and experimental effort. This naturally raises the question of automating this design step, i.e., is there an algorithm synthesizing *optimal* timeouts?

The underlying stochastic model this article relies on is provided by *continuous-time Markov chains with alarms (ACTMCs)*. Intuitively, ACTMCs extend continuous-time Markov chains by generally distributed *alarm events*, where at most one alarm is enabled during a system execution and non-alarm events can disable the alarm. In *parametric ACTMCs*, every alarm distribution depends on one single parameter ranging over a given interval of eligible values. For example, a timeout is a Dirac distributed alarm event where the parameter specifies its duration. A *parameter function* assigning to every alarm a parameter value within an interval of eligible values yields a (non-parametric) ACTMC. We propose an algorithm that synthesizes a parameter function for arbitrarily small $\varepsilon > 0$ achieving an ε-optimal mean-payoff.

Motivating example. To get some intuition about the described task, consider a dynamic power management of a disk drive inspired by [36]. The behavior of the disk drive can be described as follows (see Figure 1): At every moment, the drive is either *active* or *asleep*, and it maintains a queue of incoming I/O operations of capacity N. The events of *arriving* and *completing* an I/O operation have exponential distributions with rates 1.39 and 12.5, respectively. When the queue is full, all newly arriving I/O operations are rejected. The I/O operations are performed only in the *active* mode. When the drive is *active* and the queue becomes empty, an internal timer is set to a given value d_s. If then no further I/O request is received within the next d_s time units, then the *sleep* event changes the mode to *asleep*. When the drive is *asleep* and some I/O operation arrives, the internal timer is set to a given value d_w and after d_w time the *wakeup* event changes the mode to *active*. We annotate costs in terms of energy per time unit or instantaneous energy costs for events. The power consumption is 4 and 2 per time unit in the states *active* and *asleep*, respectively. Moving from *asleep* to *active* requires 4 units of energy. Rejecting a newly arrived I/O request when the queue is full is undesirable, penalized by costs of 6. All other transitions incur with cost 1. Obviously, the designer of the disk drive controller has some freedom in choosing the delays d_s and d_w, i.e., they are free parameters of Dirac distributions. However, d_w cannot be lower than the minimal time required to wake up the drive, which is constrained by the physical properties of the hardware used in the drive. Further, there is also a natural upper bound on d_s and d_w given by the capacity of the internal timer. Observe that if d_s is too small, then many costly transitions from *asleep* to *active* are performed; and if d_s is too large, a lot of time is wasted in the more power consuming $active_0$ state. Similarly, if d_w is too small, a switch to the *active* mode is likely to be invoked with a few I/O operations in the queue, and more energy could have been saved by waiting somewhat longer; and if d_w is too large, the risk of rejecting newly arriving I/O operations increases. Now, we may ask the following instance of an optimal-parameter synthesis problem we deal with in this article:

> *What values should a designer assign to the delays d_s and d_w such that the long-run average power consumption is minimized?*

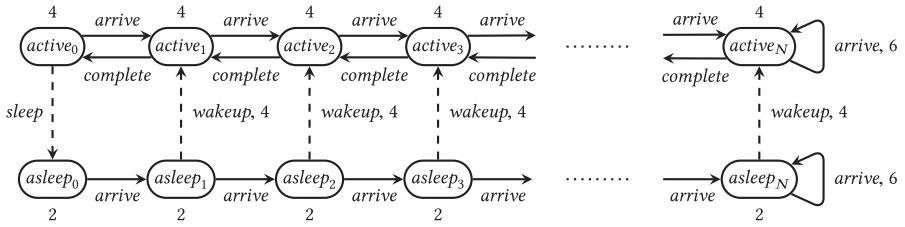

Fig. 1. Dynamic power manager of a disk drive.

Contribution. The main result of our article is a *symbolic* algorithm for ε-optimal parameter synthesis that is *generic* in the sense that it is applicable to all ACTMCs where the optimized alarm events satisfy four abstractly formulated criteria. We show that these criteria are fulfilled, e.g., for timeout events modeled by Dirac distributions, uniformly distributed alarms (used, e.g., in variants of the CSMA/CD protocol [8]), Erlang distributions (used in telecommunications traffic engineering [1]), and Weibull distributions (used to model hardware failures [33]). Our synthesis approach for such parametric ACTMCs fulfilling the aforementioned criteria then is as follows: For a given $\varepsilon > 0$, our algorithm first computes a sufficiently small discretization step such that an ε-optimal parameter function exists even when its range is restricted to the discretized parameter values. The semantics of the parametric ACTMC over discretized parameter values is provided by a semi Markov decision process (semi-MDP), turning the parameter synthesis problem into a scheduler synthesis problem in semi-MDPs. Since the discretization step is typically very small, an *explicit* construction of this semi-MDP is computationally infeasible. Instead, our algorithm employs a symbolic variant of the standard policy iteration algorithm for finding schedulers that have ε-optimal mean-payoff. We start with *some* parameter function that is gradually improved until a fixed point is reached. In each improvement step, our algorithm computes a small *candidate subset* of the discretized parameter values such that a possible improvement is realizable by one of these candidate values. This is achieved by designing a suitable *ranking* function for each of the optimized events such that an optimal parameter value is the minimal value of the ranking function in the interval of eligible parameter values. Then, the algorithm approximates the roots of the symbolic derivative of the ranking function and constructs the candidate subset by collecting all discretized parameter values close to the approximated roots. This symbolic approach leads to a drastic efficiency improvement that makes the resulting algorithm applicable to problems of realistic size. We show correctness of the algorithm for the subclass of parametric ACTMCs with *localized alarms*, i.e., parametric ACTMCs where for each alarm event there is a uniquely defined state where the value of the alarm is set. In Section 3.4, we also discuss solution methods applicable on ACTMCs with non-localized alarms, which is an NP-hard problem. This article is an extended version of the conference paper published in [4].

Related work. Synthesis of optimal timeouts guaranteeing quantitative properties in timed systems was considered in [15]. There are various parametric formalisms for timed systems that deal with some sort of synthesis, such as parametric timed automata [2, 22, 23], parametric one-counter automata [18], parametric timed Petri nets [37], or parametric Markov models [19]. However, all works referred to above do not consider models with continuous-time distributions. Consequently, they synthesize different parameters than we do. Contrary, the synthesis of appropriate rates in CTMCs was efficiently solved in [12, 14, 20, 21]. A special variant of ACTMC, where only alarms with Dirac distributions are allowed, has been considered in [9, 24, 25]. Their algorithms synthesize ε-optimal alarm parameters toward an expected reachability objective. Using a simulation-based approach, the optimization environment of the tool TimeNET is able to

approximate locally optimal distribution parameters in stochastic Petri nets, e.g., using methods as simulated annealing, hill climbing, or genetic algorithms. To the best of our knowledge, we present the first algorithm that approximates globally mean-payoff optimal parameters of non-exponential distributions in continuous-time models.

The (non-parametric) ACTMCs form a subclass of Markov regenerative processes (MRP) [3, 13, 32]. Alternatively, ACTMCs can be also understood as a generalized semi-Markov processes (GSMPs) with at most one non-exponential event enabled in each state or as bounded stochastic Petri nets (SPNs) [17] with at most one non-exponential transition enabled in any reachable marking [13]. Note that ACTMCs are analytically tractable thanks to methods for subordinated Markov-chains (SMCs) that allow for efficient computation of transient and steady-state distributions [13, 29]. Recently, methods for computing steady-state distributions in larger classes of regenerative GSMPs or SPNs have been presented in [31]. We did not incorporate this method into our approach as our methods to compute sufficiently small discretization and approximation precisions to guarantee ε-optimal mean-payoffs are not directly applicable for this class of systems. To the best of our knowledge there are no efficient algorithms with a guaranteed error for computation of steady-state distribution for a general GSMP (or SPN). For some cases it is even known that the steady-state distribution does not exist [10].

2 PRELIMINARIES

Let $\mathbb{N}_{\geq 0}$, $\mathbb{N}_{>0}$, $\mathbb{Q}_{\geq 0}$, $\mathbb{Q}_{>0}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$ denote the set of all non-negative integers, positive integers, non-negative rational numbers, positive rational numbers, non-negative real numbers, and positive real numbers, respectively. For a countable set E, we denote by $\mathcal{D}(E)$ the set of discrete probability distributions over E, i.e., functions $\mu : E \to \mathbb{R}_{\geq 0}$ where $\sum_{e \in E} \mu(e) = 1$. The *support* of μ is the set of all $e \in E$ with $\mu(e) > 0$. A *cumulative distribution function (CDF)* of a real-valued random variable $X : E \to \mathbb{R}_{\geq 0}$ is a nondecreasing function $F : \mathbb{R}_{\geq 0} \to [0, 1]$, where $F(x)$ is the probability that X takes a value less or equal to $x \in \mathbb{R}_{\geq 0}$. In particular, we consider the following principal CDFs F in this article: To model timeouts at some concrete value d, we use a CDF F of a Dirac distribution such that $F(\tau) = 1$ for all $\tau \geq d$ and $F(\tau) = 0$ for all $\tau < d$. Similarly, when we select a random delay that is uniformly distributed in the interval $[0.01, d]$, then for all $\tau < 0.01$ $F(\tau) = 0$ and for all $\tau \geq 0.01$ $F(\tau) = \min\{1, (\tau - 0.01)/(d - 0.01)\}$. For Erlang distributions, we have $F(\tau) = 0$ for all $\tau \leq 0$ and $F(\tau) = 1 - e^{-\tau d} \cdot \sum_{n=0}^{k-1} (\tau d)^n / n!$ for all $\tau > 0$ and a fixed constant $k \in \mathbb{N}_{>0}$. Finally, we consider Weibull distributions, i.e., where $F(\tau) = 0$ for all $\tau \leq 0$ and $F(\tau) = 1 - e^{-(\tau d)^k}$ for all $\tau > 0$ and a fixed constant $k \in \mathbb{N}_{>0}$.[1] A *probability matrix* over some finite set E is a function $M : E \times E \to \mathbb{R}_{\geq 0}$ where $M(e, \cdot) \in \mathcal{D}(E)$ for all $e \in E$.

2.1 Continuous-time Markov Chains with Alarms

A *continuous-time Markov chain (CTMC)* is a triple $C = (S, \lambda, P)$, where S is a finite set of states, $\lambda \in \mathbb{R}_{>0}$ is a common exit rate,[2] and P is a probability matrix over S. Transitions in C are exponentially distributed over the time, i.e., the probability of moving from a state s to state s' within time τ is $P(s, s') \cdot (1 - e^{-\lambda \cdot \tau})$. We extend CTMCs by generally distributed events called *alarms*. A *CTMC with alarms (ACTMC)* over a finite set of alarms A is a tuple

$$\mathcal{A} = (\, S, \lambda, P, A, \langle S_a \rangle, \langle P_a \rangle, \langle F_a \rangle \,),$$

[1] Note that a Weibull and Erlang distributions with $k = 1$ are exponential distributions.

[2] We can assume without restrictions that the parameter λ is the same for all states of S, since every CTMC can be effectively transformed into an equivalent CTMC satisfying this property by the standard uniformization method (see, e.g., [34]).

where (S, λ, P) is a CTMC and $\langle S_a \rangle, \langle P_a \rangle$, and $\langle F_a \rangle$ are tuples defined as follows: $\langle S_a \rangle = (S_a)_{a \in A}$ where S_a is a nonempty set of states where an alarm $a \in A$ is enabled; $\langle P_a \rangle = (P_a)_{a \in A}$ where P_a is a probability matrix of some alarm $a \in A$ for which $P_a(s, s) = 1$ if $s \in S \setminus S_a$; and $\langle F_a \rangle = (F_a)_{a \in A}$ where $F_a : \mathbb{R}_{\geq 0} \to [0, 1]$ is a CDF according to which the ringing time of an alarm $a \in A$ is distributed. We require that $F(0) = 0$ and that the associated distribution has finite mean. Note that then a positive ringing time is chosen almost surely. We require $S_a \cap S_{a'} = \emptyset$ for $a \neq a'$, i.e., in each state at most one alarm is enabled. The set of states where some alarm is enabled is denoted by S_{on}, and we also use S_{off} to denote the set $S \setminus S_{\text{on}}$. The pairs $(s, s') \in S \times S$ with $P(s, s') > 0$ and $P_a(s, s') > 0$ are referred to as *delay transitions* and *a-alarm transitions*, respectively.

Operational behavior. Since in every state only one alarm is enabled, the semantics of an ACTMC can be seen as an infinite CTMC amended with a timer that runs backwards and is set whenever a new alarm is set or the alarm gets disabled. A *run* of the ACTMC $\mathcal{A}$ is an infinite sequence $(s_0, \eta_0), t_0, (s_1, \eta_1), t_1, \ldots$, where η_i is the current value of the timer, t_i is the time spent in s_i, and (s_i, η_i) is called *configuration*. In each configuration (s_i, η_i), we require $\eta_i = \infty$ iff $s_i \in S_{\text{off}}$. Hence, if $s_0 \in S_a$ for some $a \in A$, then the value of η_0 is selected randomly according to F_a. In a current configuration (s_i, η_i), a random delay t is chosen according to the exponential distribution with rate λ. Then, the time t_i and the next configuration (s_{i+1}, η_{i+1}) are determined as follows:

- **(alarm rings)** If $s_i \in S_a$ and $\eta_i \leq t$, then $t_i \stackrel{\text{def}}{=} \eta_i$ and s_{i+1} is selected randomly according to $P_a(s_i, \cdot)$. The value of η_{i+1} is either set to ∞ or selected randomly according to F_b for some $b \in A$, depending on whether the chosen s_{i+1} belongs to S_{off} or S_b, respectively (note that it may happen that $b = a$).
- **(leave before alarm rings)** If $t < \eta_i$, then $t_i \stackrel{\text{def}}{=} t$ and s_{i+1} is selected randomly according to $P(s_i, \cdot)$. Clearly, if $s_{i+1} \in S_{\text{off}}$, then $\eta_{i+1} \stackrel{\text{def}}{=} \infty$. Further, if $s_{i+1} \in S_b$ and $s_i \notin S_b$ for some $b \in A$, then η_{i+1} is selected randomly according to F_b. Otherwise, $s_i, s_{i+1} \in S_b$ for some $b \in A$ and $\eta_{i+1} \stackrel{\text{def}}{=} \eta_i - t$ (note that here $\eta_i > t$).

Similar to standard CTMCs, we define a probability space over all runs initiated in a given $s_0 \in S$. We say that $\mathcal{A}$ is *strongly connected* if its underlying graph is, i.e., for all $s, s' \in S$, where $s \neq s'$, there is a finite sequence $s_0, \ldots, s_n$ of states such that $s = s_0$, $s' = s_n$, and $P(s_i, s_{i+1}) > 0$ or $P_a(s_i, s_{i+1}) > 0$ (for some $a \in A$) for all $0 \leq i < n$.

Note that the timer is set to a new value in a state s only if $s \in S_a$ for some $a \in A$, and the previous state either does not belong to S_a or the transition used to enter s was an alarm transition.[3] Formally, we say that $s \in S_a$ is an *a-setting state* if there exists $s' \in S$ such that either $P_b(s', s) > 0$ for some $b \in A$ (here, we do *not* require $a \neq b$), or $s' \notin S_a$ and $P(s', s) > 0$. The set of all alarm-setting states is denoted by S_{set}. If $S_{\text{set}} \cap S_a$ is a singleton for each $a \in A$, then we say that the alarms in $\mathcal{A}$ are *localized*.

Cost structures and mean-payoff for ACTMCs. We use the standard cost structures that assign non-negative cost values to both states and transitions (see, e.g., [35]). More precisely, we consider the following cost functions: $\mathcal{R} : S \to \mathbb{R}_{\geq 0}$, which assigns a cost rate $\mathcal{R}(s)$ to every state s such that the cost $\mathcal{R}(s)$ is paid for every time unit spent in s, and functions $\mathcal{I}, \mathcal{I}_A : S \times S \to \mathbb{R}_{\geq 0}$ that assign to each delay transition and each alarm-setting transition, respectively, an instant execution cost. For every run $\omega = (s_0, \eta_0), t_0, (s_1, \eta_1), t_1, \ldots$ of $\mathcal{N}$, we define the associated *mean-payoff* by

$$\text{MP}(\omega) \quad = \quad \limsup_{n \to \infty} \frac{\sum_{i=0}^{n} (\mathcal{R}(s_i) \cdot t_i + \mathcal{I}(s_i, s_{i+1}))}{\sum_{i=0}^{n} t_i}.$$

[3] In fact, another possibility (which does not require any special attention) is that s is the initial state of a run.

Here, $\mathcal{J}(s_i, s_{i+1})$ is either $\mathcal{I}(s_i, s_{i+1})$ or $\mathcal{I}_A(s_i, s_{i+1})$, depending on whether $t_i < \eta_i$ or not, respectively. We use $\mathbb{E}[\mathrm{MP}]$ to denote the expectation of MP. In general, MP may take more than one value with positive probability. However, if the graph of the underlying ACTMC is strongly connected, then almost all runs yield the same mean-payoff value independent of the initial state [13].

2.2 Parametric ACTMCs

In ACTMCs the distribution functions for the alarms are already fixed. We now vary one parameter d of the alarm distributions (cf. beginning of Section 2) and ask what parameter values minimize the expected long-run average costs. [4] A *parametric ACTMC* is defined similarly as an ACTMC, but instead of the concrete distribution function F_a, we specify a *parameterized* distribution function $F_a[x]$ together with an interval $[\ell_a, u_a]$ of eligible parameter values for every $a \in A$. We use $F_a[d]$ to denote the distribution obtained by instantiating the parameter x with some $d \in [\ell_a, u_a]$. Formally, a parametric ACTMC is a tuple,

$$\mathcal{N} = (\, S, \lambda, P, A, \langle S_a \rangle, \langle P_a \rangle, \langle F_a[x] \rangle, \langle \ell_a \rangle, \langle u_a \rangle \,),$$

where all components are defined in the same way as for ACTMC except for the tuples $\langle F_a[x] \rangle$, $\langle \ell_a \rangle$, and $\langle u_a \rangle$ of all $F_a[x]$, ℓ_a, and u_a discussed above. Strong connectedness, localized alarms, and cost structures are defined as for (non-parametric) ACTMCs.

A *parameter function* for $\mathcal{N}$ is a function $\mathbf{d} : A \to \mathbb{R}$ such that $\mathbf{d}(a) \in [\ell_a, u_a]$ for every $a \in A$. We only consider parametric ACTMCs where each parameter function $\mathbf{d}$ yields an ACTMC given by replacing each $F_a[x]$ in $\mathcal{N}$ with the distribution function $F_a[\mathbf{d}(a)]$. We denote the arising ACTMC by $\mathcal{N}^{\mathbf{d}}$. When cost structures are defined on $\mathcal{N}$, we use $\mathbb{E}^{\mathbf{d}}[\mathrm{MP}]$ to denote the expected mean-payoff in $\mathcal{N}^{\mathbf{d}}$. For a given $\varepsilon > 0$, we say that a parameter function $\mathbf{d}$ is *ε-optimal* if

$$\mathbb{E}^{\mathbf{d}}[\mathrm{MP}] \leq \inf_{\mathbf{d}'} \; \mathbb{E}^{\mathbf{d}'}[\mathrm{MP}] + \varepsilon,$$

where $\mathbf{d}'$ ranges over all parameter functions for $\mathcal{N}$.

2.3 Semi-Markov Decision Processes

A *semi-Markov decision process (semi-MDP)* is a tuple $\mathcal{M} = (M, Act, Q, t, c)$, where M is a finite set of states, $Act = \biguplus_{m \in M} Act_m$ is a set of actions where $Act_m \neq \emptyset$ is a subset of actions enabled in a state m, $Q : Act \to \mathcal{D}(M)$ is a function assigning the probability $Q(b)(m')$ to move from $m \in M$ to $m' \in M$ executing $b \in Act_m$, and functions $t, c : Act \to \mathbb{R}_{\geq 0}$ provide the expected time and costs when executing an action, respectively.[5] A *run* in $\mathcal{M}$ is an infinite sequence $\omega = m_0, b_0, m_1, b_1, \ldots$ where $b_i \in Act_{m_i}$ for every $i \in \mathbb{N}_{\geq 0}$. The mean-payoff of ω is

$$\mathrm{MP}(\omega) = \limsup_{n \to \infty} \frac{\sum_{i=0}^{n} c(b_i)}{\sum_{i=0}^{n} t(b_i)}.$$

A (stationary and deterministic) *policy* for $\mathcal{M}$ is a function $\sigma : M \to Act$ such that $\sigma(m) \in Act_m$ for all $m \in M$. *Applying* σ to $\mathcal{M}$ yields the standard probability measure $\Pr^{\sigma}_{\mathcal{M}}$ over all runs initiated in a given initial state. We say that $\mathcal{M}$ is *strongly connected* if for each policy σ and all $m, m' \in M$, where $m \neq m'$, there is a finite sequence $m_0, \ldots, m_n$ of states such that $m = m_0$, $m' = m_n$, and $Q(\sigma(m_i))(m_{i+1}) > 0$ for all $i \in \mathbb{N}_{\geq 0}$. The expected mean-payoff achieved by σ is

[4] In our current setting, distribution functions with several parameters can be accommodated by choosing the parameter to optimize and fixing the others. In some cases, we can also use simple extensions to synthesize, e.g., both d_1 and d_2 for the uniform distribution in $[d_1, d_2]$.

[5] For our purposes, the actual distribution of the time and costs spent before executing some action is irrelevant, only their expectations matter, see Section 11.4 in [35].

denoted by $\mathbb{E}^{\sigma}_{\mathcal{M}}[\mathrm{MP}]$. An *optimal*[6] policy achieving the *minimal* expected mean-payoff is guaranteed to exist, and it is computable by a simple *policy iteration algorithm* for strongly connected semi-MDPs (see, e.g., [35]).

κ-Approximations of semi-MDPs. Let $\mathcal{M} = (M, Act, Q, t, c)$ be a semi-MDP and $\kappa \in \mathbb{Q}_{>0}$. We say that $Q^{\kappa} : Act \to \mathcal{D}(M)$ and $t^{\kappa}, c^{\kappa} : Act \to \mathbb{R}_{\geq 0}$ are *κ-approximations* of Q, t, c, respectively, if for all $m, m' \in M$ and $b \in Act_m$ it holds that $Q(b)$ and $Q^{\kappa}(b)$ have the same support, $|Q(b)(m') - Q^{\kappa}(b)(m')| \leq \kappa$, $|t(b) - t^{\kappa}(b)| \leq \kappa$, and $|c(b) - c^{\kappa}(b)| \leq \kappa$. A *$\kappa$-approximation* of $\mathcal{M}$ is a semi-MDP $(M, Act, Q^{\kappa}, t^{\kappa}, c^{\kappa})$ where Q^{κ}, t^{κ}, c^{κ} are κ-approximations of Q, t, c, respectively. We denote by $[\mathcal{M}]_{\kappa}$ the set of all κ-approximations of $\mathcal{M}$.

3 SYNTHESIZING ε-OPTIMAL PARAMETER FUNCTIONS

In the following, we fix a strongly connected parametric ACTMC,

$$\mathcal{N} = (\, S, \lambda, P, A, \langle S_a \rangle, \langle P_a \rangle, \langle F_a[x] \rangle, \langle \ell_a \rangle, \langle u_a \rangle\,),$$

with cost functions $\mathcal{R}$, $\mathcal{I}$, and $\mathcal{I}_A$, and aim toward an algorithm synthesizing an ε-optimal parameter function for $\mathcal{N}$. Here, ε-optimality is understood with respect to the expected mean-payoff. That is, we deal with the following computational problem:

ε-optimal parameter synthesis for parametric ACTMCs

Input: $\varepsilon \in \mathbb{Q}_{>0}$ and a strongly connected parametric ACTMC $\mathcal{N}$ with rational[7] transition probabilities, rate λ, bounds $\langle \ell_a \rangle$, $\langle u_a \rangle$, and cost functions $\mathcal{R}$, $\mathcal{I}$, and $\mathcal{I}_A$.
Output: An ε-optimal parameter function $\mathbf{d}$.

Our approach to solve the above problem for parametric ACTMCs with localized alarms is based on a reduction to the problem of synthesizing expected mean-payoff optimal policies in semi-MDPs. For non-localized parametric ACTMCs, the approach is similar but we use a more general formalism than semi-MDPs. For presentation reasons, we first focus on the localized case, and hence in the following we assume $\mathcal{N}$ to be with localized alarms.

3.1 The Set of Semi-Markov Decision Processes $[\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_{\kappa}$

Let $a \in A$, and let $s \in S_a \cap S_{\mathrm{set}}$. Recall that we assume $\mathcal{N}$ to be localized and thus, s is the uniquely defined a-setting state. Then, for every $d \in [\ell_a, u_a]$ consider runs initiated in a configuration (s, η) where η is chosen randomly according to $F_a[d]$. Almost all such runs eventually visit a *regenerative* configuration (s', η') where either $s' \in S_{\mathrm{off}}$ or η' is chosen randomly in $s' \in S_{\mathrm{set}}$, i.e., either all alarms are disabled or one is newly set. We use $\Pi_s(d)$ to denote the associated probability distribution over $S_{\mathrm{set}} \cup S_{\mathrm{off}}$, i.e., $\Pi_s(d)(s')$ is the probability of visiting a regenerative configuration of the form (s', η') from s without previously visiting another regenerative configuration. Note that $\Pi_s(d)(\cdot)$ has the same support for all $d \in [\ell_a, u_a]$. Further, we use $\mathfrak{C}_s(d)$ and $\Theta_s(d)$ to denote the expected accumulated costs and the expected time elapsed until visiting a regenerative configuration, respectively. We use the same notation also for $s \in S_{\mathrm{off}}$, where $\Pi_s(d)(\cdot) = P(s, \cdot)$, $\Theta_s(d) = 1/\lambda$, and $\mathfrak{C}_s(d) = \mathcal{R}(s)/\lambda + P(s, \cdot) \cdot \mathcal{I}_P$. Note that here the functions are constant, i.e., they are independent of d. The semi-MDP $\mathcal{M}_{\mathcal{N}} = (S_{\mathrm{set}} \cup S_{\mathrm{off}}, Act, Q, t, c)$ is defined over actions

$$Act = \{\langle\!\langle s, d \rangle\!\rangle : d \in [\ell_a, u_a], s \in S_{\mathrm{set}} \cap S_a, a \in A\} \cup \{\langle\!\langle s, 0 \rangle\!\rangle : s \in S_{\mathrm{off}}\},$$

[6] This policy is optimal not only among stationary and deterministic policies, but even among more general (randomized and history-dependent) policies.
[7] The additional restrictions imply finitely representable input.

where for all $\langle\!\langle s, d \rangle\!\rangle \in Act$, we have $Q(\langle\!\langle s, d \rangle\!\rangle) = \Pi_s(d)$, $t(\langle\!\langle s, d \rangle\!\rangle) = \Theta_s(d)$ and $c(\langle\!\langle s, d \rangle\!\rangle) = \in\!\!\!\!\in_s(d)$. Note that the action space of $\mathcal{M}_{\mathcal{N}}$ is dense and that $\Pi_s(d)$, $\Theta_s(d)$, and $\in\!\!\!\!\in_s(d)$ might be irrational. For our algorithms, we have to ensure a finite action space as well as rational probability and expectation values. We thus define the δ-discretization of $\mathcal{M}_{\mathcal{N}}$ as $\mathcal{M}_{\mathcal{N}}\langle\delta\rangle = (S_{\text{set}} \cup S_{\text{off}}, Act^{\delta}, Q^{\delta}, t^{\delta}, c^{\delta})$ for a given discretization function $\delta : A \to \mathbb{Q}_{>0}$ as follows: $\mathcal{M}_{\mathcal{N}}\langle\delta\rangle$ is defined as $\mathcal{M}_{\mathcal{N}}$ above, but over the action space $Act^{\delta} = \bigcup_{s \in S_{\text{set}} \cup S_{\text{off}}} Act^{\delta}_s$ with

$$Act^{\delta}_s = \{\langle\!\langle s, d \rangle\!\rangle : d = \ell_a + i \cdot \delta(a) < u_a, i \in \mathbb{N}_{\geq 0}\} \cup \{\langle\!\langle s, u_a \rangle\!\rangle\},$$

for $s \in S_{\text{set}} \cap S_a$ and $Act^{\delta}_s = \{\langle\!\langle s, 0 \rangle\!\rangle\}$ otherwise.

To ensure rational values of $\Pi_s(d)$, $\Theta_s(d)$, and $\in\!\!\!\!\in_s(d)$, we consider the set of κ-approximations $[\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_{\kappa}$ of $\mathcal{M}_{\mathcal{N}}\langle\delta\rangle$ for any given approximation constant $\kappa \in \mathbb{Q}_{>0}$. Note that as $\mathcal{N}$ is strongly connected, every $\mathcal{M} \in [\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_{\kappa}$ is also strongly connected.

3.2 An Explicit Parameter-synthesis Algorithm

Every policy σ minimizing the expected mean-payoff in $\mathcal{M}_{\mathcal{N}}$ yields an optimal parameter function $\mathbf{d}^{\sigma}$ for $\mathcal{N}$ defined by $\mathbf{d}^{\sigma}(a) = d$ where $\sigma(s) = \langle\!\langle s, d \rangle\!\rangle$ for the unique a-setting state s. A naïve approach toward an ε-optimal parameter function minimizing the expected mean-payoff in $\mathcal{N}$ is to compute sufficiently small discretization function δ and approximation constant κ such that synthesizing an optimal policy in any $\mathcal{M} \in [\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_{\kappa}$ yields an ε-optimal parameter function for $\mathcal{N}$. As $\mathcal{M}$ is finite and contains only rational probability and expectation values, the synthesis of an optimal policy for $\mathcal{M}$ can then be carried out using standard algorithms for semi-MDPs (see, e.g., [35]). This approach is applicable under the following mild assumptions:

(1) For every $\varepsilon \in \mathbb{Q}_{>0}$, there are computable $\delta : A \to \mathbb{Q}_{>0}$ and $\kappa \in \mathbb{Q}_{>0}$ such that for every $\mathcal{M} \in [\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_{\kappa}$ and every optimal policy σ for $\mathcal{M}$, the associated parameter function $\mathbf{d}^{\sigma}$ is ε-optimal for $\mathcal{N}$.
(2) For all $\kappa \in \mathbb{Q}_{>0}$ and $s \in S_{\text{set}}$, there are computable rational κ-approximations Π^{κ}_s, Θ^{κ}_s, and $\in\!\!\!\!\in^{\kappa}_s$ of Π_s, Θ_s, and $\in\!\!\!\!\in_s$, respectively.

Assumption 1 usually follows from perturbation bounds on the expected mean-payoff using a straightforward error-propagation analysis. Assumption 2 can be obtained, e.g., by first computing $\kappa/2$-approximations of Π_s, Θ_s, and $\in\!\!\!\!\in_s$ for $s \in S_{\text{set}} \cap S_a$, considering a as alarm to which a Dirac distribution is assigned to, and then integrating the obtained functions over the probability measure determined by $F_a[x]$ to get the resulting κ-approximation (see also [9, 13]). Hence, Assumptions 1 and 2 rule out only those types of distributions that are rarely used in practice. In particular, the assumptions are satisfied for uniform, Dirac, Erlang, and Weibull distributions, as we will show in Section 4. Note that Assumption 2 implies that for all $\delta : A \to \mathbb{Q}_{>0}$ and $\kappa \in \mathbb{Q}_{>0}$, there is a computable semi-MDP $\mathcal{M}$ of $[\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_{\kappa}$.

3.3 A Symbolic Parameter-synthesis Algorithm

Usually, the naïve *explicit approach* to parameter synthesis explained above is computationally infeasible due a large number of actions in $\mathcal{M} \in [\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_{\kappa}$. We now present a symbolic parameter-synthesis algorithm that computes the set of states of some $\mathcal{M}$ (see Assumption 1) but avoids computing the set of all actions of $\mathcal{M}$ and their effects. The algorithm is obtained by modifying the standard policy iteration [35] for semi-MDPs.

Standard policy iteration. When applied to $\mathcal{M}$, standard policy iteration starts by picking an arbitrary policy σ, which is then repeatedly improved until a fixed point is reached. In each iteration, the current policy σ is first evaluated by computing the associated *gain* $g \in \mathbb{Q}$ and

bias $\mathbf{h} : S \to \mathbb{Q}$.[8] Then, for each state $s \in S_{\text{set}}$, every outgoing action $\langle\!\langle s, d \rangle\!\rangle$ is ranked by the function

$$\mathcal{R}_s^\kappa[g, \mathbf{h}](d) \stackrel{\text{def}}{=} \in\!\!\!\!\!\!\in_s^\kappa(d) - g \cdot \Theta_s^\kappa(d) + \Pi_s^\kappa(d) \cdot \mathbf{h}, \qquad (\times)$$

where $\in\!\!\!\!\!\!\in_s^\kappa$, Θ_s^κ, and Π_s^κ are the determining functions of $\mathcal{M}$. If the action chosen by σ at s does not have the best (minimal) rank, then it is improved by redefining $\sigma(s)$ to some best-ranked action. The new policy is then evaluated by computing its gain and bias and possibly improved again. The standard algorithm terminates when for all states the current policy cannot be improved.

Analytical κ-approximations. In many cases $\Pi_s(d)$, $\Theta_s(d)$, and $\in\!\!\!\!\!\!\in_s(d)$ for $s \in S_{\text{set}}$ are expressible as infinite sums where the summands comprise elementary functions such as polynomials or $\exp(\cdot)$. Given $\kappa \in \mathbb{Q}_{>0}$ one may effectively truncate these infinite sums into finitely many initial summands such that the obtained expressions are differentiable in the interval $[\ell_a, u_a]$ and yield *analytical κ-approximations* $\mathbf{\Pi}_s^\kappa(d)$, $\mathbf{\Theta}_s^\kappa(d)$, and $\mathbf{\in\!\!\!\!\!\!\in}_s^\kappa(d)$, respectively. Note that we use bold type for analytical κ-approximations, complementing the non-bold type used for rational κ-approximations.

Now, we can introduce the *analytical* version of the ranking function, denoted by $\boldsymbol{\mathcal{R}}_s^\kappa[g, \mathbf{h}](d)$, obtained by replacing the rational κ-approximations in the right-hand side of Equation $(\times)$ with their analytical counterparts:

$$\boldsymbol{\mathcal{R}}_s^\kappa[g, \mathbf{h}](d) \stackrel{\text{def}}{=} \mathbf{\in\!\!\!\!\!\!\in}_s^\kappa(d) - g \cdot \mathbf{\Theta}_s^\kappa(d) + \mathbf{\Pi}_s^\kappa(d) \cdot \mathbf{h}. \qquad (\star)$$

Note that $\mathcal{R}_s^\kappa[g, \mathbf{h}](d)$ given in Equation $(\times)$ and $\boldsymbol{\mathcal{R}}_s^\kappa[g, \mathbf{h}](d)$ given above in Equation $(\star)$ are *different* functions with different properties. The function $\boldsymbol{\mathcal{R}}_s^\kappa[g, \mathbf{h}](d)$ is differentiable for $d \in [\ell_a, u_a]$ when g and $\mathbf{h}$ are constant. Observe that the discretized parameters minimizing $\mathcal{R}_s^\kappa[g, \mathbf{h}](d)$ are either close to ℓ_a, u_a, or roots of the derivative of $\boldsymbol{\mathcal{R}}_s^\kappa[g, \mathbf{h}](d)$. Using the isolated roots and bounds ℓ_a and u_a, we identify a usually small set of candidate actions and explicitly evaluate only those instead of all actions. As $\mathbf{\Pi}_s^\kappa(d)$, $\mathbf{\Theta}_s^\kappa(d)$, and $\mathbf{\in\!\!\!\!\!\!\in}_s^\kappa(d)$ may be *irrational* for rational arguments, they might not be computable even for the discretized parameter values. However, when Assumption 2 is fulfilled, it is safe to use *rational* κ-approximations $\Pi_s^\kappa(d)$, $\Theta_s^\kappa(d)$, and $\in\!\!\!\!\!\!\in_s^\kappa(d)$ for the evaluation step. The analytical κ-approximations $\mathbf{\Pi}_s^\kappa(d)$, $\mathbf{\Theta}_s^\kappa(d)$, and $\mathbf{\in\!\!\!\!\!\!\in}_s^\kappa(d)$ are hence used to *identify* a subset of candidate actions that may provide some improvement, while the rational and computable κ-approximations $\Pi_s^\kappa(d)$, $\Theta_s^\kappa(d)$, and $\in\!\!\!\!\!\!\in_s^\kappa(d)$ are used to *evaluate* the candidate actions.

Before we provide our symbolic algorithm, we formally state the additional assumptions required to guarantee its soundness:

(3) For all $a \in A$, $s \in S_{\text{set}} \cap S_a$, $\delta : A \to \mathbb{Q}_{>0}$ and $\kappa \in \mathbb{Q}_{>0}$, there are analytical κ-approximations $\mathbf{\Pi}_s^\kappa$, $\mathbf{\Theta}_s^\kappa$, and $\mathbf{\in\!\!\!\!\!\!\in}_s^\kappa$ of Π_s, Θ_s, and $\in\!\!\!\!\!\!\in_s$, respectively, such that the function $\boldsymbol{\mathcal{R}}_s^\kappa[g, \mathbf{h}](d)$ from Equation $(\star)$, where $g \in \mathbb{Q}$ and $\mathbf{h} : S_{\text{set}} \cup S_{\text{off}} \to \mathbb{Q}$ are constant, is differentiable for $d \in [\ell_a, u_a]$. Further, there is an algorithm to approximate the roots of the derivative of $\boldsymbol{\mathcal{R}}_s^\kappa[g, \mathbf{h}](d)$ in the interval $[\ell_a, u_a]$ up to the absolute error $\delta(a)/2$.

(4) For each $s \in S_{\text{set}}$ in which alarm $a \in A$ is set there is a computable constant $\Pi_s^{\min} \in \mathbb{Q}_{>0}$ such that for all $d \in [\ell_a, u_a]$ and $s' \in S_{\text{set}} \cup S_{\text{off}}$, we have that $\Pi_s(d)(s') > 0$ implies $\Pi_s(d)(s') \geq \Pi_s^{\min}$.

Note that compared to Assumption 2, the κ-approximations of Assumption 3 are harder to construct: We require closed forms for $\mathbf{\Pi}_s^\kappa$, $\mathbf{\Theta}_s^\kappa$, and $\mathbf{\in\!\!\!\!\!\!\in}_s^\kappa$ making the symbolic derivative of $\boldsymbol{\mathcal{R}}_s^\kappa[g, \mathbf{h}](d)$ computable and suitable for effective root approximation.

[8] Here, it suffices to know that g is a scalar and $\mathbf{h}$ is a vector assigning numbers to states; for more details, see Sections 8.2.1 and 8.6.1 in [35].

ALGORITHM 1: Symbolic policy iteration

input: A strongly connected parametric ACTMC $\mathcal{N}$ with localized alarms, rational-valued cost functions $\mathcal{R}$, $\mathcal{I}$, $\mathcal{I}_A$, and $\varepsilon \in \mathbb{Q}_{>0}$ such that Assumptions 1–4 are fulfilled.
output: An ε-optimal parameter function $\mathbf{d}$.

1 compute the sets S_{set} and S_{off}
2 compute the constant κ, function δ, and for all $s \in S_{\text{set}}$ the constant $\Pi_s^{\min}$ of Assumptions 1 and 4
3 let $\xi = \min\{\kappa/2, \Pi_s^{\min}/3 : s \in S_{\text{set}}\}$
4 fix the functions $\Pi_s^{\xi}, \Theta_s^{\xi}$, and $\in_s^{\xi}$ of Assumption 2 determining $\mathcal{M}_\xi \in [\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_\xi$
5 choose an arbitrary state $s' \in S_{\text{set}} \cup S_{\text{off}}$ and a policy σ' for $\mathcal{M}_\xi$
6 **repeat**
7 $\quad \sigma := \sigma'$
$\quad$ `// policy evaluation`
8 $\quad$ compute the *gain*, i.e., the scalar $g := \mathbb{E}^{\sigma}[\text{MP}]$
9 $\quad$ compute the *bias*, i.e., the vector $\mathbf{h}\colon S \to \mathbb{Q}$ satisfying $\mathbf{h}(s') = 0$ and for each $s \in S_{\text{set}} \cup S_{\text{off}}$, $\mathbf{h}(s) = \in_s^{\xi}(d) - g \cdot \Theta_s^{\xi}(d) + \Pi_s^{\xi}(d) \cdot \mathbf{h}$, where $\sigma(s) = \langle\!\langle s, d \rangle\!\rangle$
10 $\quad$ **foreach** $a \in A$ *and* $s \in S_{\text{set}} \cap S_a$ **do**
$\quad\quad$ `// policy improvement`
11 $\quad\quad$ compute the set R of $\delta(a)/2$-approximations of the roots of the derivative of $\boldsymbol{\mathcal{R}}_s^{\xi}[g, \mathbf{h}](d)$ in $[\ell_a, u_a]$ using Assumption 3
12 $\quad\quad$ $C := \left\{ \sigma(s) \right\} \cup \left\{ \langle\!\langle s, d \rangle\!\rangle \in Act_s^{\delta} : |d - r| \le 3 \cdot \delta(a)/2, \text{for } r \in R \cup \{\ell_a, u_a\} \right\}$
13 $\quad\quad$ $B := \operatorname{argmin}_{\langle\!\langle s,d\rangle\!\rangle \in C} \mathcal{R}_s^{\xi}[g, \mathbf{h}](d)$
14 $\quad\quad$ **if** $\sigma(s) \in B$ **then** $\sigma'(s) := \sigma(s)$
15 $\quad\quad$ **else** pick an $\langle\!\langle s, d \rangle\!\rangle \in B$ and set $\sigma'(s) := \langle\!\langle s, d \rangle\!\rangle$
16 $\quad$ **end**
17 **until** $\sigma = \sigma'$
18 **return** $\mathbf{d}^{\sigma}$

Symbolic policy iteration algorithm. Algorithm 1 closely mimics the standard policy iteration algorithm except for the definition of new precision ξ at line 3 and the policy improvement part. The local extrema points of $\boldsymbol{\mathcal{R}}_s^{\xi}[g, \mathbf{h}](d)$ (cf. Equation ($\star$)) in the interval $[\ell_a, u_a]$ are identified by computing roots of its symbolic derivative (line 11). Then, we construct a small set C of *candidate actions* that are close to these roots or the bounds ℓ_a, u_a (line 12). Each given candidate action is then evaluated using the function $\mathcal{R}_s^{\xi}[g, \mathbf{h}](d) = \in_s^{\xi}(d) - g \cdot \Theta_s^{\xi}(d) + \Pi_s^{\xi}(d) \cdot \mathbf{h}$ (cf. Equation ($\times$)). An improving candidate action is chosen based on the computed values (lines 14 and 15).

Theorem 3.1 (Correctness of Algorithm 1). *The symbolic policy iteration algorithm solves the ε-optimal parameter synthesis problem for localized parametric ACTMCs and cost functions that fulfill Assumptions 1–4.*

Proof. First, observe that in Algorithm 1 the actions of the currently improved policy are maintained in the set of candidate actions (line 12). From the properties of the policy iteration [35], it is hence guaranteed that if the policy is changed, then the gain and bias strictly (lexicographically) improve. Since the number of policies in $\mathcal{M}_\xi$ is finite, Algorithm 1 terminates. A challenging point is that we compute only approximate minima of the function $\boldsymbol{\mathcal{R}}_s^{\xi}[g, \mathbf{h}](d)$, which might be *different*

from the function $\mathcal{R}_s^{\xi}[g,\mathbf{h}](d)$ used to evaluate the candidate actions. There may exist an action that is not in the candidate set C but has minimal $\mathcal{R}_s^{\xi}[g,\mathbf{h}](d)$. A priori it is hence not clear whether the policy computed by Algorithm 1 is optimal for $\mathcal{M}_{\xi}$. Fortunately, due to Assumption 1, we know that if the computed policy is optimal for *some* $\mathcal{M}' \in [\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_{\kappa}$, then it induces ε-optimal parameters for $\mathcal{N}$. Therefore, it is sufficient to construct some Π'_s, Θ'_s, and $\in'_s$ for each $s \in S_{\text{set}}$ determining such a candidate $\mathcal{M}'$.

Let $s \in S_{\text{set}}$ be a state and $a \in A$ be the uniquely defined alarm such that $s \in S_a$. Furthermore, we consider the values and functions of the last iteration of the repeat-until loop of Algorithm 1 when applied to the given parametric ACTMC $\mathcal{N}$. That is, we use the gain g, bias $\mathbf{h}$, current policy σ, approximations Π_s^{ξ}, $\in_s^{\xi}$, Θ_s^{ξ}, $\mathbf{\Pi}_s^{\xi}$, $\mathbf{\in}_s^{\xi}$, $\mathbf{\Theta}_s^{\xi}$, from the last iteration of the repeat-until loop and candidate sets C and B in the inner foreach iteration concerning s. Let policy σ select an action $\sigma(s) = \langle\!\langle s, d\rangle\!\rangle$ in s, i.e., d is the alarm parameter of a in s, and consider the sets of candidate alarm parameters $D \overset{\text{def}}{=} \left\{d'' : \langle\!\langle s, d''\rangle\!\rangle \in Act_s^{\delta}\right\}$ and $C' \overset{\text{def}}{=} \{d'' : \langle\!\langle s, d''\rangle\!\rangle \in C\}$.

For each state $s \in S_{\text{set}}$, we need to define κ-approximations Π'_s, Θ'_s, and $\in'_s$ of Π_s, Θ_s, and $\in_s$ on D, respectively, and such that for $\mathcal{R}'_s[g,\mathbf{h}](\cdot) \overset{\text{def}}{=} \in'_s(\cdot) - g \cdot \Theta'_s(\cdot) + \Pi'_s(\cdot) \cdot \mathbf{h}$ and every $d' \in D \setminus \{d\}$,

$$\mathcal{R}'_s[g,\mathbf{h}](d) \quad \leq \quad \mathcal{R}'_s[g,\mathbf{h}](d').$$

First, we will construct Π'_s, Θ'_s, $\in'_s$ and prove the theorem assuming that we have in hand certain "shifted" κ-approximations $\mathbf{\Pi}'_s$, $\mathbf{\in}'_s$, $\mathbf{\Theta}'_s$ that have some good properties. Then, we will construct the "shifted" κ-approximations $\mathbf{\Pi}'_s$, $\mathbf{\in}'_s$, $\mathbf{\Theta}'_s$ and show that they have the desired properties.

Assume we have in hand "shifted" κ-approximations $\mathbf{\Pi}'_s$, $\mathbf{\in}'_s$, $\mathbf{\Theta}'_s$ (of Π_s, $\in_s$, Θ_s on $[\ell_a, u_a]$) and $\boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}](d') \overset{\text{def}}{=} \mathbf{\in}'_s(d') - g \cdot \mathbf{\Theta}'_s(d') + \mathbf{\Pi}'_s(d') \cdot \mathbf{h}$ for each $d' \in [\ell_a, u_a]$ such that

$$\exists c \geq 0 : \forall d' \in [\ell_a, u_a] : \boldsymbol{\mathcal{R}}_s^{\xi}[g,\mathbf{h}](d') + c = \boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}](d') \tag{1}$$

and

$$\forall d' \in C' : \mathcal{R}_s^{\xi}[g,\mathbf{h}](d') \leq \boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}](d'). \tag{2}$$

Now, for each $d' \in D$, we set

- $\Pi'_s(d') \overset{\text{def}}{=} \Pi_s^{\xi}(d')$, $\in'_s(d') \overset{\text{def}}{=} \in_s^{\xi}(d')$, $\Theta'_s(d') \overset{\text{def}}{=} \Theta_s^{\xi}(d')$ if $d' = d$ and
- $\Pi'_s(d') \overset{\text{def}}{=} \mathbf{\Pi}'_s(d')$, $\in'_s(d') \overset{\text{def}}{=} \mathbf{\in}'_s(d')$, $\Theta'_s(d') \overset{\text{def}}{=} \mathbf{\Theta}'_s(d')$ if $d' \neq d$.

From Assumption 3 and the definition of $\boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}]$ it follows that $\boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}]$ has the same arguments of local extrema as $\boldsymbol{\mathcal{R}}_s^{\xi}[g,\mathbf{h}]$ on $[\ell_a, u_a]$. Further, the definition of the candidate set C (line 12 of Algorithm 1) and C' yields that $C' \cap \operatorname{argmin}_{d' \in D} \boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}](d')$ is nonempty. We thus can pick an arbitrary $\underline{d} \in C' \cap \operatorname{argmin}_{d' \in D} \boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}](d')$ and then for each $d' \in D \setminus \{d\}$, we have

$$\mathcal{R}'_s[g,\mathbf{h}](d) = \mathcal{R}_s^{\xi}[g,\mathbf{h}](d) \leq \boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}](\underline{d}) \leq \boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}](d') = \mathcal{R}'_s[g,\mathbf{h}](d').$$

The left inequality follows from the definition of $\boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}]$ (see Equation (2)), the right inequality follows from the definition of $\underline{d}$.

To complete the proof, it remains to define the "shifted" κ-approximations $\mathbf{\Pi}'_s$, $\mathbf{\in}'_s$, $\mathbf{\Theta}'_s$ (of Π_s, $\in_s$, Θ_s on $[\ell_a, u_a]$) such that they satisfy Equations (1) and (2). Let $\overline{d} \in \operatorname{argmax}_{d' \in C'} \mathcal{R}_s^{\xi}[g,\mathbf{h}](d') - \boldsymbol{\mathcal{R}}_s^{\xi}[g,\mathbf{h}](d')$. If $\mathcal{R}_s^{\xi}[g,\mathbf{h}](\overline{d}) - \boldsymbol{\mathcal{R}}_s^{\xi}[g,\mathbf{h}](\overline{d}) \leq 0$, then we set $\Delta\Pi \overset{\text{def}}{=} \mathbf{0}$, $\Delta\Theta \overset{\text{def}}{=} 0$, and $\Delta\in \overset{\text{def}}{=} 0$. Otherwise, we set

- $\Delta\Pi \overset{\text{def}}{=} \Pi_s^{\xi}(\overline{d}) - \mathbf{\Pi}_s^{\xi}(\overline{d})$,
- $\Delta\Theta \overset{\text{def}}{=} \Theta_s^{\xi}(\overline{d}) - \mathbf{\Theta}_s^{\xi}(\overline{d})$, and

- $\Delta\in \stackrel{\text{def}}{=} \in_s^\xi(\overline{d}) - \boldsymbol{\in}_s^\xi(\overline{d})$.

Now, for all $d' \in D$, we put

- $\boldsymbol{\Pi}'_s(d') \stackrel{\text{def}}{=} \boldsymbol{\Pi}_s^\xi(d') + \Delta\Pi$,
- $\boldsymbol{\Theta}'_s(d') \stackrel{\text{def}}{=} \boldsymbol{\Theta}_s^\xi(d') + \Delta\Theta$, and
- $\boldsymbol{\in}'_s(d') \stackrel{\text{def}}{=} \boldsymbol{\in}_s^\xi(d') + \Delta\in$.

We show that Equations (1) and (2) hold: For each $d' \in [\ell_a, u_a]$, we have that

$$\begin{aligned}\boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}](d') &= \boldsymbol{\in}'_s(d') - g \cdot \boldsymbol{\Theta}'_s(d') + \boldsymbol{\Pi}'_s(d') \cdot \mathbf{h}\\ &= (\boldsymbol{\in}_s^\xi(d') + \Delta\in) - g \cdot (\boldsymbol{\Theta}_s^\xi(d') + \Delta\Theta) + (\boldsymbol{\Pi}_s^\xi(d') + \Delta\Pi) \cdot \mathbf{h}\\ &= \boldsymbol{\mathcal{R}}_s^\xi[g,\mathbf{h}](d') + c,\end{aligned}$$

where $c = 0$ if $\mathcal{R}_s^\xi[g,\mathbf{h}](\overline{d}) - \boldsymbol{\mathcal{R}}_s^\xi[g,\mathbf{h}](\overline{d}) \le 0$ and $c = \mathcal{R}_s^\xi[g,\mathbf{h}](\overline{d}) - \boldsymbol{\mathcal{R}}_s^\xi[g,\mathbf{h}](\overline{d}) > 0$ otherwise. Thus, Equation (1) holds. Since $c = \max\{0, \mathcal{R}_s^\xi[g,\mathbf{h}](\overline{d}) - \boldsymbol{\mathcal{R}}_s^\xi[g,\mathbf{h}](\overline{d})\}$ and $\overline{d} \in \operatorname{argmax}_{d'\in C'} \mathcal{R}_s^\xi[g,\mathbf{h}](d') - \boldsymbol{\mathcal{R}}_s^\xi[g,\mathbf{h}](d')$, for each $d' \in C'$ it holds that

$$\mathcal{R}_s^\xi[g,\mathbf{h}](d') \le \boldsymbol{\mathcal{R}}_s^\xi[g,\mathbf{h}](d') + c = \boldsymbol{\mathcal{R}}'_s[g,\mathbf{h}](d'),$$

which implies Equation (2). It remains to show that $\boldsymbol{\Pi}'_s, \boldsymbol{\Theta}'_s, \boldsymbol{\in}'_s$ are κ-approximations of $\Pi_s, \Theta_s, \in_s$ on $[\ell_a, u_a]$: Note, that for each $s' \in S_{\text{set}} \cup S_{\text{off}}$ it holds that $\Delta\Pi(s') \le 2\xi$, $\Delta\Theta \le 2\xi$, and $\Delta\in \le 2\xi$. Since $\xi \le \kappa/2$, $\boldsymbol{\Theta}'_s$ and $\boldsymbol{\in}'_s$ are κ-approximations of Θ_s and $\in_s$ on $[\ell_a, u_a]$, respectively; and for each $d' \in [\ell_a, u_a]$ and $s' \in S_{\text{set}} \cup S_{\text{off}}$ it holds that $|\boldsymbol{\Pi}'_s(d')(s') - \Pi_s(d')(s')| \le 2\xi \le \kappa$.

Furthermore, for each $d' \in [\ell_a, u_a]$, $\boldsymbol{\Pi}'_s(d')$ is a distribution and it has the same support as $\Pi_s(d')$: Let us fix $d' \in [\ell_a, u_a]$. Since $\boldsymbol{\Pi}_s^\xi$ and Π_s^ξ are ξ-approximations of Π_s, for each $s' \in S_{\text{set}} \cup S_{\text{off}}$, we have if $\Pi_s(d')(s') = 0$ then $\boldsymbol{\Pi}_s^\xi(d')(s') = 0$ and $\Pi_s^\xi(d')(s') = 0$ (if defined) and thus also $\boldsymbol{\Pi}'_s(d')(s') = 0$. Moreover, since $\xi \le \Pi_s^{\min}/3$ and $|\boldsymbol{\Pi}'_s(d')(s') - \Pi_s(d')(s')| \le 2\xi$, we have that if $\Pi_s(d')(s') > 0$ then $\boldsymbol{\Pi}'_s(d')(s') > 0$ for each $s' \in S_{\text{set}} \cap S_{\text{off}}$. Thus, $\boldsymbol{\Pi}'_s(d')$ has the same support as $\Pi_s(d')$. Finally, observe that $\sum_{s'\in S_{\text{set}}\cup S_{\text{off}}} \Delta\Pi(s') = 0$. Thus, it holds that $\sum_{s'\in S_{\text{set}}\cup S_{\text{off}}} \boldsymbol{\Pi}'_s(d')(s') = 1$. All the previous statements imply that $\boldsymbol{\Pi}'_s(d')$ is a distribution and thus, $\boldsymbol{\Pi}'_s$ is a κ-approximation of Π_s on $[\ell_a, u_a]$. □

3.4 Parametric ACTMCs with Non-localized Alarms

In the preceding section, we presented the symbolic Algorithm 1 to solve the ε-optimal parameter synthesis problem for parametric ACTMCs with localized alarms. However, our methods can also be extended to synthesize parameters when the given parametric ACTMC $\mathcal{N}$ has non-localized alarms. Note that if we create the semi-MDP $\mathcal{M}_\mathcal{N}$ in the same way as for the localized case, there might be multiple a-setting states for some alarm a. Hence, we need to restrict ourselves to policies that, for each alarm a, return the action with the same alarm parameter in all a-setting states. That is, we restrict the policy space to policies σ that for each $a \in A$ satisfy that $\{d : s \in S_{\text{set}} \cap S_a, \langle\!\langle s, d\rangle\!\rangle = \sigma(s)\}$ is a singleton. Let $D_\mathcal{M}$ be the set of policies of a semi-MDP $\mathcal{M}$ that satisfy this restriction.

As in the localized case, our aim is now to find a policy for a semi-MDP $\mathcal{M} \in [\mathcal{M}_\mathcal{N}\langle\delta\rangle]_\kappa$ that has a minimal expected mean payoff among policies in $D_\mathcal{M}$. This is equivalent to finding stationary deterministic policies in a certain subclass of partially observable MDPs (POMDPs) that is a generalization of semi-MDPs. This problem is known to be NP-hard [30], and hence we hardly can expect an exact polynomial-time algorithm. There are two exponential-time algorithms that solve this problem: an exhaustive search through the policy space and a *branch and bound method*

that prunes the search space on the fly [30]. These methods are not feasible in our case, since the semi-MDPs we would generate have a large action (policy) space. There are also heuristic approaches that run much faster but might not return an optimal policy: *gradient-based policy iteration* (GBPI) [11], *modified value iteration* (MVI) [38], and regular gradient descent [6]. We can easily accommodate our symbolic approach in the GBPI and MVI methods.

In the context of this article, we explain the GBPI as an extension of the standard policy iteration when executed on the semi-MDP $\mathcal{M} \in [\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_\kappa$. The initial policy is chosen from $D_{\mathcal{M}}$. In the policy improvement step, in addition to the gain and bias, we compute also the steady-state distribution $\psi : S_{\text{set}} \cup S_{\text{off}} \to \mathbb{Q}_{\geq 0}$ of $\mathcal{M}$ for the current policy. This can be done by the standard polynomial-time methods. In the policy improvement step, we then rank each action $\langle\!\langle s, d \rangle\!\rangle$ for every state $s \in S_{\text{set}} \cap S_a$ and alarm $a \in A$ by a function

$$\sum_{s \in S_{\text{set}} \cap S_a} \psi(s) \cdot \mathcal{R}^\kappa_s[g, \mathbf{h}](d),$$

where $\mathcal{R}^\kappa_s[g, \mathbf{h}](d) = \in^\kappa_s(d) - g \cdot \Theta^\kappa_s(d) + \Pi^\kappa_s(d) \cdot \mathbf{h}$ and $\in^\kappa_s$, Θ^κ_s, and Π^κ_s are the determining functions of $\mathcal{M}$. Hence, when using the current policy the GBPI gives larger weights to ranking functions corresponding to states where $\mathcal{M}$ stays a larger amount of time. Note that for each $a \in A$ the GBPI method ranks actions of states in $S_{\text{set}} \cap S_a$ by the same function. This implies that the improved policy will be in $D_{\mathcal{M}}$. The remaining parts of the GBPI are the same as for the standard policy iteration and one can easily adapt Algorithm 1 to use the GBPI. Note that we need to strengthen Assumption 3 such that roots of the derivative of $\sum_{s \in S_{\text{set}} \cap S_a} \psi(s) \cdot \boldsymbol{\mathcal{R}}^\kappa_s[g, \mathbf{h}](d)$ can be efficiently found for any given rational distribution ψ. This holds for all alarm distributions that we consider in the next section. Note that the GBPI may not return the optimal policy and it may not even terminate [28], i.e., a theorem corresponding to Theorem 3.1 can only be expected when posing additional assumptions on $\mathcal{N}$. However, the use of GBPI may result in satisfactory policies for realistic models, what was demonstrated in [11, 28]. For more details on applications of the generalized approach for non-localized parametric ACTMCs, see [26].

4 EXAMPLE ALARM DISTRIBUTIONS

We now demonstrate that our approach using the explicit and symbolic algorithms developed in the preceding sections is applicable to solve the ε-optimal parameter synthesis problem for parametric ACTMCs with well-known distributions. This technical section also provides theoretical background needed for implementation of Algorithm 1. Even though some of the obtained bounds are loose to simplify the presentation, our experimental evaluation provided in the next section shows promising results. Note that the following theorem holds for any kind of parametric ACTMCs, localized and non-localized ones.

Theorem 4.1. *Assumptions 1–4 are fulfilled for parametric ACTMCs with rational-valued cost functions where for all $a \in A$ we have that $F_a[x]$ is either a Dirac, uniform, exponential, Erlang, or a Weibull distribution.*

In the rest of this section, we develop the proof of Theorem 4.1. First, we provide technical lemmas in Section 4.1 to estimate perturbation bounds on semi-MDPs to get small errors in the expected mean payoff. Then, in Section 4.2, we provide auxiliary assumptions and their relationship to Assumptions 1–4 that will ease the actual proof of Theorem 4.1 for all mentioned distributions types. For the following, let us fix a strongly connected parametric ACTMC $\mathcal{N} = (S, \lambda, P, A, \langle S_a \rangle, \langle F_a[d] \rangle, \langle \ell_a \rangle, \langle u_a \rangle, \langle P_a \rangle)$ and cost functions $\mathcal{R}, \mathcal{I}, \mathcal{I}_A$, where all constants and functions are rational.

4.1 Computation of Sufficiently Small Precisions

For obtaining the perturbation bound κ of Assumption 1, we require technical statements that connect the error bound of a fraction with perturbation bounds of its numerator and denominator. Intuitively, to guarantee an error φ of a fraction, the numerator and the denominator have to be computed with a certain precision. The proof of the following lemma is technical and thus omitted—it can be found, e.g., in [5, 26].

Lemma 4.2. *For every $a, b, a', b', \overline{a}, \overline{b}, \underline{b}, \varphi \in \mathbb{R}_{>0}$ such that $a \leq \overline{a}$, and $\underline{b} \leq b \leq \overline{b}$, we have that*

$$\text{if both } |a - a'| \text{ and } |b - b'| \text{ are } \leq \frac{\underline{b}^2 \cdot \varphi}{\overline{a} + \overline{b} + \overline{b} \cdot \varphi} \text{ then } \left| \frac{a}{b} - \frac{a'}{b'} \right| \leq \varphi.$$

Another technical lemma establishes an upper bound on κ such that the expected mean-payoff achieved by a given policy changes among κ-approximations at most by a given $\varepsilon > 0$.

Lemma 4.3. *Let $\mathcal{M}$ be a strongly connected semi-MDP, σ be a policy such that $\mathcal{M}$ stays strongly connected when the set of actions is restricted to those selected by σ. Let $\mathcal{M}'$ be a κ-approximation of $\mathcal{M}$. Then for every error $\varepsilon > 0$, we have $|\mathbb{E}^{\sigma}_{\mathcal{M}}[\mathrm{MP}] - \mathbb{E}^{\sigma}_{\mathcal{M}'}[\mathrm{MP}]| \leq \varepsilon$ if*

$$\kappa \leq \min \left\{ \frac{(t_{\min}/2)^2 \cdot \frac{\varepsilon}{n}}{2w_{\max} \cdot (2/Q_{\min})^n \cdot (2 + \frac{\varepsilon}{n}) \cdot (1 + 2n w_{\max} (2/Q_{\min})^n)}, \frac{Q_{\min}}{2}, \frac{t_{\min}}{2}, \frac{c_{\max}}{2} \right\}.$$

Here, $Q_{\min}, t_{\min}, t_{\max}, c_{\max} \in \mathbb{Q}_{>0}$ are bounds on the minimal probability, minimal and maximal expected time step, and maximal expected costs occurring in $\mathcal{M}$, n is the number of states of $\mathcal{M}$, and $w_{\max} = \max\{c_{\max}, t_{\max}\}$.

Proof. Let $\mathcal{M} = (M, Act, Q, t, c)$ be a semi-MDP with $M = \{m_1, \ldots, m_n\}$ and σ some policy for $\mathcal{M}$. Note that every κ-approximation $\mathcal{M}'$ of $\mathcal{M}$ can be obtained via a sequence of semi-MDPs $\mathcal{M}_1, \ldots, \mathcal{M}_{n+1}$, where $\mathcal{M}_1 = \mathcal{M}$, $\mathcal{M}_{n+1} = \mathcal{M}'$, and every $\mathcal{M}_{i+1}$ is obtained from $\mathcal{M}_i$ by modifying only the values of $Q(b)$, $t(b)$, and $c(b)$ with $b \in Act_{m_i}$. Note further that the bounds $Q_{\min}, t_{\min}, t_{\max}, c_{\max} \in \mathbb{Q}_{>0}$ may not hold for all $\mathcal{M}_i$ due to the changes in the previous semi-MDPs. Hence, we require

$$\forall m \in M : \kappa \leq \min\{Q_{\min}, t_{\min}, c_{\max}\}/2,$$

and then the new constants $Q'_{\min} = Q_{\min}/2$, $t'_{\min} = t_{\min}/2$, $t'_{\max} = 2t_{\max}$, and $c'_{\max} = 2c_{\max}$ correctly bound all semi-MDPs in the sequence. Formally, for each $1 \leq i \leq n+1$, $\mathcal{M}_i = (M, Act, Q_i, t_i, c_i)$ satisfies

- $Q_i(b)(m') > 0$ implies $Q_i(b)(m') \geq Q_{\min}/2 = Q'_{\min}$,
- $t'_{\min} = t_{\min}/2 \leq t_i(b) \leq 2t_{\max} = t'_{\max}$,
- $c_i(b) \leq 2c_{\max} = c'_{\max}$

for all $m, m' \in M$ and $b \in Act_m$.

Now, it suffices to construct $\kappa \in \mathbb{Q}_{>0}$ such that $\kappa \leq \min\{Q_{\min}, t_{\min}, c_{\max}\}/2$ and the expected mean-payoff obtained by applying σ to $\mathcal{M}_i$ and $\mathcal{M}_{i+1}$ differs by at most ε/n.

Let us discuss one particular change, say in $m_i = m$, and for the sake of simplicity omit indexes in the following. For every $m' \in M$, let $\mathbb{E}^{\sigma}[\mathrm{Cost}(m, m')]$ and $\mathbb{E}^{\sigma}[\mathrm{Time}(m, m')]$ be the expected cost and time accumulated before visiting m' (in at least one transition) along a run that has been initiated in m. As $\mathcal{M}$ is strongly connected, every m' is eventually visited by a run initiated in m with probability one and the mean-payoff value $\mathbb{E}^{\sigma}[\mathrm{MP}]$ achieved by σ in $\mathcal{M}$ can be expressed as

$$\mathbb{E}^{\sigma}[\mathrm{MP}] = \frac{\mathbb{E}^{\sigma}[\mathrm{Cost}(m, m)]}{\mathbb{E}^{\sigma}[\mathrm{Time}(m, m)]}. \tag{3}$$

To use Lemma 4.2, we need to provide an upper bound on the numerator and upper and lower bounds on the denominator. The expected number of transitions required to visit m'' from m' can be bounded from above by $(n/Q'_{\min})^n$. Hence,

- $\mathbb{E}^\sigma[\mathrm{Cost}(m', m'')] \leq c'_{\max} \cdot (n/Q'_{\min})^n$, and
- $t'_{\min} \leq \mathbb{E}^\sigma[\mathrm{Time}(m', m'')] \leq t'_{\max} \cdot (n/Q'_{\min})^n$.

Moreover, according to Lemma 4.2, we need to bound the changes in the numerator and denominator. For this purpose, observe that using standard flow-equation arguments, we obtain

$$\mathbb{E}^\sigma[\mathrm{MP}] = \frac{\mathbb{E}^\sigma[\mathrm{Cost}(m,m)]}{\mathbb{E}^\sigma[\mathrm{Time}(m,m)]} = \frac{c(\sigma(m)) + \sum_{m' \in M \setminus \{m\}} Q(\sigma(m))(m') \cdot \mathbb{E}^\sigma[\mathrm{Cost}(m',m)]}{t(\sigma(m)) + \sum_{m' \in M \setminus \{m\}} Q(\sigma(m))(m') \cdot \mathbb{E}^\sigma[\mathrm{Time}(m',m)]}. \tag{4}$$

Note that for $m' \neq m$, $\mathbb{E}^\sigma[\mathrm{Cost}(m',m)]$ and $\mathbb{E}^\sigma[\mathrm{Time}(m',m)]$ do not depend on the actions (and their Q, t, and c values) of Act_m. Hence, these values do not change when modifying $Q(b)$, $t(b)$, and $c(b)$ only for $b \in Act_m$, and we can treat them as constants. If $Q(\sigma(m))(m')$, $t(\sigma(m))$, and $c(\sigma(m))$ change at most by κ, then the numerator and the denominator of the above fraction (4) change at most by $\kappa + n \cdot \kappa \cdot c'_{\max}(n/Q'_{\min})^n$ and $\kappa + n \cdot \kappa \cdot t'_{\max}(n/Q'_{\min})^n$, respectively. To this end, we use maximum to bound both of them. By Lemma 4.2, the error of the fraction is bounded by ε/n if

$$\kappa + n \cdot \kappa \cdot w'_{max}(n/Q'_{\min})^n \leq \frac{(t'_{\min})^2 \cdot \frac{\varepsilon}{n}}{c'_{\max} \cdot (n/Q'_{\min})^n + t'_{\max} \cdot (n/Q'_{\min})^n \cdot (1 + \frac{\varepsilon}{n})},$$

where $w'_{max} = \max\{c'_{\max}, t'_{\max}\}$, which can be strengthened to

$$\kappa + n \cdot \kappa \cdot w'_{max}(n/Q'_{\min})^n \leq \frac{(t'_{\min})^2 \cdot \frac{\varepsilon}{n}}{w'_{max} \cdot (n/Q'_{\min})^n \cdot (2 + \frac{\varepsilon}{n})},$$

i.e.,

$$\kappa \leq \frac{(t'_{\min})^2 \cdot \frac{\varepsilon}{n}}{w'_{max} \cdot (n/Q'_{\min})^n \cdot (2 + \frac{\varepsilon}{n}) \cdot (1 + n \cdot w'_{max}(n/Q'_{\min})^n)}.$$

Using the bounds of the original semi-MDP and the above mentioned restriction on κ, we obtain

$$\kappa \leq \min\left\{ \frac{(t_{\min}/2)^2 \cdot \frac{\varepsilon}{n}}{2w_{\max} \cdot (2n/Q_{\min})^n \cdot (2+\frac{\varepsilon}{n}) \cdot (1+2n w_{\max}(2n/Q_{\min})^n)}, \frac{Q_{\min}}{2}, \frac{t_{\min}}{2}, \frac{c_{\max}}{2} \right\},$$

where $w_{\max} = \max\{c_{\max}, t_{\max}\}$. □

4.2 Auxiliary Assumptions

We define new Assumptions **A)** and **B)** for $\mathcal{N}$ that guarantee important discretization bounds and, as we will show later on, imply Assumption 1 and 4:

A) For every $s \in S_{\mathrm{set}}$, there are effectively computable positive rational bounds

$$\Pi_s^{\min}, \Theta_s^{\min}, \Theta_s^{\max}, \text{ and } \Subset_s^{\max}$$

such that for all $d \in [\ell_a, u_a]$ where a is the alarm of s, i.e., $s \in S_a$, we have
- $\Pi_s^{\min} \leq \Pi_s(d)(s')$ for all $s' \in S_{\mathrm{set}} \cup S_{\mathrm{off}}$ where $\Pi_s(d)(s') > 0$
- $\Theta_s^{\min} \leq \Theta_s(d) \leq \Theta_s^{\max}$
- $\Subset_s(d) \leq \Subset_s^{\max}$.

B) For every $a \in A$, $s \in S_{\mathrm{set}} \cap S_a$, and $\kappa \in \mathbb{Q}_{>0}$ there is a discretization bound $\delta_{(s,\kappa)} \in \mathbb{Q}_{>0}$ such that for every $d, d' \in [\ell_a, u_a]$ and $|d - d'| \leq \delta_{(s,\kappa)}$ it holds that

- $|\Pi_s(d)(s') - \Pi_s(d')(s')| \leq \kappa$ for all $s' \in S_{\text{set}} \cup S_{\text{off}}$,
- $|\Theta_s(d) - \Theta_s(d')| \leq \kappa$, and
- $|\mathbb{\in}_s(d) - \mathbb{\in}_s(d')| \leq \kappa$.

Intuitively, Assumption **B)** connects κ-approximation and δ-discretization.

Lemma 4.4. *In case $\mathcal{N}$ fulfills Assumptions **A)** and **B)**, Assumptions 1 and 4 are fulfilled as well.*

Proof. We have to show that for every $\varepsilon \in \mathbb{Q}_{>0}$, there are computable $\delta : A \to \mathbb{Q}_{>0}$ and $\kappa \in \mathbb{Q}_{>0}$ such that for every $\mathcal{M} \in [\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_\kappa$ and every optimal policy σ for $\mathcal{M}$, the associated parameter function $\mathbf{d}^\sigma$ is ε-optimal for $\mathcal{N}$.

Assumption **A)** provides that rational bounds on the minimal probability, minimal and maximal time step, and maximal costs occurring in $\mathcal{M}_{\mathcal{N}}$ are effectively computable for any $s \in S_{\text{set}}$. Computing such bounds for every $s \in S_{\text{set}}$ and applying Lemma 4.2 yields that for every $\varepsilon \in \mathbb{Q}_{>0}$, we can compute a sufficiently small $\kappa \in \mathbb{Q}_{>0}$ such that for every $\mathcal{M} \in [\mathcal{M}_{\mathcal{N}}]_\kappa$ and every optimal policy σ for $\mathcal{M}$ the associated parameter function $\mathbf{d}^\sigma$ is ε-optimal for $\mathcal{N}$. Let us fix such an $\varepsilon \in \mathbb{Q}_{>0}$ and a $\kappa \in \mathbb{Q}_{>0}$. Thus, when computing the action effects, an error of κ is possible. To connect approximation and discretization precision, we dedicate a precision of $\kappa/2$ for the action effects (i.e., we will allow only $\kappa/2$-approximation for Assumption 1) and a precision of $\kappa/2$ for the errors caused by discretization. Let $\delta : A \to \mathbb{Q}_{>0}$ be a function obtained from Assumption **B)** by setting $\delta(a) \stackrel{\text{def}}{=} \min\{\delta_{(s,\kappa/2)} : s \in S_a \cap S_{\text{set}}\}$ for each $a \in A$. We show that an optimal policy σ of $\mathcal{M} \in [\mathcal{M}_{\mathcal{N}}\langle\delta\rangle]_{\kappa/2}$ is optimal for some $\mathcal{M}' \in [\mathcal{M}_{\mathcal{N}}]_\kappa$ what induces an ε-optimal policy $\mathbf{d}^\sigma$ for $\mathcal{N}$ and validity of Assumption 1: For each $d \in [\ell_a, u_a)$ we define $\delta'(d) \stackrel{\text{def}}{=} \ell_a + i \cdot \delta(a)$ where $i \in \mathbb{N}_{\geq 0}$ is such that $d \in [\ell_a + i \cdot \delta(a), \min\{u_a, \ell_a + (i+1) \cdot \delta(a)\})$. Let $\Pi_s, \Theta_s, \mathbb{\in}_s$ be the determining functions of $\mathcal{M}$. We define the determining functions $\Pi'_s, \Theta'_s, \mathbb{\in}'_s$ of $\mathcal{M}' \in [\mathcal{M}_{\mathcal{N}}]_{2\kappa/2}$ as follows: $\Pi'_s(d) \stackrel{\text{def}}{=} \Pi_s(\delta'(d))$, $\Theta'_s(d) \stackrel{\text{def}}{=} \Theta_s(\delta'(d))$, and $\mathbb{\in}'_s(d) \stackrel{\text{def}}{=} \mathbb{\in}_s(\delta'(d))$ for each $d \in [\ell_a, u_a)$ and $\Pi'_s(u_a) \stackrel{\text{def}}{=} \Pi_s(u_a)$, $\Theta'_s(u_a) \stackrel{\text{def}}{=} \Theta_s(u_a)$, and $\mathbb{\in}'_s(u_a) \stackrel{\text{def}}{=} \mathbb{\in}_s(u_a)$. Clearly, $\mathcal{M}' \in [\mathcal{M}_{\mathcal{N}}]_{2\kappa/2} = [\mathcal{M}_{\mathcal{N}}]_\kappa$ and σ is an optimal policy of $\mathcal{M}'$. Assumption 4 is a trivial consequence of Assumption **A)**. □

We now show for each kind of distribution mentioned in Theorem 4.1 that Assumptions **A)**, **B)**, 2, and 3 are fulfilled. Application of Lemma 4.4 then completes the proof of Theorem 4.1.

4.3 Dirac Alarm Distributions

We start with showing that the Assumptions are fulfilled for alarm distributions that are Dirac. We assume a fixed $s \in S_{\text{set}}$ such that $s \in S_a$, $F_a[d](\tau) = 1$ for all $\tau \geq d$, and $F_a[d](\tau) = 0$ for all $\tau < d$, where $d \in [\ell_a, u_a] \subset (0, \infty)$.[9]

Note that for Dirac distributions, we can easily obtain $\Pi_s(d)$ by employing a Poisson distribution ranging over the number of exponentially distributed delay transitions until time d. Then,

$$\Pi_s(d) \quad = \quad \sum_{i=0}^{\infty} e^{-\lambda d} \frac{(\lambda d)^i}{i!} \cdot \left(\mathbf{1}_s \cdot \overline{P}^i\right) \cdot P_a,$$

where i represents the number of exponential transitions fired before the alarm rings (i.e., time d), $e^{-\lambda d} \frac{(\lambda d)^i}{i!}$ is the corresponding probability according to the Poisson distribution, $\mathbf{1}_s$ is a vector of zeroes except for $\mathbf{1}_s(s) = 1$, and $\overline{P} : S \times S \to [0, 1]$ is a probability matrix that is as P but where all states in S_{off} are made absorbing, i.e., $\overline{P}(s', \cdot) = P(s', \cdot)$ for all $s' \in S_a$, and $\overline{P}(s', s') = 1$ for all $s' \in S \setminus S_a$.

[9] Note that we need to restrict ℓ_a and u_a to work with correct parametric ACTMC.

Similarly, $\in_s$ is the expected total cost computable as follows:

$$\in_s(d) = \sum_{i=0}^{\infty} e^{-\lambda d}\frac{(\lambda d)^i}{i!}\left(\sum_{j=0}^{i-1}\left(\mathbf{1}_s\cdot\overline{P}^j\right)\cdot\left(\frac{d\cdot\overline{\mathcal{R}}}{i+1}+\overline{\mathcal{I}}\right) + \left(\mathbf{1}_s\cdot\overline{P}^i\right)\cdot\left(\frac{d\cdot\overline{\mathcal{R}}}{i+1}+\overline{\mathcal{I}}_A\right)\right),$$

where $\overline{\mathcal{R}}$ is a vector that is the same as $\mathcal{R}$ but returns 0 for all states of $S\setminus S_a$, and $\overline{\mathcal{I}}, \overline{\mathcal{I}}_A : S \to \mathbb{R}_{\geq 0}$ are vectors that assign zero to states $S\setminus S_a$ and to each state of S_a they assign the expected instant execution cost of the next delay and the next alarm transition, respectively. Note that Θ_s is a special case of $\in_s$ where $\mathcal{R}(s) = 1$ for all $s \in S$, and $\mathcal{I}, \mathcal{I}_A$ return zero for all arguments.

Note that the functions $\Pi_s(\hat{\tau})$, $\Theta_s(\hat{\tau})$, and $\in_s(\hat{\tau})$ are defined as infinite sums. However, for every $\hat{\tau} \in \mathbb{Q}_{>0}$ and $\kappa \in \mathbb{Q}_{>0}$, we can effectively compute a $k \in \mathbb{N}_{>0}$ and truncated versions of Π_s, Θ_s, and $\in_s$ for all $s \in S_{\text{set}}$, obtained by taking the first k summands of the corresponding defining series. Then, for all $s \in S_{\text{set}}$ and $d \leq \hat{\tau}$ these truncated versions under-approximate up to an absolute error of at most κ the values of $\Pi_s(d)$, $\Theta_s(d)$, and $\in_s(d)$, respectively. We restrict k to be always larger than $|S_a|$, thus we detect all non-zero probabilities and all distributions will have the same support. We use

$$\Pi_s^{\hat{\tau},\kappa}, \Theta_s^{\hat{\tau},\kappa}, \text{ and } \in_s^{\hat{\tau},\kappa}$$

to denote these truncated versions that are κ-approximations for all values d up to $\hat{\tau}$. Note that due to the factor $e^{-\lambda d}$, the values of $\Pi_s^{\hat{\tau},\kappa}$, $\Theta_s^{\hat{\tau},\kappa}$, and $\in_s^{\hat{\tau},\kappa}$ are still irrational even for rational d's. Nevertheless, these values can be effectively approximated by rational numbers up to a given arbitrarily small error. Also note that the components of $\Pi_s^{\hat{\tau},\kappa}$ may not sum up to 1. When we need to understand $\Pi_s^{\hat{\tau},\kappa}$ as a distribution, the remaining probability is split evenly among the states where $\Pi_s^{\hat{\tau},\kappa}$ is positive.

Assumption A). Deriving the bounds $\Pi_s^{\min}$, $\Theta_s^{\min}$, $\Theta_s^{\max}$, and $\in_s^{\max}$ is easy. We put

- $\Pi_s^{\min} \stackrel{\text{def}}{=} (P_{\min})^n \cdot \min\left\{\frac{e^{-\lambda d}(\lambda d)^k}{k!} : 0 \leq k \leq n, d \in \{\ell_a, u_a\}\right\}$
- $\Theta_s^{\min} \stackrel{\text{def}}{=} \int_0^{\ell_a} x\cdot\lambda\cdot e^{-\lambda x}\,\mathrm{d}x + \ell_a\cdot e^{-\lambda\ell_a} = \frac{1-e^{-\lambda\cdot\ell_a}}{\lambda}$,
- $\Theta_s^{\max} \stackrel{\text{def}}{=} u_a$, and
- $\in_s^{\max} \stackrel{\text{def}}{=} \mathcal{R}_{\max}\cdot u_a + \mathcal{I}_{\max}\cdot(\lambda\, u_a + 1)$,

where $n = |S_a|$, $P_{\min} = \min\{P(s,s'), P_a(s,s') : s \in S_a, s' \in S\}$, $\mathcal{R}_{\max} = \max\{\mathcal{R}(s) : s \in S_a\}$, and $\mathcal{I}_{\max} = \max\{\mathcal{I}(s,s'), \mathcal{I}_A(s,s') : s \in S_a, s' \in S\}$. Note that although some of the defining expressions involve the function e^x, we can still effectively under-approximate their values by positive rationals.

Assumption B). To define an appropriate $\delta_{(s,\kappa)} \in \mathbb{Q}_{>0}$, first note that for every $s' \in S$, $d \in [\ell_a, u_a]$, and every $\delta \in \mathbb{Q}_{>0}$ such that $d-\delta \in [\ell_a, u_a]$, we have that

- $|\Theta_s(d) - \Theta_s(d-\delta)| \leq \delta$,
- $|\in_s(d) - \in_s(d-\delta)| \leq \delta\mathcal{R}_{\max} + 2\lambda\delta\mathcal{I}_{\max}$, and
- $|\Pi_s(d)(s') - \Pi_s(d-\delta)(s')| \leq \lambda\delta$,

where $\mathcal{R}_{\max}$ and $\mathcal{I}_{\max}$ are as defined above. Note also that $\delta\mathcal{R}_{\max}$ bounds the change of rate cost caused by changing d by δ, $\lambda\delta$ bounds the change of the number of delay transitions in time d vs. time $d-\delta$, and $\lambda\delta$ can be also used as a bound on the change of probabilities in the transient probability distribution in CTMCs with rate λ after time d vs. $d-\delta$ (see, e.g., Theorem 2.1.1 of

[34]). Hence, for a given $\kappa > 0$, we can set

$$\delta_{(s,\kappa)} = \min\left\{\kappa, \frac{\kappa}{\mathcal{R}_{\max} + 2\lambda\mathcal{I}_{\max}}\right\}.$$

Assumption 2). Let $\hat{\tau} = u_a$. We first obtain functions $\Pi_s^{\hat{\tau},\kappa/2}$, $\Theta_s^{\hat{\tau},\kappa/2}$, and $\in_s^{\hat{\tau},\kappa/2}$ and then $\kappa/2$ approximate them by rational functions using truncated Taylor series for $e^{(\cdot)}$. We define Π_s^{κ}, Θ_s^{κ}, and $\in_s^{\kappa}$ as the obtained rational $\kappa/2$-approximations of $\Pi_s^{\hat{\tau},\kappa/2}$, $\Theta_s^{\hat{\tau},\kappa/2}$, and $\in_s^{\hat{\tau},\kappa/2}$, respectively.

Assumption 3). For $\hat{\tau} = u_a$, we put

$$\mathbf{\Pi}_s^{\kappa} \stackrel{\text{def}}{=} \Pi_s^{\hat{\tau},\kappa},\ \mathbf{\Theta}_s^{\kappa} \stackrel{\text{def}}{=} \Theta_s^{\hat{\tau},\kappa},\ \mathbf{\in}_s^{\kappa} \stackrel{\text{def}}{=} \in_s^{\hat{\tau},\kappa},\ \text{and recall } \boldsymbol{\mathcal{R}}_s^{\kappa}[g,\mathbf{h}](d) \stackrel{\text{def}}{=} \mathbf{\in}_s^{\kappa}(d) - g\cdot\mathbf{\Theta}_s^{\kappa}(d) + \mathbf{\Pi}_s^{\kappa}(d)\cdot\mathbf{h},$$

where $g \in \mathbb{Q}$ and $\mathbf{h} : S_{\text{set}} \cup S_{\text{off}} \to \mathbb{Q}$ are constant (cf. Equation ($\star$)). Observe that $\mathbf{\in}_s^{\kappa}(d), \mathbf{\Theta}_s^{\kappa}(d)$, and $\mathbf{\Pi}_s^{\kappa}(d)\cdot\mathbf{h}$ are all of the form $e^{-\lambda d}\cdot \mathit{Poly}(d)$, where $\mathit{Poly}(d)$ is a polynomial. Hence, the first derivative of $\boldsymbol{\mathcal{R}}_s^{\kappa}[g,\mathbf{h}](d)$ also has the special form $e^{-\lambda d}\cdot V_{s,g,\mathbf{h}}$ where $V_{s,g,\mathbf{h}}$ is a polynomial of one free variable. Since $e^{-\lambda d}$ is positive for all $d \in \mathbb{R}_{\geq 0}$, the derivative $e^{-\lambda d}\cdot V_{s,g,\mathbf{h}}$ is zero iff $V_{s,g,\mathbf{h}}$ is zero. Hence, the roots of the derivative in the interval $[\ell_a, u_a]$ can be approximated by approximating the roots of $V_{s,g,\mathbf{h}}$, which is computationally easy (e.g., using tools such as Maple [7]).

4.4 Symbolic Functions for Action Effects with Non-Dirac Distributions

To illustrate the main idea that we will repeatedly apply in the following sections, let us exemplify the cost function $\in_s(d)$ for a non-Dirac continuous distribution $F_a[d]$ with a known density function $f_a[d] : \mathbb{R} \to \mathbb{R}_{\geq 0}$. This function can be computed by integration of the cost function for Dirac distribution, denoted as Dirac-$\in_s(\cdot)$, multiplied by the density function. That is,

$$\in_s(d) = \int_{min_a}^{max_a} \text{Dirac-}\in_s(\tau)\cdot f_a[d](\tau)\ \mathrm{d}\tau, \tag{5}$$

where τ stands for the randomly chosen time according to the non-Dirac distribution and min_a and max_a bound the support of $f_a[d]$. Note that for every (even non-continuous) distributions, we can obtain $\in_s(d)$ by employing Lebesgue integral, i.e., let μ_a be the measure of $F_a[d]$, then

$$\in_s(d) = \int_{[min_a, max_a]} \text{Dirac-}\in_s(\tau)\ \mu_a(\mathrm{d}\tau).$$

With g standing for some function $\Pi_s, \Theta_s, \in_s, \Pi_s^{\hat{\tau},\kappa}, \Theta_s^{\hat{\tau},\kappa}, \in_s^{\hat{\tau},\kappa}, \Pi_s^{\min}, \Theta_s^{\min}, \Theta_s^{\max}$, or $\in_s^{\max}$, we use the notation Dirac-g in the following sections to denote the respective quantities computed for the Dirac distribution.

4.5 Uniform Alarm Distributions

In a (continuous) uniform distribution there are two parameters, the lower bound on possible random values, say α, and the upper bound, say β. Let us note that using a smart construction, we can easily synthesize both of the parameters by our algorithm. Note that the first parameter α can be modeled as an alarm with Dirac distribution (parametrized by α that we can synthesize) that subsequently enables a uniformly distributed alarm with the first parameter fixed to 0 and the second parameter, say β', that is also subject of synthesis. Then, the required parameters of the original uniform distribution are α and $\beta = \alpha + \beta'$. Note that the newly created uniformly distributed alarm may not be localized, what can rule out the applicability of Algorithm 1 and force us to use the synthesis algorithm for ACTMCs with non-localized alarms.

Therefore, in what follows we consider alarms that are *uniformly* distributed on a time interval starting in 0 with a parametrized length. That is, we assume a fixed $s \in S_{\text{set}}$ such that $s \in S_a$,

$F_a[d](\tau) = \frac{\tau}{d}$ for all $0 \leq \tau \leq d$, $F_a[d](\tau) = 0$ for all $\tau < 0$, and $F_a[d](\tau) = 1$ for all $\tau > d$, where $d \in [\ell_a, u_a] \subset (0, \infty)$.[10]

Assumption A). Note that for each $d \in [\ell_a, u_a]$, the uniform distribution has its support at most on $[0, u_a]$. Moreover, Dirac-Θ_s, and Dirac-$\Subset_s$ are bounded from above by Dirac-$\Theta_s^{\max}$, and Dirac-$\Subset_s^{\max}$ on $[0, u_a]$, respectively. Thus, also the integral expressions defining Θ_s and $\Subset_s$ are bounded, and we have

- $\Theta_s^{\max} \stackrel{\text{def}}{=} \text{Dirac-}\Theta_s^{\max} = u_a$ and
- $\Subset_s^{\max} \stackrel{\text{def}}{=} \text{Dirac-}\Subset_s^{\max} = \mathcal{R}_{\max} \cdot u_a + \mathcal{I}_{\max} \cdot (\lambda\, u_a + 1)$.

Note that for each $d \in [\ell_a, u_a]$, the uniform distribution has at least $1/2$ of the probability on $[\ell_a/2, u_a]$. Thus, we can use half of Dirac-$\Pi_s^{\min}$ and Dirac-$\Theta_s^{\min}$ for Dirac distribution on $[\ell_a/2, u_a]$:

- $\Pi_s^{\min} \stackrel{\text{def}}{=} (P_{\min})^n \cdot \min\left\{ \frac{e^{-\lambda d}(\lambda d)^k}{2 \cdot k!} : 0 \leq k \leq n, d \in \{\ell_a/2, u_a\} \right\}$ and
- $\Theta_s^{\min} \stackrel{\text{def}}{=} \frac{1 - e^{-\lambda \cdot \ell_a/2}}{2\lambda}$,

where n and $P_{\min}$ are the same as in Section 4.3.

Assumptions 2) and 3). We show that $\Subset_s$ can be κ-approximated by a function $\boldsymbol{\Subset}_s^{\kappa}(d)$ of the form $V(d)/d$, where $V(d)$ is a computable polynomial of rational coefficients. By using the same technique, we can construct also $\boldsymbol{\Theta}_s^{\kappa}$ and $\boldsymbol{\Pi}_s^{\kappa}$, which are of the same form. Hence, the function $\boldsymbol{\mathcal{R}}_s^{\kappa}[g, \mathbf{h}](d) = \boldsymbol{\Subset}_s^{\kappa}(d) - g \cdot \boldsymbol{\Theta}_s^{\kappa}(d) + \boldsymbol{\Pi}_s^{\kappa}(d) \cdot \mathbf{h}$ has also the same form, since g and $\mathbf{h}$ are constant. After differentiation of $\boldsymbol{\mathcal{R}}_s^{\kappa}[g, \mathbf{h}](d)$, we get a function of form $V'(d)/d^2$, where we isolate roots of polynomial $V'(d)$ to obtain the candidate actions. Note that the denominator d^2 may disable root $d = 0$ that is not within the eligible parameter values. Generally, the correctness of the symbolic algorithms is not harmed by including candidate actions computed from the "false positive" root, since these actions do not correspond to values that are near extrema of $\boldsymbol{\mathcal{R}}_s^{\kappa}[g, \mathbf{h}](d)$ and some other action of the candidate set must outperform all of them. Thus, the function $\boldsymbol{\mathcal{R}}_s^{\kappa}[g, \mathbf{h}](d)$ fulfills Assumptions 2 and 3.

Now, we show that $\Subset_s$ can be κ-approximated by a function $\boldsymbol{\Subset}_s^{\kappa}(d)$ of the form $V(d)/d$, where $V(d)$ is a computable polynomial of rational coefficients. From definition $\Subset_s(d)$ equals to $\int_0^d 1/d \cdot \text{Dirac-}\Subset_s(\tau)\, \mathrm{d}\tau$. Instead of Dirac-$\Subset_s(\tau)$, we can use its $\kappa/2$-approximation Dirac-$\Subset_s^{u_a, \kappa/2}(\tau) = P(\tau) \cdot e^{-\lambda\tau}$, where $P(\tau)$ is an univariate polynomial with rational coefficients. The main difficulty is the integration of $P(\tau) \cdot e^{-\lambda\tau}$. Fortunately, we can use the Taylor series representation for $e^{-\lambda\tau}$, i.e.,

$$e^{-\lambda\tau} = \sum_{n=0}^{\infty} \frac{1}{n!} \cdot (-\lambda\tau)^n.$$

Then, we can truncate the infinite sum, such that we cause at most $\kappa/2$ error in $P(\tau) \cdot e^{-\lambda\tau}$ for each $\tau \in [0, u_a]$ (observe that we can compute the maximal and minimal values of polynomial $P(\tau)$ for $\tau \in [0, u_a]$ using differentiation and root isolation of polynomials). Thus, we obtain polynomial $P(\tau) \cdot S(\tau)$ with rational coefficients, that κ-approximates Dirac-$\Subset_s(\tau)$. Hence, we can set

$$\boldsymbol{\Subset}_s^{\kappa}(d) = \frac{1}{d} \cdot \int_0^d P(\tau) \cdot S(\tau)\ \mathrm{d}\tau = \frac{1}{d} \cdot V(d),$$

which is a κ-approximation of $\Subset_s$, since $V(d)$ is κd-approximation of $\int_0^d P(\tau) \cdot S(\tau)\ \mathrm{d}\tau$.

[10] Note that we need to restrict ℓ_a and u_a to work with a correct parametric ACTMC.

Assumption B). Let us fix $s \in S_{\text{set}} \cap S_a$ and $\kappa \in \mathbb{Q}_{>0}$. We will show how to compute a discretization bound $\delta_{(s,\kappa,\in)} \in \mathbb{Q}_{>0}$ such that for every $d, d' \in [\ell_a, u_a]$ where $|d - d'| \leq \delta_{(s,\kappa,\in)}$ it holds that

$$|\in_s(d) - \in_s(d')| \leq \kappa.$$

First, we set $\kappa' \stackrel{\text{def}}{=} \kappa/3$. Using κ', we κ'-approximate $\in_s$ function by polynomial $\boldsymbol{\in}_s^{\kappa'}$ defined above for Assumption 3. Then, we use $\boldsymbol{\in}_s^{\kappa'}$ to obtain safe bounds on its first derivative (e.g., using its second derivative and root isolation on interval $[\ell_a, u_a]$). We use these bounds to obtain sufficiently small $\delta_{(s,\kappa,\in)}$ to cause at most κ' error in $\boldsymbol{\in}_s^{\kappa'}$. Then for every $d, d' \in [\ell_a, u_a]$ where $|d - d'| \leq \delta_{(s,\kappa,\in)}$ it holds that

$$|\in_s(d) - \in_s(d')| \leq 2\kappa' + |\boldsymbol{\in}_s^{\kappa'}(d) - \boldsymbol{\in}_s^{\kappa'}(d')| \leq 3\kappa' = \kappa.$$

Similarly, we can define and obtain bounds $\delta_{(s,\kappa,\Pi)}$ and $\delta_{(s,\kappa,\Theta)}$. Then $\delta_{(s,\kappa)} \stackrel{\text{def}}{=} \min_{\Gamma \in \{\Pi,\Theta,\in\}} \delta_{(s,\kappa,\Gamma)}$.

4.6 Erlang, Weibull, and Other Continuous Alarm Distributions

As we already noted in Section 2, the abstract assumptions enabling the applicability of our synthesis algorithms are satisfied also by other distributions. Since the arguments are increasingly more involved (and not used in our experiments reported in Section 5), we just sketch the arguments that show how to handle distributions with polynomial approximations and illustrate them on the Erlang and Weibull distributions. Note that exponential distribution is a special case of the Erlang and Weibull distributions, where the fixed shape constant k is 1.

In what follows, we assume a fixed $s \in S_{\text{set}} \cap S_a$ for some $a \in A$ and $F_a[d](\tau)$ such that for each $d \in [\ell_a, u_a]$ the CDF $F_a[d](\tau)$ is continuous and differentiable for all $\tau > 0$ and $F_a[d](\tau) = 0$ for all $\tau \leq 0$.

Assumption A). Let $f_a[d](\tau)$ be the derivative of $F_a[d](\tau)$, i.e., a probability density function. Note that for uniform distribution, we first bounded the support of the density function. Here the support of the density function can be whole $\mathbb{R}_{\geq 0}$ even for d bounded by ℓ_a and u_a. Hence, we first provide bounds min'_s and max'_s, satisfying

$$\int_0^{\infty} f_a[d](\tau) \cdot \langle\ .\ \rangle \,\mathrm{d}\tau - \int_{min'_s}^{max'_s} f_a[d](\tau) \cdot \langle\ .\ \rangle \,\mathrm{d}\tau \ \leq\ \frac{1}{2},$$

for all $\langle\ .\ \rangle \in \{\text{Dirac-}\Pi_s(\tau), \text{Dirac-}\Theta_s(\tau), \text{Dirac-}\in_s(\tau)\}$. To find such bounds, we strengthen the condition by requiring $\langle\ .\ \rangle \in \{\text{Dirac-}\Pi_s^{\max}, \text{Dirac-}\Theta_s^{\max}, \text{Dirac-}\in_s^{\max}\}$. Formally, we use 1 for $\text{Dirac-}\Pi_s^{\max}$, $\text{Dirac-}\Theta_s^{\max} \stackrel{\text{def}}{=} \tau$, and $\text{Dirac-}\in_s^{\max} \stackrel{\text{def}}{=} \mathcal{R}_{\max} \cdot \tau + \mathcal{I}_{\max} \cdot (\lambda\tau{+}1)$, where $\mathcal{R}_{\max}$ and $\mathcal{I}_{\max}$ are the same as in Section 4.3. Note that the bounds min'_s and max'_s exist because the density function has to decrease in its extrema faster than at most linearly increasing (in τ) functions $\text{Dirac-}\Theta_s^{\max}$ and $\text{Dirac-}\in_s^{\max}$.

Having such bounds min'_s and max'_s, we can safely set

- $\Theta_s^{\max} \stackrel{\text{def}}{=} \text{Dirac-}\Theta_s^{\max} + 1/2 = max'_s + 1/2$, and
- $\in_s^{\max} \stackrel{\text{def}}{=} \text{Dirac-}\in_s^{\max} + 1/2 = \mathcal{R}_{\max} \cdot max'_s + \mathcal{I}_{\max} \cdot (\lambda\, max'_s + 1) + 1/2$.

Let $c_s \stackrel{\text{def}}{=} \min_{d \in [\ell_a, u_a]} \int_{min'_s}^{max'_s} f_a[d](\tau)\,\mathrm{d}\tau$. One can easily bound c_s from below by $1/2$ or more exactly by $(1 - 1/(2 \cdot \max\{1, \text{Dirac-}\Theta_s^{\max}, \text{Dirac-}\in_s^{\max}\}))$ where τ is substituted by $\max'_s$. Using c_s, we can set

- $\Pi_s^{\min} \stackrel{\text{def}}{=} c_s \cdot \text{Dirac-}\Pi_s^{\min} = c_s \cdot (P_{\min})^n \cdot \min\{\frac{e^{-\lambda d}(\lambda d)^k}{k!} : 0{\leq}k{\leq}n, d \in \{min'_s, max'_s\}\}$, and
- $\Theta_s^{\min} \stackrel{\text{def}}{=} c_s \cdot \text{Dirac-}\Theta_s^{\min} = c_s \cdot \frac{1 - e^{-\lambda \cdot min'_s}}{2\lambda}$,

where n and $P_{\min}$ are the same as in Section 4.3.

Assumptions 2) and 3). Employing Assumption A), we can compute sufficiently small κ. Now, we define max_s such that for all $d \in [\ell_a, u_a]$ and $\langle\,.\,\rangle \in \{1, max_s, \mathcal{R}_{\max} \cdot \max_s + \mathcal{I}_{\max} \cdot (\lambda \max_s + 1)\}$ (i.e., $\langle\,.\,\rangle$ is the same as above, but we use max_s instead of τ), we have

$$1 - \int_0^{max_s} f_a[d](\tau)\,\mathrm{d}\tau \;\leq\; \frac{\kappa}{2 \cdot \langle\,.\,\rangle}.$$

Now, let $U(\tau)$ be a polynomial approximation of $f_a[d](\tau) \cdot \text{Dirac-}\mathbb{\in}_s(\tau)$ whose error is bounded by $\kappa/(2 \cdot max_s)$ on interval $[0, max_s]$. Then $\mathbb{\in}_s^\kappa = \int_0^{max_s} U(\tau)\,\mathrm{d}\tau$ is a κ-approximation of $\mathbb{\in}_s$, since

$$\left| \int_0^{\infty} f_a[d](\tau) \cdot \text{Dirac-}\mathbb{\in}_s(\tau)\,\mathrm{d}\tau - \int_0^{max_s} f_a[d](\tau) \cdot \text{Dirac-}\mathbb{\in}_s(\tau)\,\mathrm{d}\tau \right| \;\leq\; \frac{\kappa}{2},$$

which follows from the setting of max_s, and since

$$\left| \int_0^{max_s} f_a[d](\tau) \cdot \text{Dirac-}\mathbb{\in}_s(\tau)\,\mathrm{d}\tau - \int_0^{max_s} U(\tau)\,\mathrm{d}\tau \right| \;\leq\; \left| \int_0^{max_s} \frac{\kappa}{2 \cdot max_s}\,\mathrm{d}\tau \right| \;=\; \frac{\kappa}{2}.$$

Note that $U(\tau)$ is a polynomial with variable τ, but might not be polynomial in the parameter d. Thus, there is a question whether $\mathbb{\in}_s^\kappa = \int_0^{max_s} U(\tau)\,\mathrm{d}\tau$ is differentiable for every $d \in [\ell_a, u_a]$.

As a concrete example, let us consider a **Weibull distribution** with a fixed shape parameter $k \in \mathbb{N}_{>0}$ and the scale parameter $1/d$ (i.e., we aim at synthesizing an ε-optimal scale). The Weibull density function is

$$kd \cdot (\tau d)^{k-1} \cdot e^{-(\tau d)^k}.$$

As we know from Section 4.3, $\text{Dirac-}\mathbb{\in}_s^{max_s,\kappa}(\tau) = P(\tau) \cdot e^{-\lambda\tau}$, where $P(\tau)$ is a polynomial function. Note that in $f_a[d](\tau) \cdot \text{Dirac-}\mathbb{\in}_s^{max_s,\kappa}(\tau)$, we can approximate $e^{-(\tau d)^k} \cdot e^{-\lambda\tau}$, i.e., $e^{-(\tau d)^k - \lambda\tau}$, to an arbitrary precision for all $d \in [\ell_a, u_a]$ and $\tau \in [0, max_s]$ using a finite prefix of a Taylor series

$$\sum_{n=0}^{\infty} \frac{1}{n!} \cdot (-(\tau d)^k - \lambda\tau)^n.$$

Then, for some sufficiently high bound $i \in \mathbb{N}_{>0}$,

$$U(\tau) \;\stackrel{\text{def}}{=}\; kd \cdot (\tau d)^{k-1} \cdot P(\tau) \cdot \sum_{n=0}^{i} \frac{1}{n!} \cdot (-(\tau d)^k - \lambda\tau)^n$$

is a polynomial function of τ and d, and hence, $\mathbb{\in}_s^\kappa \stackrel{\text{def}}{=} \int_0^{max_s} U(\tau)\,\mathrm{d}\tau$ is a polynomial function of d. Similarly, we can provide polynomial κ-approximations $\mathbb{\Pi}_s^\kappa$ and $\mathbb{\Theta}_s^\kappa$, thus Assumption 3 is fulfilled. Moreover, $\mathbb{\Pi}_s^\kappa$, $\mathbb{\Theta}_s^\kappa$, and $\mathbb{\in}_s^\kappa$ are computable for each $d \in [\ell_a, u_a] \cap \mathbb{Q}$, thus Assumption 2 also holds.

As a second example, let us consider an **Erlang distribution** with a fixed shape parameter $k \in \mathbb{N}_{>0}$ and the rate parameter d (i.e., we aim at synthesizing an ε-optimal scale). The Erlang density function is

$$\frac{d^k \cdot \tau^{k-1}}{(k-1)!} \cdot e^{-\tau d}.$$

Similar to the Weibull distribution, we approximate $e^{-\tau d} \cdot e^{-\lambda\tau}$ using a finite prefix of a Taylor series $\sum_{n=0}^{\infty} \frac{1}{n!}(-\tau d - \lambda\tau)^n$. Then, for some sufficiently high bound $i \in \mathbb{N}_{>0}$,

$$U(\tau) \;\stackrel{\text{def}}{=}\; \frac{d^k \cdot \tau^{k-1}}{(k-1)!} \cdot P(\tau) \cdot \sum_{n=0}^{i} \frac{1}{n!} \cdot (-\tau d - \lambda\tau)^n$$

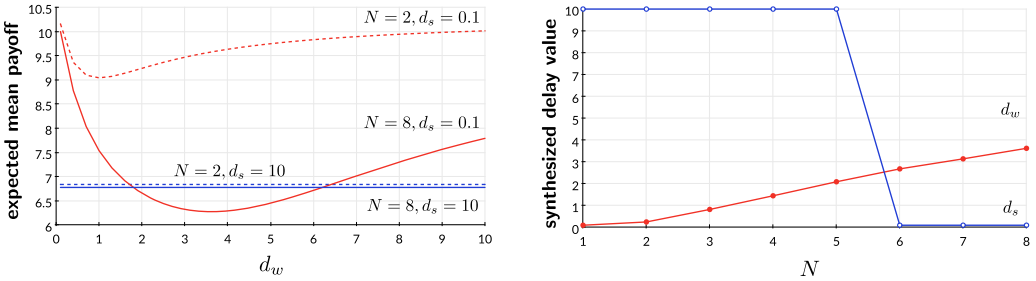

Fig. 2. Results for the disk drive example: optimal expected mean-payoff (left), and trade-off illustrated by the synthesized delay values (right).

is a polynomial function of τ and d, and hence, $\mathbf{\in}_s^\kappa \stackrel{\text{def}}{=} \int_0^{max_s} U(\tau)\,\mathrm{d}\tau$ is a polynomial function of d. Similarly, we can provide polynomial κ-approximations $\mathbf{\Pi}_s^\kappa$ and $\mathbf{\Theta}_s^\kappa$, thus Assumption 3 is fulfilled. Moreover, $\mathbf{\Pi}_s^\kappa$, $\mathbf{\Theta}_s^\kappa$, and $\mathbf{\in}_s^\kappa$ are computable for each $d \in [\ell_a, u_a] \cap \mathbb{Q}$, thus Assumption 2 also holds.

Assumption B). The proof of this assumption follows the same arguments as in Section 4.5.

5 EXPERIMENTAL EVALUATION

We demonstrate feasibility of the symbolic algorithm presented in Section 3 on the running example of Figure 1 and on a preventive maintenance model inspired by [16]. The experiments were carried out[11] using our prototype implementation of the symbolic algorithm implemented in Maple [7]. Maple is appropriate as it supports the root isolation of univariate polynomials with arbitrary high precision due to its symbolic engine. The implementation currently supports Dirac and uniform distributions only, but could be easily extended by any other distribution that fulfills Assumptions 1–4. Note that our models do not match our definition of ACTMC, since states have various exit rates. This is easy to solve by the standard uniformization method.

Disk drive model. In the running example of this article (see Section 1 and Figure 1) we aimed toward synthesizing delays d_s and d_w such that the long-run average power consumption of the disk drive is ε-optimal. Let us describe the impact of choosing delay values d_s and d_w on the expected mean-payoff in more detail. In Figure 2 (left), we illustrate the trade-off between choosing different delays d_w depending on delays $d_s \in \{0.1, 10\}$ and queue sizes $N \in \{2, 8\}$. When the queue is small, e.g., $N = 2$ (dashed curves), the expected mean-payoff is optimal for large d_s (here, $d_s = 10$). Differently, when the queue size is large, e.g., $N = 8$ (solid curves), it is better to choose small d_s (here, $d_s = 0.1$) to minimize the expected mean-payoff with d_w chosen at the minimum of the solid curve at around 3.6. This illustrates that the example is non-trivial.

The results of applying our synthesis algorithm for determining ε-optimal delays d_s and d_w depending on different queue sizes $N \in \{1, \ldots, 8\}$ with common delay bounds $\ell = 0.1$ and $u = 10$ are depicted in Figure 2 (right). From this figure, we observe that for increasing queue sizes, also the synthesized value d_w increases, whereas the optimal value for d_s is u in case $N < 6$ and ℓ otherwise.

The table in Figure 3 shows the running time of creation and solving of the Maple models, as well as the largest polynomial degrees for selected queue sizes $N = \{2, 4, 6, 8\}$ and error bounds $\varepsilon = \{0.1, 0.01, 0.001, 0.0005\}$. In all cases, discretization step sizes of $10^{-25} < \delta(\cdot) < 10^{-19}$ were required to obtain results guaranteeing ε-optimal parameter functions. These small discretization constants underpin that the ε-optimal parameter synthesis problem cannot be carried out using the explicit

[11] We used a machine equipped with Intel Core™ i7-3770 CPU processor at 3.40 GHz and 8 GiB of DDR RAM.

N	ε	creating time [s]	solving time [s]	poly degree
2	0.1	0.15	0.24	46
	0.01	0.15	0.25	46
	0.001	0.16	0.28	53
	0.0005	0.16	0.33	53
4	0.1	0.14	0.25	46
	0.01	0.16	0.25	46
	0.001	0.16	0.28	53
	0.0005	0.16	0.33	53
6	0.1	0.16	0.35	46
	0.01	0.16	0.35	46
	0.001	0.17	0.40	53
	0.0005	0.18	0.43	53
8	0.1	0.19	0.35	46
	0.01	0.19	0.35	46
	0.001	0.20	0.43	53
	0.0005	0.22	0.44	53

Fig. 3. Statistics of the symbolic algorithm applied to the disk drive example.

N	ε	creating time [s]	solving time [s]	poly degree	results
2	0.1	0.15	1.80	86	$\mathbb{E}[MP]$ 0.85524
	0.01	0.15	2.57	92	d_o 1.82752
	0.001	0.15	2.97	96	d_p 0.66167
	0.0001	0.15	3.84	101	d_q 2.05189
4	0.1	0.83	1.92	86	$\mathbb{E}[MP]$ 0.46127
	0.01	0.92	2.40	92	d_o 1.92513
	0.001	1.04	3.06	97	d_p 0.66167
	0.0001	1.04	4.24	101	d_q 2.05189
6	0.1	2.25	2.18	87	$\mathbb{E}[MP]$ 0.33060
	0.01	2.36	2.53	92	d_o 1.95764
	0.001	2.37	3.83	97	d_p 0.66167
	0.0001	2.41	4.41	101	d_q 2.05189
8	0.1	17.08	2.22	87	$\mathbb{E}[MP]$ 0.29536
	0.01	17.60	2.81	93	d_o 1.96540
	0.001	17.78	3.33	97	d_p 0.66167
	0.0001	17.87	4.48	102	d_q 2.05189

Fig. 4. Results and statistics of the symbolic algorithm applied to the preventive maintenance example.

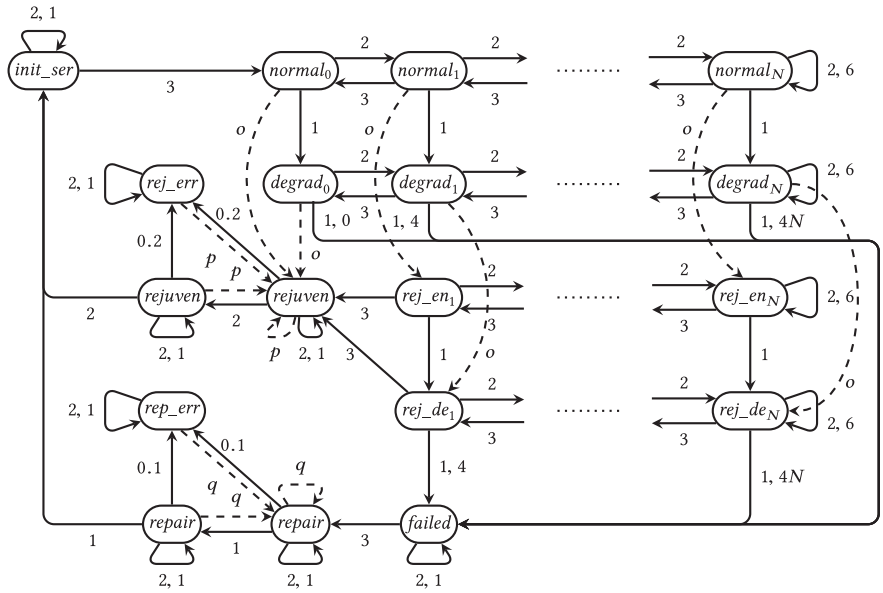

Fig. 5. Preventive maintenance of a server.

approach (our implementation of the explicit algorithm runs out of memory for all of the listed instances). However, the symbolic algorithm evaluating roots of polynomials with high degree is capable to solve the problem within seconds in all cases. This can be explained through the small number of candidate actions we had to consider (always at most 200).

Preventive maintenance. As depicted in Figure 5, we consider a slightly modified model of a server that is susceptible to software faults [16]. A *rejuvenation* is the process of performing a *preventive maintenance* of the server after a fixed period of time (usually during night time) to prevent performance degradation or even failure of the server. The first row of states in Figure 5

represents the normal behavior of the server. Jobs arrive with rate 2 and are completed with rate 3. If job arrives and queue is full, then it is rejected, which is penalized by cost 6. Degradation of server is modeled by delay transitions of rates 1 leading to *degrad* states of the second row or eventually leading to the *failed* state. The failure causes rejection of all jobs in the queue and incurs cost 4 for each rejected job. After the failure is reported (delay event with rate 3), the repair process is initiated and completed after two exponentially distributed steps of rate 1. The repair can also fail with a certain probability (rate 0.1), thus after uniformly distributed time, the repair process is restarted. After each successful repair, the server is initialized by an exponential event with rate 3. The rejuvenation procedure is enabled after staying in *normal* or *degrad* states for time d_o. Then the rejuvenation itself is initiated after all jobs in the queue are completed. The rejuvenation procedure behaves similarly as the repair process, except that it is twice as fast (all rates are doubled).

First, we want to synthesize the value of the delay after which the rejuvenation is enabled, i.e., we aim toward the optimal schedule for rejuvenation. Furthermore, we synthesize the shifts d_p and d_q of the uniform distributions with length 2 associated with rejuvenation and repair, respectively, i.e., the corresponding uniform distribution function is $F_x[d_x](\tau) = \min\{1, \max\{0, (\tau - d_x)/2\}\}$, where $x \in \{p, q\}$. The interval of eligible values is $[0.1, 10]$ for all synthesized parameters. Similar to the previous example, we show results of experiments for queue sizes $N = \{2, 4, 6, 8\}$ and error bounds $\varepsilon = \{0.1, 0.01, 0.001, 0.0001\}$ in Figure 4. The CPU time of model creation grows (almost quadratically) to the number of states, caused by multiplication of large matrices in Maple. As within the disk-drive example, we obtained the solutions very fast, since we had to consider small number of candidate actions (always at most 500).

Optimizations in the implementation. For the sake of a clean presentation in this article, we established *global* theoretical upper bounds on δ and κ sufficient to guarantee ε-optimal solutions. The theoretical bounds assume the worst underlying transition structure of a given ACTMC. In the prototype implementation, we applied some optimizations, e.g., computing different uniformization rates for each alarm $a \in A$ and its states S_a [26]. Also, to achieve better perturbation bounds on the expected mean-payoff, i.e., to compute bounds on expected time and cost to reach some state from all other states, we rely on techniques presented in [9, 24, 26]. Using these optimizations, for instance in the experiment of disk drive model, some discretization bounds δ were improved from $2.39 \cdot 10^{-239}$ to $7.03 \cdot 10^{-19}$. Note that even with these optimizations, the explicit algorithm for parameter synthesis would not be feasible as, more than 10^{18} actions would have to be considered for each state. This would clearly exceed the memory limit of state-of-the art computers.

Non-localized alarms. For our case studies, we implemented the symbolic algorithm for parametric ACTMCs with localized alarms. We are currently developing a new implementation in the prominent probabilistic model checker Prism [27] that will be more suitable for experiments with the different heuristics for ACTMCs with non-localized alarms (GBPI and MVI methods, as described in Section 3.4). According to experimental evaluation of our symbolic algorithm for parametric ACTMCs with localized alarms, we conjecture that these algorithms will be very fast. However, since they are heuristics, we need to provide more detailed performance evaluation, which is a subject of our future research.

REFERENCES

[1] H. Akimaru and K. Kawashima. 1999. *Teletraffic: Theory and Applications* (2nd ed.). Springer, London.
[2] R. Alur, T. A. Henzinger, and M. Y. Vardi. 1993. Parametric real-time reasoning. In *Proceedings of STOC*. ACM Press, New York, NY, 592–601. DOI: https://doi.org/10.1145/167088.167242

[3] E. G. Amparore, P. Buchholz, and S. Donatelli. 2014. A structured solution approach for Markov regenerative processes. In *Proceedings of QEST (LNCS)*, Vol. 8657. Springer, Cham, 9–24. DOI: https://doi.org/10.1007/978-3-319-10696-0

[4] C. Baier, C. Dubslaff, Ľ. Korenčiak, A. Kučera, and V. Řehák. 2017. Mean-payoff optimization in continuous-time Markov chains with parametric alarms. In *Proceedings of QEST (LNCS)*, Vol. 10503. Springer, Cham, 190–206. DOI: https://doi.org/10.1007/978-3-319-66335-7_12

[5] C. Baier, C. Dubslaff, Ľ. Korenčiak, A. Kučera, and V. Řehák. 2017. Mean-payoff optimization in continuous-time Markov chains with parametric alarms. Retrieved from arxiv:cs.PF/1706.06486.

[6] J. Barzilai and J. M. Borwein. 1988. Two-point step size gradient methods. *IMA J. Numer. Anal.* 8, 1 (Jan. 1988), 141–148. DOI: https://doi.org/10.1093/imanum/8.1.141

[7] L. Bernardin et al. 2012. *Maple 16 Programming Guide*. Maplesoft, Waterloo.

[8] D. P. Bertsekas and R. G. Gallager. 1992. *Data Networks* (2nd ed.). Prentice-Hall, Englewood Cliffs, NJ.

[9] T. Brázdil, Ľ. Korenčiak, J. Krčál, P. Novotný, and V. Řehák. 2015. Optimizing performance of continuous-time stochastic systems using timeout synthesis. In *Proceedings of QEST (LNCS)*, Vol. 9259. Springer, Cham, 141–159. DOI: https://doi.org/10.1007/978-3-319-22264-6

[10] T. Brázdil, J. Krčál, J. Křetínský, and V. Řehák. 2011. Fixed-delay events in generalized semi-Markov processes revisited. In *Proceedings of CONCUR (LNCS)*, Vol. 6901. Springer, Heidelberg, 140–155. DOI: https://doi.org/10.1007/978-3-642-23217-6_10

[11] X.-R. Cao and H.-T. Fang. 2002. Gradient-based policy iteration: An example. In *Proceedings of CDC*, Vol. 3. IEEE, Piscataway, NJ, 3367–3371. DOI: https://doi.org/10.1109/CDC.2002.1184395

[12] M. Češka, F. Dannenberg, M. Kwiatkowska, and N. Paoletti. 2014. Precise parameter synthesis for stochastic biochemical systems. In *Proceedings of CMSB (LNCS)*, Vol. 8859. Springer, Cham, 86–98. DOI: https://doi.org/10.1007/978-3-319-12982-2_7

[13] H. Choi, V. G. Kulkarni, and K. S. Trivedi. 1994. Markov regenerative stochastic Petri nets. *Perform. Eval.* 20, 1–3 (Jan. 1994), 337–357. DOI: https://doi.org/10.1016/0166-5316(94)90021-3

[14] L. de Alfaro. 1998. Stochastic transition systems. In *Proceedings of CONCUR (LNCS)*, Vol. 1466. Springer, Heidelberg, 423–438. DOI: https://doi.org/10.1007/BFb0055639

[15] M. Diciolla, C. H. P. Kim, M. Kwiatkowska, and A. Mereacre. 2014. Synthesising optimal timing delays for timed I/O automata. In *Proceedings of EMSOFT*. ACM Press, New York, NY. DOI: https://doi.org/10.1145/2656045.2656073

[16] R. German. 2000. *Performance Analysis of Communication Systems, Modeling with Non-Markovian Stochastic Petri Nets*. Wiley, Chichester.

[17] P. J. Haas. 2002. *Stochastic Petri Nets: Modelling, Stability, Simulation*. Springer, New York, NY.

[18] C. Haase, S. Kreutzer, J. Ouaknine, and J. Worrell. 2009. Reachability in succinct and parametric one-counter automata. In *Proceedings of CONCUR (LNCS)*, Vol. 5710. Springer, Heidelberg, 369–383. DOI: https://doi.org/10.1007/978-3-642-04081-8_25

[19] E. M. Hahn, H. Hermanns, and L. Zhang. 2011. Probabilistic reachability for parametric Markov models. *Int. J. Software Tools Technol. Transfer* 13, 1 (Jan. 2011), 3–19. DOI: https://doi.org/10.1007/s10009-010-0146-x

[20] T. Han, J.-P. Katoen, and A. Mereacre. 2008. Approximate parameter synthesis for probabilistic time-bounded reachability. In *Proceedings of RTSS*. IEEE, Piscataway, NJ, 173–182. DOI: https://doi.org/10.1109/RTSS.2008.19

[21] S. K. Jha and C. J. Langmead. 2011. Synthesis and infeasibility analysis for stochastic models of biochemical systems using statistical model checking and abstraction refinement. *Theoret. Comput. Sci.* 412, 21 (May 2011), 2162–2187. DOI: https://doi.org/10.1016/j.tcs.2011.01.012

[22] A. Jovanović and M. Kwiatkowska. 2014. Parameter synthesis for probabilistic timed automata using stochastic game abstractions. In *Proceedings of RP (LNCS)*, Vol. 8762. Springer, Cham, 176–189. DOI: https://doi.org/10.1007/978-3-319-11439-2_14

[23] A. Jovanovic, D. Lime, and O. H. Roux. 2015. Integer parameter synthesis for real-time systems. *IEEE Trans. Software Eng.* 41, 5 (May 2015), 445–461. DOI: https://doi.org/10.1109/TSE.2014.2357445

[24] Ľ. Korenčiak, A. Kučera, and V. Řehák. 2016. Efficient timeout synthesis in fixed-delay CTMC using policy iteration. In *Proceedings of MASCOTS*. IEEE, Piscataway, NJ, 367–372. DOI: https://doi.org/10.1109/MASCOTS.2016.34

[25] Ľ. Korenčiak, V. Řehák, and A. Farmadin. 2016. Extension of PRISM by synthesis of optimal timeouts in fixed-delay CTMC. In *Proceedings of iFM (LNCS)*, Vol. 9681. Springer, Cham, 130–138. DOI: https://doi.org/10.1007/978-3-319-33693-0_9

[26] Ľ. Korenčiak. 2017. *Parameter Synthesis in Continuous-Time Stochastic Systems*. Ph.D. Dissertation. Faculty of Informatics, Masaryk University, Brno.

[27] M. Kwiatkowska, G. Norman, and D. Parker. 2011. PRISM 4.0: Verification of probabilistic real-time systems. In *Proceedings of CAV (LNCS)*, Vol. 6806. Springer, Heidelberg, 585–591. DOI: https://doi.org/10.1007/978-3-642-22110-1_47

[28] Y. Li, B. Yin, and H. Xi. 2011. Finding optimal memoryless policies of POMDPs under the expected average reward criterion. *Eur. J. Operat. Res.* 211, 3 (June 2011), 556–567. DOI: https://doi.org/10.1016/j.ejor.2010.12.014

[29] C. Lindemann. 1993. An improved numerical algorithm for calculating steady-state solutions of deterministic and stochastic Petri net models. *Perform. Eval.* 18, 1 (July 1993), 79–95. DOI: https://doi.org/10.1016/0166-5316(93)90028-S

[30] M. L. Littman. 1994. Memoryless policies: Theoretical limitations and practical results. In *Simulation of Adaptive Behavior*, Vol. 3. MIT Press, Cambridge, MA, 238–245. Retrieved from http://dl.acm.org/citation.cfm?id=189829.189893

[31] S. Martina, M. Paolieri, T. Papini, and E. Vicario. 2016. Performance evaluation of Fischer's protocol through steady-state analysis of Markov regenerative processes. In *Proceedings of MASCOTS*. IEEE, Piscataway, NJ, 355–360. DOI: https://doi.org/10.1109/MASCOTS.2016.72

[32] D. L. P. Minh, D. D. L. Minh, and A. L. Nguyen. 2012. Regenerative Markov chain Monte Carlo for any distribution. *Communications in Statistics - Simulation and Computation* 41, 9 (May 2012), 1745–1760. DOI: https://doi.org/10.1080/03610918.2011.615433

[33] W. Nelson. 1985. Weibull analysis of reliability data with few or no failures. *J. Qual. Technol.* 17, 3 (July 1985), 140–146.

[34] J. R. Norris. 1998. *Markov Chains*. Cambridge University Press, New York, NY.

[35] M. L. Puterman. 1994. *Markov Decision Processes*. Wiley, Hoboken, NJ.

[36] Q. Qiu, Q. Wu, and M. Pedram. 1999. Stochastic modeling of a power-managed system: Construction and optimization. In *Proceedings of ISLPED*. IEEE, New York, NY, 194–199. DOI: https://doi.org/10.1109/LPE.1999.799438

[37] L.-M. Traonouez, D. Lime, and O. H. Roux. 2008. Parametric model-checking of time Petri nets with stopwatches using the state-class graph. In *Proceedings of FORMATS (LNCS)*, Vol. 5215. Springer, Heidelberg, 280–294. DOI: https://doi.org/10.1007/978-3-540-85778-5_20

[38] L. Yang, J. Dankert, and J. Si. 2008. A performance gradient perspective on gradient-based policy iteration and a modified value iteration. *Int. J. Innovat. Creat. Change* 1, 4 (Oct. 2008), 509–520. DOI: https://doi.org/10.1108/17563780810919096

Received January 2018; revised August 2018; accepted January 2019