Symbolic Minimum Expected Time Controller Synthesis for Probabilistic Timed Automata[*]

Aleksandra Jovanović[1], Marta Kwiatkowska[1], and Gethin Norman[2]

[1] Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK
[2] School of Computing Science, University of Glasgow, Glasgow, G12 8RZ, UK

Abstract. In this paper we consider the problem of computing the minimum expected time to reach a target and the synthesis of the corresponding optimal controller for a probabilistic timed automaton (PTA). Although this problem admits solutions that employ the digital clocks abstraction or statistical model checking, symbolic methods based on zones and priced zones fail due to the difficulty of incorporating probabilistic branching in the context of dense time. We work in a generalisation of the setting introduced by Asarin and Maler for the corresponding problem for timed automata, where simple and nice functions are introduced to ensure finiteness of the dense-time representation. We find restrictions sufficient for value iteration to converge to the minimum expected time on the uncountable Markov decision process representing the semantics of a PTA. We formulate a Bellman operator on the backwards zone graph of a PTA and prove that value iteration using this operator equals that computed over the PTA's semantics. This enables us to extract an ε-optimal controller from value iteration in the standard way.

1 Introduction

Systems which exhibit real-time, probabilistic and nondeterministic behaviour are widespread and ubiquitous in many areas such as medicine, telecommunications, robotics and transport. Timing constraints are often vital to the correctness of embedded devices and stochasticity is needed due to unreliable channels, randomisations and component failure. Finally, nondeterminism is an important concept which allows us to model and analyse systems operating in a distributed environment and/or exhibiting concurrency. A natural model for such systems, *probabilistic timed automata* (PTAs), a probabilistic extension of timed automata (TAs) [1], was proposed in [20]. They are finite-state automata equipped with real-valued clocks which measure the passage of time and whose transitions are probabilistic. Transitions are expressed as discrete probability distributions over the set of edges, namely a successor location and a set of clocks to reset.

An important class of properties on PTAs are *probabilistic reachability* properties. They allow us to check statements such as: "with probability 0.05 or less the system aborts" or "the data packet will be delivered within 1 second

[*] This research is supported by ERC AdG VERIWARE.

with minimum 0.95 probability". Model checking algorithms for these properties are well studied. Forwards reachability [20] yields only approximate probability values (upper bounds on maximum reachability probabilities). An abstraction refinement method, based on stochastic games, has subsequently been proposed in [17] for the computation of exact values and implemented in PRISM [18]. An alternative method is backward reachability [21], also giving exact values. These are all symbolic algorithms based on *zones*, a structure that represents in a concise way sets of the automaton states with equivalent behaviour.

Another important class of properties, which is the focus of this paper, is *expected reachability.* They can express statements such as "the expected number of packets sent before failure is at least 100" or "the expected time until a message is delivered is at most 20ms". These properties turned out to be more difficult to verify on PTAs and currently no symbolic approach exists. Even for TAs, the research first concentrated on checking whether there exist system behaviours that satisfy a certain property ϕ (for example, reaching the target set of states). In many situations this is not sufficient, as we often want to distinguish between behaviours that reach target states in 10 or in 1000 seconds. In [2], a backward fixed-point algorithm was proposed for controller synthesis for TAs, which generates a controller that reaches the target in minimum time. The analogous problem for priced timed automata, a model comprising more general reward (or cost) structures, was also considered. The minimum reward reachability for this model has been solved using the region graph method [4], and later extended for more efficient *priced zones* [22] and implemented in Uppaal [23].

Contributions. We propose the first zone-based algorithm to compute the minimum expected time to reach a target set and synthesise the ε-optimal controller for PTAs. The semantics of a PTA is an uncountable Markov decision process (MDP). Under suitable restrictions, we are able to prove that value iteration converges to the minimum expected time on this MDP. We formulate a Bellman operator on the backwards zone graph of a PTA and show that value iteration using this operator yields the same value as that computed on the MDP. This enables us to extract an ε-optimal controller from value iteration in the standard way. This problem has been open for several years, with previous symbolic zone-based methods, including priced zones, being unsuitable for computing expected values since accumulated rewards are *unbounded.* In order to represent the value functions we introduce rational k-simple and rational k-nice functions, a generalisation of Asarin and Maler's classes of functions [2].

Related work. Expected reachability properties of PTAs can be verified using the *digital clocks* method [19], which assumes an integral model of time as opposed to a dense model of time. This method, however, suffers from state-space explosion. In [12], the minimum expected reward for priced timed games has been solved using *statistical model checking* and Uppaal-SMC [11]. This is orthogonal to numerical model checking, based on simulation and hypothesis testing, giving only approximate results which are not guaranteed to be correct.

In [7] the authors consider priced probabilistic timed automata and study reward-bounded probabilistic reachability, which determines whether the maximal probability to reach a set of target locations, within given bounds on the accumulated reward and elapsed time, exceeds a threshold. Although this problem is shown to be undecidable [6], a semi-decidable backwards algorithm using priced zones, which terminates if the problem is affirmative, is implemented in FORTUNA [8].

Outline. In Section 2 we define MDPs and give existing results concerning optimal reward computation for uncountable MDPs. Section 3 defines PTAs and introduces the assumptions needed for the adoption of the results of Section 2. In Section 4, we present our algorithm for computing the minimum expected time and synthesis of an ε-optimal controller using the backwards zone graph of a PTA. Section 4 also introduces a representation of the value functions that generalise the simple and rational nice functions of [2] and gives an example demonstrating the approach. We conclude with Section 5.

An extended version of this paper, with proofs, is available as [15].

2 Background

Let $\mathbb{R}$ be the set of non-negative reals, $\mathbb{N}$ the integers, $\mathbb{Q}$ the rationals and $\mathbb{Q}_+$ the non-negative rationals. A discrete probability distribution over a set S is a function $\mu : S{\to}[0,1]$ such that $\sum_{s \in S} \mu(s) = 1$ and the set $\{s \in S \mid \mu(s){>}0\}$ is finite. We denote by $\mathsf{Dist}(S)$ the set of distributions over S.

Markov Decision Processes (MDPs) is a widely used formalism for modelling systems which exhibit both nondeterministic and probabilistic behaviour.

Definition 1. *An MDP is a tuple $\mathcal{M} = (S, s_0, A, P_\mathcal{M}, R_\mathcal{M})$, where S is a (possibly uncountable) set of states, $s_0 \in S$ is an initial state, A is a (possibly uncountable) set of actions, $P_\mathcal{M} : (S{\times}A) \to \mathsf{Dist}(S)$ is a (partial) probabilistic transition function and $R_\mathcal{M} : (S{\times}A) \to \mathbb{R}$ is a reward function.*

A state s of an MDP $\mathcal{M}$ has a set of enabled actions, denoted $A(s)$, given by the set of actions for which $P_\mathcal{M}(s, a)$ is defined. A transition in $\mathcal{M}$ from state s is first made by nondeterministically selecting an available action $a \in A(s)$. After the choice is made, a successor state s' is selected randomly according to the probability distribution $P_\mathcal{M}(s, a)$, i.e. the probability that a transition to s' occurs is equal to $P_\mathcal{M}(s, a)(s')$, and the reward $R_\mathcal{M}(s, a)$ is accumulated when making this transition.

An infinite *path* is a sequence $\omega = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots$ of transitions such that $P_\mathcal{M}(s_i, a)(s_{i+1}){>}0$ for all $i{\geqslant}0$, and it represents a particular resolution of both nondeterminism and probability. A finite path is a prefix of an infinite path ending in a state. The $(i{+}1)$th state of a path ω is denoted by $\omega(i)$ and the action associated with the $(i{+}1)$th transition by $\mathit{step}(\omega, i)$. We denote the set of all infinite (finite) paths of $\mathcal{M}$ by $\mathit{IPaths}_\mathcal{M}$ ($\mathit{FPaths}_\mathcal{M}$) and the last state of a finite path ω by $\mathit{last}(\omega)$.

A *strategy* (also called an *adversary* or *policy*) of $\mathcal{M}$ resolves the choice between actions in each state, based on the execution so far.

Definition 2. *A strategy of an MDP $\mathcal{M}$ is a function $\sigma : \mathit{FPaths}_{\mathcal{M}} {\rightarrow} \mathsf{Dist}(A)$ such that $\sigma(\omega)(a){>}0$ only if $a \in A(\mathit{last}(\omega))$.*

For a fixed strategy σ and state s, we can define a probability measure $\mathcal{P}^{\sigma}_{s}$ over the set of infinite paths starting in s [16]. A strategy σ is memoryless if its choices only depend on the current state, and deterministic if $\sigma(\omega)$ is a point distribution for all $\omega \in \mathit{FPaths}_{\mathcal{M}}$. The set of strategies of $\mathcal{M}$ is denoted by $\Sigma_{\mathcal{M}}$.

Two fundamental quantitative properties of MDPs are the probability of reaching a set of target states and the expected reward accumulated before reaching a target. For a strategy σ, state s and set of target states F, the probability of reaching F and expected reward accumulated before reaching F from s under σ are given by:

$$\begin{aligned} \mathbb{P}^{\sigma}_{\mathcal{M}}(s,F) &\overset{\text{def}}{=} \mathcal{P}^{\sigma}_{s}\{\omega \in \mathit{IPaths}_{\mathcal{M}} \mid \omega(i) \in F \text{ for some } i \in \mathbb{N}\} \\ \mathbb{E}^{\sigma}_{\mathcal{M}}(s,F) &\overset{\text{def}}{=} \textstyle\int_{\omega \in \mathit{IPaths}_{\mathcal{M}}} \mathit{rew}(\omega,F)\, \mathrm{d}\mathcal{P}^{\sigma}_{s} \end{aligned}$$

where for any infinite path ω:

$$\mathit{rew}(\omega,F) \overset{\text{def}}{=} \begin{cases} \sum_{i=0}^{\min\{k-1 \mid \omega(k) \in F\}} R_{\mathcal{M}}(\omega(i), \mathit{step}(\omega,i)) & \text{if } \omega(k) \in F \text{ for some } k \in \mathbb{N} \\ \infty & \text{otherwise.} \end{cases}$$

The standard approach is to analyse the optimal values of these measures, i.e. the minimum and maximum values over all strategies. In this paper, we are concerned with the maximum probability of reaching a target and minimum expected accumulated reward before reaching a target:

$$\mathbb{P}^{\max}_{\mathcal{M}}(s,F) \overset{\text{def}}{=} \sup_{\sigma \in \Sigma_{\mathcal{M}}} \mathbb{P}^{\sigma}_{\mathcal{M}}(s,F) \quad \text{and} \quad \mathbb{E}^{\min}_{\mathcal{M}}(s,F) \overset{\text{def}}{=} \inf_{\sigma \in \Sigma_{\mathcal{M}}} \mathbb{E}^{\sigma}_{\mathcal{M}}(s,F)\,.$$

The optimal values can be computed using a *Bellman operator* [5]. More precisely, under certain conditions on the MDP and target set under study, using a Bellman operator the optimal values can be obtained through a number of techniques, including *value iteration* and *policy iteration*, see for example [10,9]. Concerning minimum expected reachability we have the following definition.

Definition 3. *Given an MDP $\mathcal{M}$ and target set F, the Bellman operator $T_{\mathcal{M}} : (S{\rightarrow}\mathbb{R}) \rightarrow (S{\rightarrow}\mathbb{R})$ for minimum expected reachability is defined as follows. For any $f : S \rightarrow \mathbb{R}$ and $s \in S$:*

$$T_{\mathcal{M}}(f)(s) = \begin{cases} 0 & \textit{if } s \in F \\ \inf_{a \in A(s)} \{R_{\mathcal{M}}(s,a) + \sum_{s' \in S} P_{\mathcal{M}}(s,a)(s') {\cdot} f(s')\} & \textit{otherwise.} \end{cases}$$

Value iteration for $T_{\mathcal{M}}$ corresponds to repeatedly applying the operator when starting from some initial approximation f_0 until some convergence criterion is met, e.g. computing $T^{n+1}(f_0){=}T(T^n(f_0))$ until $\|T^{n+1}(f_0){-}T^n(f_0)\| \leqslant \varepsilon$ for some threshold ε. On the other hand, policy iteration starts with an arbitrary, deterministic and memoryless strategy, and then tries repeatedly to construct an improved (deterministic and memoryless) strategy. This is achieved by computing the expected reachability values for the current strategy and, if possible, updating the actions choices so that the expected reachability values decrease.

We now adapt the results of [14] for the total expected reward for possibly uncountable-state and uncountable-action set MDPs. The conditions imposed by [14] correspond, in our setting, to those given below (since we restrict to discrete distributions and non-negative reward values, the assumptions we require are weaker).

Assumption 1. *For an MDP $\mathcal{M}$ and target set F:*

(a) *$A(s)$ is compact for all $s \in S$;*
(b) *$R_{\mathcal{M}}$ is bounded and $a \mapsto R_{\mathcal{M}}(s, a)$ is lower semi-continuous for all $s \in S$;*
(c) *if σ is a memoryless, deterministic strategy which is not proper, then $\mathbb{E}^{\sigma}_{\mathcal{M}}(s, F)$ is unbounded for some $s \in S$;*
(d) *there exists a proper, memoryless, deterministic strategy;*

where a strategy σ is called proper if $\mathbb{P}^{\sigma}_{\mathcal{M}}(s, F){=}1$ for all $s \in S$.

Using these assumptions we have the following result.

Theorem 1 ([14]). *If $\mathcal{M}$ and F are an MDP and target set for which Assumption 1 holds, then:*

- *there exists a memoryless, deterministic strategy that achieves the minimum expected reward of reaching F;*
- *the minimum expected reward values are the unique solutions to $T_{\mathcal{M}}$;*
- *value iteration over $T_{\mathcal{M}}$ converges to the minimum expected reward values when starting from any bounded function;*
- *policy iteration converges to the minimum expected reward values when starting from any proper, memoryless, deterministic strategy.*

3 Probabilistic Timed Automata

We now introduce Probabilistic Timed Automata, a modelling framework for systems which incorporate probabilistic, nondeterministic and real-time behaviour.

Clocks, Clock Valuations and Zones. Let $\mathcal{X}$ be a set of real-valued variables called clocks, which increase at the same, constant rate. A function $v : \mathcal{X} \to \mathbb{R}$ is called clock valuation function and the set of all clock valuations is denoted as $\mathbb{R}^{\mathcal{X}}$. Let $\mathbf{0}$ be a valuation that assigns 0 to all clocks in $\mathcal{X}$. For any $R \subseteq \mathcal{X}$ and any valuation v on $\mathcal{X}$, we write $v[R]$ for the valuation on $\mathcal{X}$ such that $v[R](x){=}0$ if $x \in R$ and $v[R](x){=}v(x)$ otherwise. For $t \in \mathbb{R}$, $v{+}t$ denotes the valuation which assigns $(v{+}t)(x){=}v(x){+}t$ to all $x \in \mathcal{X}$. A zone is an expression of the form: $\zeta := x{\sim}c \mid x{-}y{\sim}c \mid \zeta{\wedge}\zeta$, where $x, y \in \mathcal{X}$, $\sim \in \{<, \leqslant, >, \geqslant\}$ and $c \in \mathbb{N}$. The set of zones on $\mathcal{X}$ is denoted $\mathit{Zones}(\mathcal{X})$. A clock valuation v satisfies a zone ζ, denoted $v{\models}\zeta$, if ζ resolves to true after substituting each occurrence of clock x with $v(x)$. A zone ζ represents the set of clock valuations v which satisfy it.

We require a number of classical operations on zones [24]. Zone $\nearrow\zeta$ contains all valuations reachable from a valuation in ζ by letting time pass. Conversely, $\swarrow\zeta$ contains all valuations that can reach ζ by letting time pass. Furthermore, for a set of clocks R, $\zeta[R]$ includes the valuations obtained by those in ζ by resetting the clocks R and $[R]\zeta$ the valuations which result in a valuation in ζ when the clocks in R are reset to 0.

Definition 4. *A PTA $\mathcal{P}$ is a tuple $(L, l_0, \mathcal{X}, Act, \mathsf{enab}, \mathsf{prob}, \mathsf{inv})$ where: L is a finite set of locations; $l_0 \in L$ is an initial location; $\mathcal{X}$ is a finite set of clocks; Act is a finite set of actions; $\mathsf{enab} : (L \times Act) \to Zones(\mathcal{X})$ is an enabling condition; $\mathsf{prob} : (L \times Act) \to \mathsf{Dist}(2^{\mathcal{X}} \times L)$ is a probabilistic transition function; $\mathsf{inv} : L \to Zones(\mathcal{X})$ is an invariant condition.*

A state of $\mathcal{P}$ is a pair $(l, v) \in L \times \mathbb{R}^{\mathcal{X}}$ such that the clock valuation v satisfies the invariant $\mathsf{inv}(l)$. A transition is a time-action pair (t, a) corresponding to letting time t elapse and then performing the action a. In a state (l, v), time can elapse as long as the invariant $\mathsf{inv}(l)$ remains continuously satisfied and action a can be performed only if the enabling condition $\mathsf{enab}(l, a)$ is then satisfied. If transition (t, a) is performed, then the set of clocks to reset and successor location are selected randomly according to the probability distribution $\mathsf{prob}(l, a)$.

For $(l, a) \in L \times Act$, an element $(R, l') \in 2^{\mathcal{X}} \times L$ such that $\mathsf{prob}(l, a)(R, l') > 0$ is called an *edge* of (l, a) and the set of all edges of (l, a) is denoted $\mathsf{edges}(l, a)$.

Definition 5. *For PTA $\mathcal{P} = (L, l_0, \mathcal{X}, Act, \mathsf{prob}, \mathsf{inv})$ its semantics is given by the (infinite-state) MDP $[\![\mathcal{P}]\!] = (S, s_0, \mathbb{R} \times Act, P_{[\![\mathcal{P}]\!]}, R_{[\![\mathcal{P}]\!]})$ where:*

- *$S = \{(l, v) \in L \times \mathbb{R}^{\mathcal{X}} \mid v \models \mathsf{inv}(l)\}$ and $s_0 = (l_0, \mathbf{0})$;*
- *$P_{[\![\mathcal{P}]\!]}((l, v), (t, a)) = \mu$ if and only if $v+t' \models \mathsf{inv}(l)$ for all $0 \leqslant t' \leqslant t$, $v+t \models \mathsf{enab}(l, a)$ and for any $(l', v') \in S$:*

$$\mu(l', v') = \sum \{\!|\, \mathsf{prob}(l, a)(R, l') \mid R \subseteq \mathcal{X} \wedge v' = (v+t)[R] \,|\!\}$$

- *$R_{[\![\mathcal{P}]\!]}(t, a) = t$ for all $(t, a) \in \mathbb{R} \times Act$.*

For Theorem 1 to be applicable to semantics of a PTA, we need to ensure Assumption 1 holds. To this end, we introduce the following assumptions.

Assumption 2. *For any PTA $\mathcal{P}$ we have:*

(a) *all invariants and enabling conditions of $\mathcal{P}$ are bounded;*
(b) *only non-strict inequalities are allowed in clock constraints ($\mathcal{P}$ is closed);*
(c) *$\mathcal{P}$ is structurally non-zeno [25] (this can be identified syntactically and in a compositional fashion [26] and guarantees time-divergent behaviour).*

Conditions (a) and (b) are necessary and sufficient to ensure $A(s)$ is compact for all states $s \in S$, i.e. Assumption 1(a) holds. Assumption 1(b) follows from Definition 5 as, for any $(t, a) \in \mathbb{R} \times Act$, we have $R_{[\![\mathcal{P}]\!]}(s, (t, a)) = t$ for all $s \in S$. Structurally non-zeno is sufficient for ensuring Assumption 1(c) holds. More precisely, if for strategy σ the probability of reaching the target is less than

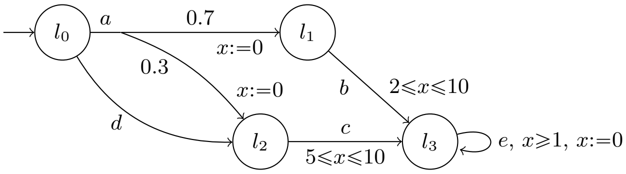

Fig. 1. PTA example

1, there is a non-negligible set of paths under σ which never reach the target and, since σ is non-zeno, elapsed time (and hence the accumulated reward) must diverge on the paths in this set.

The remaining assumption, Assumption 1(d), holds if we restrict attention to the sub-MDP of $[\![\mathcal{P}]\!]$ which contains only states s for which $\mathbb{P}^{\max}_{[\![\mathcal{P}]\!]}(s,F){=}1$ [13]. More precisely, if $\mathbb{P}^{\max}_{[\![\mathcal{P}]\!]}(s,F){=}1$, then, using the region graph construction [20], there exists a memoryless, deterministic strategy that reaches the target with probability 1, and hence this strategy will also be proper.

We have imposed several restrictions on PTAs we analyse. First, boundedness is not actually a restriction since bounded TAs are as expressive as standard TAs [4] and the result carries over to PTAs. The fact that PTAs must be closed is not a severe restriction in practice, as any PTA can be infinitesimally approximated by one with closed constraints. Non-zenoness is a standard assumption for both TAs and PTAs, as it discards unrealistic behaviours, i.e. executions for which time does not diverge.

Example 1. Consider the PTA shown in Figure 1 where the target is l_3. We assume the invariant in each location equals $x{\leqslant}10$ and the enabling conditions for transitions labelled a and d equal $x{\leqslant}10$. From the state (l_0,v), if action a is chosen, then the minimum expected time equals $0.3{\cdot}5+0.7{\cdot}2 = 2.9$. On the other hand, if action d is selected, then the minimum expected time equals $5{-}v(x)$ if $v(x){\leqslant}5$ and 0 otherwise. Therefore, in the initial state, i.e. when $v(x){=}0$, the minimum expected time equals $\min\{2.9, 5{-}0\} = 2.9$.

In this example, the optimal choices are to take transitions as soon as they are available. However, as we will see, this does not hold in general since we might need to wait longer in a location in order for an enabling condition to be satisfied later.

4 Minimum Expected Time Algorithm for PTAs

In this section we present our algorithm for the minimum expected time computation for PTAs. It is based on a backwards exploration of the state space. We adopt backwards as opposed to forwards search since, although forwards has proven successful in the context of TAs, for PTAs it yields only upper bounds for maximum probabilistic reachability [20]. For the remainder of the section we fix a PTA $\mathcal{P} = (L, l_0, \mathcal{X}, Act, \mathsf{enab}, \mathsf{prob}, \mathsf{inv})$, target set of locations F and suppose $[\![\mathcal{P}]\!] = (S, S_0, \mathbb{R}{\times}Act, P_{[\![\mathcal{P}]\!]}, R_{[\![\mathcal{P}]\!]})$ and $S_F = \{(l,v) \mid l \in F \wedge v{\models}\mathsf{inv}(l)\}$.

```
BackwardsReach($\mathcal{P}, F$)
$\mathtt{Z} := \varnothing$
$\mathtt{E} := \varnothing$
$\mathtt{Y} := \{(l, \mathsf{inv}(l)) \mid l \in F\}$
while ($\mathtt{Y} \neq \varnothing$)
  choose $\mathtt{y} \in \mathtt{Y}$
  $\mathtt{Y} := \mathtt{Y} \setminus \{\mathtt{y}\}$
  $\mathtt{Z} := \mathtt{Z} \cup \{\mathtt{y}\}$
  for $(l, a) \in (L \setminus F) \times Act$ and $(R, l') \in \mathsf{edges}(l, a)$
    $\mathtt{z} := \mathsf{dpre}(l, a, R, l')(\mathsf{tpre}(\mathtt{y}))$
    if ($\mathtt{z} \neq \varnothing$)
    if ($\mathtt{z} \notin \mathtt{Z}$) then $\mathtt{Y} := \mathtt{Y} \cup \{\mathtt{z}\}$
      $\mathtt{E} := \mathtt{E} \cup \{(\mathtt{z}, a, (R, l'), \mathtt{y})\}$
      for $(\tilde{\mathtt{z}}, a, (\tilde{R}, \tilde{l}'), \tilde{\mathtt{y}}) \in \mathtt{E}$ such that $(\tilde{R}, \tilde{l}') \neq (R, l')$
        if ($\mathtt{z} \wedge \tilde{\mathtt{z}} \neq \varnothing$) and $\mathtt{z} \wedge \tilde{\mathtt{z}} \notin \mathtt{Z}$ then $\mathtt{Y} := \mathtt{Y} \cup \{\mathtt{z} \wedge \tilde{\mathtt{z}}\}$
for $\mathtt{z} \in \mathtt{Z}$ and $(\mathtt{z}', a, (R, l'), \mathtt{z}'') \in \mathtt{E}$ do
   if $\mathtt{z} \subseteq \mathtt{z}'$ then
     $\mathtt{E} := \{(\mathtt{z}, a, (R, l'), \mathtt{z}'')\} \cup \mathtt{E}$
return $(\mathtt{Z}, \mathtt{E})$
```

Fig. 2. Backward reachability algorithm

Symbolic States. A symbolic state $\mathtt{z}$ of $\mathcal{P}$ is a pair $(l, \zeta) \in L \times Zones(\mathcal{X})$ representing the set of PTA states $\{(l, v) \mid v \models \zeta\}$. Let $\mathtt{Z}_F = \{(l, \mathsf{inv}(l)) \mid l \in F\}$, i.e. the target set of symbolic states. For any symbolic states $\mathtt{z} = (l, \zeta)$ and $\mathtt{z}' = (l, \zeta')$ let $\mathtt{z} \wedge \mathtt{z}' = (l, \zeta \wedge \zeta')$, $\mathtt{z} \subseteq \mathtt{z}'$ if and only if $\zeta \subseteq \zeta'$ and $\mathtt{z} = \varnothing$ if and only if $\zeta = \mathtt{false}$. The time and discrete predecessor operations for $\mathtt{z} = (l, \zeta)$ are defined as follows:

$$\mathsf{tpre}(\mathtt{z}) = (l, \swarrow\zeta \wedge \mathsf{inv}(l))$$

$$\mathsf{dpre}(l'', a, (R, l'))(\mathtt{z}) = \begin{cases} (l'', \mathtt{false}) & \text{if } l \neq l' \\ (l'', [R]\zeta \wedge \mathsf{enab}(l'', a)) & \text{otherwise} \end{cases}$$

where $(R, l') \in \mathsf{edges}(l'', a)$, $l'' \in L$ and $a \in Act$.

Backward Reachability Algorithm. We use a slightly modified version of the backward reachability algorithm on symbolic states taken from [21] (the same operations are performed, we just add action labels to the edge tuples). The modified version is given in Figure 2.

The backwards algorithm returns a zone graph $(\mathtt{Z}, \mathtt{E})$ with symbolic states as vertices. Termination of the algorithm is guaranteed by the fact that only finitely many zones can be generated. As demonstrated in [21], from this graph one can build a finite state MDP for computing the exact maximum reachability probabilities of $[\![\mathcal{P}]\!]$. The MDP $\mathcal{M}_{(\mathtt{Z},\mathtt{E})}$ has state space $\mathtt{Z}$, action set $2^{\mathtt{E}}$ and if $\mathtt{z} \in \mathtt{Z}$ and $E \in 2^{\mathtt{E}}$, then $P_{\mathcal{M}_{(\mathtt{Z},\mathtt{E})}}(\mathtt{z}, E)$ is defined if and only if there exists $a \in Act$ such that

- $(\mathtt{z}'', a', (R, l'), \mathtt{z}') \in E$ implies $\mathtt{z}'' = \mathtt{z}$ and $a' = a$;
- $(\mathtt{z}, a, (R, l'), \mathtt{z}') \neq (\mathtt{z}, a, (\tilde{R}, \tilde{l}'), \tilde{\mathtt{z}}') \in E$ implies $(R, l') \neq (\tilde{R}, \tilde{l}')$;

where $P_{\mathcal{M}_{(\mathtt{Z},\mathtt{E})}}(\mathsf{z},E)(\mathsf{z}')=\sum\{\!|\mathsf{prob}(l,a)(R,l') \mid (\mathsf{z},a,(R,l'),\mathsf{z}') \in \mathtt{E}|\!\}$ for $\mathsf{z}' \in \mathtt{Z}$.

The following theorem shows the correspondence between the maximum reachability probabilities for $[\![\mathcal{P}]\!]$ and $\mathcal{M}_{(\mathtt{Z},\mathtt{E})}$.

Theorem 2 ([21]). *Let* $(\mathtt{Z},\mathtt{E})$ *be the zone graph returned by* $\mathsf{BackwardsReach}(\mathcal{P},F)$*, then for any state* s *of* $[\![\mathcal{P}]\!]$ *we have:*

- $\mathbb{P}^{\max}_{[\![\mathcal{P}]\!]}(s,S_F)>0$ *if and only if there exists* $\mathsf{z} \in \mathtt{Z}$ *such that* $s \in \mathsf{tpre}(\mathsf{z})$*;*
- *if* $\mathbb{P}^{\max}_{[\![\mathcal{P}]\!]}(s,S_F)>0$*, then* $\mathbb{P}^{\max}_{[\![\mathcal{P}]\!]}(s,S_F) = \max\left\{ \mathbb{P}^{\max}_{\mathcal{M}_{(\mathtt{Z},\mathtt{E})}}(\mathsf{z},\mathtt{Z}_F) \mid \mathsf{z}\in\mathtt{Z} \wedge s\in\mathsf{tpre}(\mathsf{z}) \right\}$.

Using Theorem 2 we can find the states s of $[\![\mathcal{P}]\!]$ for which $\mathbb{P}^{\max}_{[\![\mathcal{P}]\!]}(s,S_F)=1$ by computing the symbolic states z for which $\mathbb{P}^{\max}_{\mathcal{M}_{(\mathtt{Z},\mathtt{E})}}(\mathsf{z},\mathtt{Z}_F)=1$. Finding these symbolic states does not require numerical computation [13], and hence we do not need to build $\mathcal{M}_{(\mathtt{Z},\mathtt{E})}$, but can use $(\mathtt{Z},\mathtt{E})$ directly in the computation.

For the remainder of this section we assume we have computed the states of $\mathcal{M}_{(\mathtt{Z},\mathtt{E})}$, and hence of $[\![\mathcal{P}]\!]$, for which the maximum reachability probability is 1, and $[\![\mathcal{P}]\!]$ and $(\mathtt{Z},\mathtt{E})$ are the sub-MDP and sub-graph restricted to these states. Using Theorem 2, $s \in S$ if and only if there exists $\mathsf{z} \in \mathtt{Z}$ such that $s \in \mathsf{tpre}(\mathsf{z})$.

For states not considered, i.e. states for which the maximum reachability probability is less than 1, since we assume $\mathcal{P}$ is non-zeno (Assumption 2(c)) their minimum expected time equals infinity. Therefore, if we compute the minimum expected time for the states of the constructed sub-MDP, we will have found the minimum expected time for all states of the PTA.

Following the discussion in Section 3, $[\![\mathcal{P}]\!]$ now satisfies Assumption 1 and therefore we can use Theorem 1. In particular, value iteration for the Bellman operator of Definition 3 for $[\![\mathcal{P}]\!]$ and S_F converges to the minimum expected time when starting from any bounded function. Below we will present a value iteration method over $(\mathtt{Z},\mathtt{E})$ and prove that it corresponds to that for $[\![\mathcal{P}]\!]$ and S_F, and hence will also converge to the minimum expected time values for $[\![\mathcal{P}]\!]$.

Value iteration over the zone graph. To present the value iteration operator for $(\mathtt{Z},\mathtt{E})$, we require the following notation. For $(l,\zeta) \in \mathtt{Z}$, the set of edges $E \subseteq \mathtt{E}$ is an element of $\mathtt{E}(l,\zeta)$ if and only if there exists $a \in Act$ such that $\mathsf{edges}(l,a) = \{(R_1,l_1),\dots,(R_n,l_n)\}$ and $E=\{(\mathsf{z},a,(R_1,l_1),\mathsf{z}_1),\dots,(\mathsf{z},a,(R_n,l_n),\mathsf{z}_n)\}$ for some $\mathsf{z}_1,\dots,\mathsf{z}_n \in \mathtt{Z}$.

Definition 6. *The operator* $T_{(\mathtt{Z},\mathtt{E})} : (\mathtt{Z}\rightarrow(S\rightarrow\mathbb{R}))\rightarrow(\mathtt{Z}\rightarrow(S\rightarrow\mathbb{R}))$ *on the zone graph* $(\mathtt{Z},\mathtt{E})$ *is such that for* $g : \mathtt{Z}\rightarrow(S\rightarrow\mathbb{R})$*,* $(l,\zeta) \in \mathtt{Z}$ *and* $(l,v) \in S$ *where* $(l,v) \in \mathsf{tpre}(l,\zeta)$ *we have* $T_{(\mathtt{Z},\mathtt{E})}(g)(l,\zeta)(l,v)$ *equals 0 if* $l \in F$ *and otherwise equals*

$$\inf_{t\in\mathbb{R} \wedge v+t\in\zeta} \; \min_{E\in\mathtt{E}(l,\zeta)} \left\{ t + \sum_{((l,\zeta),a,(R,l'),(l',\zeta'))\in E} \mathsf{prob}(l,a)(R,l')\cdot g(l',\zeta')(l',(v+t)[R]) \right\} .$$

We now demonstrate the correspondence between value iteration using this operator over $(\mathtt{Z},\mathtt{E})$ and that given by Definition 3 over $[\![\mathcal{P}]\!]$.

Proposition 1. *If* $f : S\rightarrow\mathbb{R}$ *and* $g : \mathtt{Z}\rightarrow(S\rightarrow\mathbb{R})$ *are functions such that* $f(s) = g(\mathsf{z})(s)$ *for all* $\mathsf{z} \in \mathtt{Z}$ *and* $s \in \mathsf{tpre}(\mathsf{z})$*, then for any* $s \in S$ *and* $n \in \mathbb{N}$ *we have:* $T^n_{[\![\mathcal{P}]\!]}(f)(s) = \min\{ T^n_{(\mathtt{Z},\mathtt{E})}(g)(\mathsf{z})(s) \mid \mathsf{z} \in \mathtt{Z} \wedge s \in \mathsf{tpre}(\mathsf{z}) \}$.

Proof. Consider any $f : S \to \mathbb{R}$ and $g : \mathtt{Z} \to (S \to \mathbb{R})$ such that $f(s) = g(\mathtt{z})(s)$ for all $\mathtt{z} \in \mathtt{Z}$ and $s \in \mathtt{z}$. The proof is by induction on $n \in \mathbb{N}$. If $n{=}0$, then the result follows by construction of f and g and since $T^0_{\llbracket \mathcal{P} \rrbracket}(f) = f$ and $T^0_{(\mathtt{Z},\mathtt{E})}(g) = g$.

Next we assume the proposition holds for some $n \in \mathbb{N}$. For any $s{=}(l, v) \in S$, if $l \in F$, then by the construction of the zone graph (see Figure 2), Definition 3 and Definition 6 we have: $T^{n+1}_{\llbracket \mathcal{P} \rrbracket}(f)(s) = 0 = \min\big\{T^{n+1}_{(\mathtt{Z},\mathtt{E})}(g)(\mathtt{z})(s) \mid \mathtt{z} \in \mathtt{Z} \wedge s \in \mathsf{tpre}(\mathtt{z})\big\}$.

It therefore remains to consider the case when $s{=}(l, v) \in S$ and $l \notin F$. For any $(t', a') \in A(s)$ and $(R, l') \in \mathsf{edges}(l, a)$ by the induction hypothesis there exists $(l', \zeta_{(R,l')}) \in \mathtt{Z}$ with $(l', (v{+}t')[R]) \in \mathsf{tpre}(l', \zeta_{(R,l')})$ such that:

$$T^n_{(\mathtt{Z},\mathtt{E})}(g)(l', \zeta_{(R,l')})(l', (v{+}t')[R]) = T^n_{\llbracket \mathcal{P} \rrbracket}(f)(l', (v{+}t')[R])\,. \tag{1}$$

Now since $(t', a') \in A(s)$ and $(l', (v{+}t')[R]) \in \mathsf{tpre}(l', \zeta_{(R,l')})$ it follows from Definition 5 that $(l, v{+}t) \in \mathsf{dpre}(l, a', (R, l'))(\mathsf{tpre}(l', \zeta_{(R,l')}))$.

Since the edge $(R, l') \in \mathsf{edges}(l, a)$ was arbitrary, by the construction of the zone graph (see Figure 2), there exists $(l, \zeta) \in \mathtt{Z}$ such that $v{+}t' \in \zeta$ and edge set:

$$E' = \{((l, \zeta), a', (R, l'), (l', \zeta_{(R,l')})) \mid (R, l') \in \mathsf{edges}(l, a)\} \in \mathtt{E}(l, \zeta)\,. \tag{2}$$

Furthermore, by definition of tpre we have $(l, v) \in \mathsf{tpre}(l, \zeta)$. Now, by Definition 6, $T^{n+1}_{(\mathtt{Z},\mathtt{E})}(g)(l, \zeta)(l, v)$ equals:

$$\begin{aligned}
&\inf_{t \in \mathbb{R} \wedge v+t \in \zeta} \; \min_{E \in \mathtt{E}(l,\zeta)} \left\{ t + \sum_{((l,\zeta),a,(R,l'),(l',\zeta')) \in E} \mathsf{prob}(l, a)(R, l') \cdot T^n_{(\mathtt{Z},\mathtt{E})}(g)(l', \zeta')(l', (v{+}t)[R]) \right\} \\
&\leqslant \min_{E \in \mathtt{E}(l,\zeta)} \left\{ t' + \sum_{((l,\zeta),a',(R,l'),(l',\zeta')) \in E} \mathsf{prob}(l, a')(R, l') \cdot T^n_{(\mathtt{Z},\mathtt{E})}(g)(l', \zeta')(l', (v{+}t')[R]) \right\} \\
&\hfill \text{(since } v{+}t' \in \zeta\text{)} \\
&\leqslant t' + \sum_{((l,\zeta),a,(R,l'),(l',\zeta')) \in E'} \mathsf{prob}(l, a')(R, l') \cdot T^n_{(\mathtt{Z},\mathtt{E})}(g)(l', \zeta')(l', (v{+}t')[R]) \\
&\hfill \text{(since } E' \in \mathtt{E}(l, \zeta)\text{)} \\
&= t' + \sum_{(R,l') \in \mathsf{edges}(l,a')} \mathsf{prob}(l, a')(R, l') \cdot T^n_{\llbracket \mathcal{P} \rrbracket}(f)(l', (v{+}t')[R]) \qquad \text{(by (1) and (2))} \\
&= R_{\llbracket \mathcal{P} \rrbracket}(s, (t', a')) + \textstyle\sum_{s' \in S} P_{\llbracket \mathcal{P} \rrbracket}(s, (t', a'))(s') \cdot T^n_{\llbracket \mathcal{P} \rrbracket}(f)(s') \qquad \text{(by Definition 5)}
\end{aligned}$$

Therefore, since $(t', a') \in A(s)$ was arbitrary it follows from Definition 3 that:

$$T^{n+1}_{\llbracket \mathcal{P} \rrbracket}(f)(s) \geqslant \min\left\{\, T^{n+1}_{(\mathtt{Z},\mathtt{E})}(g)(\mathtt{z})(s) \mid \mathtt{z} \in \mathtt{Z} \wedge s \in \mathsf{tpre}(\mathtt{z}) \,\right\}. \tag{3}$$

Next we consider any $\mathtt{z}{=}(l, \zeta) \in \mathtt{Z}$ such that $v{+}t \in \zeta$ for some $t \in \mathbb{R}$ (i.e. $\mathtt{z} \in \mathtt{Z}$ such that $s \in \mathsf{tpre}(\mathtt{z})$). For any $t' \in \mathbb{R}$ such that $v{+}t' \in \zeta$ and $E' \in \mathtt{E}(l, \zeta)$ by construction of the zone graph there exists $a' \in Act$ where:

$$E' = \{((l, \zeta), a', (R, l'), (l', \zeta_{(R,l')})) \mid (R, l') \in \mathsf{edges}(l, a')\} \tag{4}$$

and $(l', (v{+}t')[R]) \in \mathsf{tpre}(l', \zeta_{(R,l')})$ for all $(R, l') \in \mathsf{edges}(l, a)$. Now by the induction hypothesis for any $(R, l') \in \mathsf{edges}(l, a)$:

$$T^n_{\llbracket \mathcal{P} \rrbracket}(f)(l', (v{+}t')[R]) \leqslant T^n_{(\mathtt{Z},\mathtt{E})}(g)(l', \zeta_{(R,l')})(l', (v{+}t')[R])\,. \tag{5}$$

Furthermore, by Definition 5 we have $(t', a') \in A(s)$. Now by Definition 3:

$$
\begin{aligned}
T^{n+1}_{\llbracket\mathcal{P}\rrbracket}(f)(l,\nu) &= \inf_{(t,a)\in A(l,v)} \left\{ R_{\llbracket\mathcal{P}\rrbracket}(s,(t,a)) + \sum_{s'\in S} P_{\llbracket\mathcal{P}\rrbracket}(s,(t,a))(s')\cdot T^{n}_{\llbracket\mathcal{P}\rrbracket}(f)(s') \right\} \\
&\leqslant R_{\llbracket\mathcal{P}\rrbracket}(s,(t',a')) + \sum_{s'\in S} P_{\llbracket\mathcal{P}\rrbracket}(s,(t',a'))(s')\cdot T^{n}_{\llbracket\mathcal{P}\rrbracket}(f)(s') \quad \text{(since } (t',a')\in A(s)\text{)} \\
&= t' + \sum_{(R,l')\in\mathsf{edges}(l,a)} \mathsf{prob}(l,a')(R,l')\cdot T^{n}_{\llbracket\mathcal{P}\rrbracket}(f)(l',(v{+}t')[R]) \quad \text{(by Definition 5)} \\
&\leqslant t' + \sum_{(R,l')\in\mathsf{edges}(l,a)} \mathsf{prob}(l,a')(R,l')\cdot T^{n}_{(\mathtt{Z},\mathtt{E})}(g)(l',\zeta_{(R,l')})(l',(v{+}t')[R]) \quad \text{(by (5))} \\
&= t' + \sum_{((l,\zeta),a,(R,l'),(l',\zeta'))\in E'} \mathsf{prob}(l,a')(R,l')\cdot T^{n}_{(\mathtt{Z},\mathtt{E})}(g)(l',\zeta_{(R,l')})(l',(v{+}t')[R]) \\
&\hspace{30em} \text{(by (4))}
\end{aligned}
$$

Since $z{=}(l,\zeta) \in \mathtt{Z}$ such that $v{+}t \in \zeta$ for some $t \in \mathbb{R}$, $t' \in \mathbb{R}$ such that $v{+}t' \in \zeta$ and $E' \in \mathtt{E}(l,\zeta)$ were arbitrary, by Definition 6 it follows that:

$$
T^{n+1}_{\llbracket\mathcal{P}\rrbracket}(f)(s) \leqslant \min\left\{ T^{n+1}_{(\mathtt{Z},\mathtt{E})}(g)(\mathsf{z})(s) \mid \mathsf{z}\in\mathtt{Z} \wedge s\in\mathsf{tpre}(\mathsf{z}) \right\}. \tag{6}
$$

Combining (3) and (6) we have:

$$
T^{n+1}_{\llbracket\mathcal{P}\rrbracket}(f)(s) = \min\left\{ T^{n+1}_{(\mathtt{Z},\mathtt{E})}(g)(\mathsf{z})(s) \mid \mathsf{z}\in\mathtt{Z} \wedge s\in\mathsf{tpre}(\mathsf{z}) \right\}.
$$

and hence, since $s \in S$ was arbitrary, the proposition holds by induction. □

Rational simple functions and rational nice functions. In [2], the authors introduce simple functions and show that all value functions encountered during the iterative procedure for computing the minimum time reachability for TAs belong to this special class. For a zone ζ, a function $f : \zeta{\to}\mathbb{R}$ is *simple* if and only if it can be represented as:

$$
f(v) = \begin{cases} c_j & \text{if } v \in C_j \\ d_l{-}v(x_l) & \text{if } v \in D_l \end{cases}
$$

where $c_j, d_l \in \mathbb{N}$, $x_l \in \mathcal{X}$, C_j and D_l are zones for $1{\leqslant}j{\leqslant}M$ and $1{\leqslant}l{\leqslant}N$.

When it comes to PTAs, due to the presence of probabilistic branching, simple functions are not sufficient, as shown by the example below. Moreover, the domain of clocks cannot be represented by zones, as we now need to allow more general linear constraints on clocks with rational coefficients.

Example 2. We return to the PTA of Example 1 (see Figure 1). Expressing the minimum expected time in the initial location as a function $f : \mathbb{R}^{\mathcal{X}}{\to}\mathbb{R}$ we have:

$$
f(v) = \left\{ \begin{array}{cl} 2.9 & \text{if } x{\leqslant}2.1 \\ 5{-}v(x) & \text{if } 2.1{\leqslant}x{\leqslant}5 \\ 0 & \text{if } 5{\leqslant}x{\leqslant}10 \end{array} \right.
$$

and hence it cannot be represented using simple functions.

We introduce *rational simple functions* to represent the functions encountered during value iteration. Let $\mathcal{X} = \{x_1, \dots, x_n\}$ and k be the maximum constant appearing in $\mathcal{P}$. By Assumption 2(a) $\mathcal{P}$ is bounded, and hence all clock values in $\mathcal{P}$ are bounded by k.

Definition 7. *A (convex)* k-polyhedron $C \subseteq \{v \in \mathbb{R}^{\mathcal{X}} \mid v(x) \leqslant k \text{ for } x \in \mathcal{X}\}$ *is defined by finitely many linear inequalities; formally, it is of the form:*

$$C = \left\{v \in \mathbb{R}^{\mathcal{X}} \mid \textstyle\sum_{i=1}^{n} q_{ij} \cdot v(x_i) \leqslant f_j \text{ for } 1 \leqslant j \leqslant M\right\}$$

where $q_{ij}, f_j \in \mathbb{Q}$ *and* $f_j \leqslant k$ *for all* $1 \leqslant i \leqslant n$ *and* $1 \leqslant j \leqslant M$ *for some* $M \in \mathbb{N}$.

Definition 8. *For zone* ζ*, a function* $f : \zeta \to \mathbb{R}$ *is* rational k-simple *if and only if it can be represented as:*

$$f(v) = \begin{cases} c_j & \text{if } v \in C_j \\ d_l - \sum_{i=1}^{n} p_{il} \cdot v(x_i) & \text{if } v \in D_l \end{cases}$$

where $c_j, d_l \in \mathbb{Q}_+$, $p_{il} \in \mathbb{Q}_+ \cap [0,1]$ *such that* $\sum_{i=1}^{n} p_{il} \leqslant 1$ *and* C_j, D_l *are* k*-polyhedra for all* $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant M$ *and* $1 \leqslant l \leqslant N$.

Furthermore, a function $f : \mathsf{Z} \to (S \to \mathbb{R})$ *is rational* k*-simple if* $f(l,\zeta)(l,\cdot) : \swarrow \zeta \to \mathbb{R}$ *is rational* k*-simple for all* $(l,\zeta) \in Z$.

We require the following definition and lemma for rational k-simple functions.

Definition 9. *If* $f : \zeta \to \mathbb{R}$ *is a rational* k*-simple function and* $R \subseteq \mathcal{X}$*, let* $f[R] : [R]\zeta \to \mathbb{R}$ *be the function where* $f[R](v) = f(v[R])$ *for all* $v \in \zeta$.

Lemma 1. *If* $f : \zeta \to \mathbb{R}$ *is rational* k*-simple and* $R \subseteq \mathcal{X}$*, then* $f[R] : [R]\zeta \to \mathbb{R}$ *is rational* k*-simple.* (The proof can be found in [15].)

During value iteration we obtain functions of the form $v \mapsto t + f(l,\zeta)(l, v+t)$ where f is rational k-simple. This motivates the introduction of rational k-nice functions, based on Asarin and Maler's k-nice functions [2].

Definition 10. *A* k-bipolyhedron *is a set of the form* $\{(v,t) \mid v \in C \wedge v+t \in D\}$ *where* C *and* D *are* k*-polyhedra. For a zone* ζ*, a function* $g : (\zeta \times \mathbb{R}) \to \mathbb{R}$ *is* rational k-nice *if and only if it can be represented as:*

$$g(v,t) = \begin{cases} c_j + t & \text{if } (v,t) \in F_j \\ d_l - \sum_{i=1}^{n} p_{il} \cdot v(x_i) + \left(1 - \sum_{i=1}^{n} p_{il}\right) \cdot t & \text{if } (v,t) \in G_l \end{cases}$$

where $c_j, d_l \in \mathbb{Q}_+$, $p_{il} \in \mathbb{Q}_+ \cap [0,1]$ *such that* $\sum_{i=1}^{n} p_{il} \leqslant 1$ *and* F_j, G_l *are rational* k*-bipolyhedra for all* $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant M$ *and* $1 \leqslant l \leqslant N$.

We require the following properties of k-nice functions (proofs are available in [15]).

Lemma 2. *A convex combination of rational* k*-nice functions is rational* k*-nice.*

Lemma 3. *The minimum of rational k-nice functions is rational k-nice.*

Lemma 4. *For any zone ζ, if $g : (\zeta{\times}\mathbb{R}){\to}\mathbb{R}$ is rational k-nice, then the function $f : \zeta{\to}\mathbb{R}$ where $f(v) = \inf_{t\in\mathbb{R}} g(v,t)$ for $v \in \zeta$ is rational k-simple.*

We are now in a position to show that that rational k-simple functions are a suitable representation for value functions.

Proposition 2. *If $f : \mathtt{Z}{\to}(S{\to}\mathbb{R})$ is a rational k-simple function, then $T_{(\mathtt{Z},\mathtt{E})}(f)$ is rational k-simple.*

Proof. Consider any rational k-simple function, $\mathtt{z} \in \mathtt{Z}$ and $E \in \mathtt{E}(\mathtt{z})$. For any $v \in \mathbb{R}^{\mathcal{X}}$ and $t \in \mathbb{R}$ we have:

$$\begin{aligned}
&t + \textstyle\sum_{((l,\zeta),a,(R,l'),(l',\zeta'))\in E} \mathsf{prob}(l,a)(R,l'){\cdot} f(l',\zeta')(l',(v{+}t)[R]) \\
&\quad = t + \textstyle\sum_{((l,\zeta),a,(R,l'),(l',\zeta'))\in E} \mathsf{prob}(l,a)(R,l'){\cdot} f[R](l',\zeta')(l',v{+}t) \\
&\hfill \text{(by Definition 9)} \\
&\quad = \textstyle\sum_{((l,\zeta),a,(R,l'),(l',\zeta'))\in E} \mathsf{prob}(l,a)(R,l'){\cdot} \big(t + f[R](l',\zeta')(l',v{+}t)\big) \qquad (7)
\end{aligned}$$

since $\mathsf{prob}(l,a)$ is a distribution. By construction f is rational k-simple, and hence for any $(\mathtt{z},a,(R,l'),\mathtt{z}) \in E$ using Lemma 1 we have $f[R]$ is also rational k-simple. Therefore, it follows from Definition 10 that:

$$(v,t) \mapsto t + f[R](l',\zeta')(l',v{+}t)$$

is rational k-nice. Thus, since $(\mathtt{z},a,(R,l'),\mathtt{z}) \in E$ was arbitrary, using Lemma 2 and (7) we have that:

$$(v,t) \mapsto t + \textstyle\sum_{((l,\zeta),a,(R,l'),(l',\zeta'))\in E} \mathsf{prob}(l,a)(R,l'){\cdot} f(l',\zeta')(l',(v{+}t)[R])$$

is also rational k-nice. Since $E \in \mathtt{E}(\mathtt{z})$ was arbitrary and $\mathtt{E}(\mathtt{z})$ is finite, Lemma 3 tells us:

$$(v,t) \mapsto \min_{E\in\mathtt{E}(\mathtt{z})} \Big\{ t + \textstyle\sum_{((l,\zeta),a,(R,l'),(l',\zeta'))\in E} \mathsf{prob}(l,a)(R,l'){\cdot} f(l',\zeta')(l',(v{+}t)[R]) \Big\}$$

is again rational k-nice. Finally, using Definition 6 and Lemma 4, it follows that $T_{(\mathtt{Z},\mathtt{E})}(f)(\mathtt{z})$ is rational k-simple as required. □

Controller Synthesis. We now give an approach for computing the minimum expected time of reaching a target in a PTA and synthesising ε-optimal strategy when starting from the initial state. We first build the backwards zone graph $(\mathtt{Z},\mathtt{E})$ (see Figure 2), then, using Theorem 2 and graph-based algorithms [13], we can find the states of $[\![\mathcal{P}]\!]$ for which the maximum probability of reaching the target is less than 1 and remove these from the zone graph. Next, using Definition 6, we apply value iteration to the zone graph which, by Proposition 2, can be performed using rational k-simple functions (and rational k-nice functions). Convergence to the minimum expected reachability values of $\mathcal{P}$ is guaranteed by

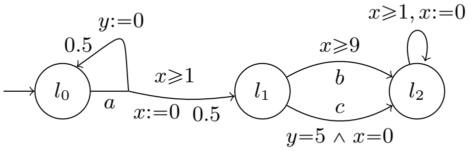

Fig. 3. PTA

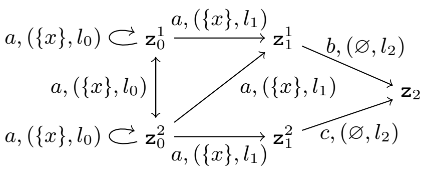

Fig. 4. Backwards Zone graph

Proposition 1 and Theorem 1. An ε-optimal deterministic, memoryless strategy can be synthesised once value iteration has converged by starting from the initial state and stepping through the backwards graph, in each state choosing the time and action that achieve the values returned by value iteration.

Example 3. The PTA in Figure 3 presents an example where waiting longer than necessary in a location can reduce the time to reach the target. Again we suppose the invariant in all locations is $x{\leqslant}10$. The target is location l_2 and the zone graph is given in Figure 4, where $\mathtt{z}_0^1{=}(l_0, x{\geqslant}1)$, $\mathtt{z}_0^2{=}(l_0, y{=}5{\wedge}x{\geqslant}1)$, $\mathtt{z}_1^1{=}(l_1, x{\geqslant}9)$, $\mathtt{z}_1^2{=}(l_1, y{=}5{\wedge}x{=}0)$ and $\mathtt{z}_2{=}(l_2, x{\geqslant}1)$. Starting from the constant 0 function f_0 and performing value iteration gives for $n{\geqslant}\ 2$:

$$T^n_{\mathtt{(Z,E)}}(f_0)(\mathtt{z}_0^1) = \begin{cases} (1{-}v(x)){+}\sum_{i=1}^{n-1} 0.5^n{\cdot}9 & \text{if } x{\leqslant}1 \\ \quad\sum_{i=1}^{n} 0.5^{n-1}{\cdot}9 & \text{if } 1{\leqslant}x{\leqslant}10 \end{cases}$$

$$T^n_{\mathtt{(Z,E)}}(f_0)(\mathtt{z}_0^2) = \begin{cases} (5{-}v(y)){+}0.5{\cdot}(\sum_{i=1}^{n} 0.5^{n-1}{\cdot}9) & \text{if } y{\leqslant}5 \\ \quad 0.5{\cdot}(\sum_{i=1}^{n-1} 0.5^{n-1}{\cdot}9) & \text{if } 5{\leqslant}y{\leqslant}10 \end{cases}$$

$$T^n_{\mathtt{(Z,E)}}(f_0)(\mathtt{z}_1^1) = \begin{cases} 9{-}v(x) & \text{if } x{\leqslant}9 \\ \quad 0 & \text{if } 9{\leqslant}y{\leqslant}10 \end{cases}$$

and $T^n_{\mathtt{(Z,E)}}(f_0)(\mathtt{z}_1^2) = T^n_{\mathtt{(Z,E)}}(f_0)(\mathtt{z}_2) = 0$. Therefore, value iteration converges to:

$$f(\mathtt{z}_0^1) = \begin{cases} (1{-}v(x)){+}9 & \text{if } x{\leqslant}1 \\ \quad 9 & \text{if } 1{\leqslant}x{\leqslant}10 \end{cases} \text{ and } f(\mathtt{z}_0^2) = \begin{cases} (5{-}v(y)){+}0.5{\cdot}9 & \text{if } y{\leqslant}5 \\ \quad 0.5{\cdot}9 & \text{if } 5{\leqslant}y{\leqslant}10 \end{cases}$$

and hence the minimum expected time for the initial state equals the minimum of $(1{-}0){+}9$ and $(5{-}0){+}0.5{\cdot}9$, yielding 9.5. Performing controller synthesis we find this corresponds to waiting until $y{=}5$, then performing the action a. If l_1 is reached, we immediately perform the action c and reach the target. On the other hand, if l_0 is reached, we repeatedly immediately perform a and, if l_1 is reached, wait until $x{=}9$ and then perform the action b reaching the target.

5 Conclusion

We have proposed an algorithm to compute the minimum expected time to reach a target set in a PTA. The algorithm is formulated as value iteration over the backwards zone graph of the PTA. We also demonstrate that there is an effective representation of the value functions in terms of rational simple and rational nice functions. However, zones are not sufficient and convex polyhedra are required. Nevertheless, the Parma Polyhedra Library [3] offers efficient ways to manipulate convex polyhedra and is commonly used in a variety of real-time verification problems. For example, methods based on priced zones for TAs and PTAs, such as [7] and [22], also use convex polyhedra, where similarly zones do not suffice.

Regarding future work, as well as working on an implementation, we note that optimisations to the backwards algorithm presented in [8], including first performing forwards reachability to restrict analysis to the reachable state space, could be considered here as well. Since policy iteration also converges (see Theorem 1), we plan to investigate this approach and compare with value iteration. Extending to linearly-priced PTAs does not appear straightforward, as rational simple functions are not sufficient. Likewise, the case of maximum expected values raises additional issues, since here one relies on minimum probabilistic reachability, which is more complex to compute using zones and convexity is lost [21].

References

1. R. Alur and D. L. Dill. A theory of timed automata. *TCS*, 126:183–235, 1994.
2. E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *Proc. HSCC'99*, volume 1569 of *LNCS*, 1999.
3. R. Bagnara, P. M. Hill, and E. Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 72(1–2):3–21, 2008.
4. G. Behrmann, A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. In *Proc. HSCC'01*, volume 2034 of *LNCS*. Springer, 2001.
5. R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.
6. J. Berendsen, T. Chen, and D. Jansen. Undecidability of cost-bounded reachability in priced probabilistic timed automata. In *Proc. TAMC'09*, volume 5532 of *LNCS*. Springer, 2009.
7. J. Berendsen, D. Jansen, and J.-P. Katoen. Probably on time and within budget – On reachability in priced probabilistic timed automata. In *Proc. QEST'06*. IEEE Press, 2006.
8. J. Berendsen, D. Jansen, and F. Vaandrager. Fortuna: Model checking priced probabilistic timed automata. In *Proc. QEST'10*. IEEE Press, 2010.
9. D. Bertsekas. *Dynamic Programming and Optimal Control*, Volumes 1 and 2. Athena Scientific, 1995.
10. D. Bertsekas and J. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.
11. P. Bulychev, A. David, K. Larsen, M. Mikučionis, D. Bøgsted P., A. Legay, and Z. Wang. UPPAAL-SMC: Statistical model checking for priced timed automata. In *Proc. QAPL'12*, volume 85 of *EPTCS*. Open Publishing Association, 2012.
12. A. David, P. Jensen, K. Larsen, A. Legay, D. Lime, M. Sørensen, and J. Taankvist. On time with minimal expected cost! In *Proc. ATVA*, volume 8837 of *LNCS*. Springer, 2014.

13. L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In *Proc. CONCUR'99*, volume 1664 of *LNCS*. Springer, 1999.
14. H. James and E. Collins. An analysis of transient Markov decision processes. *Journal of Applied Probability*, 43 (3):603–621, 2006.
15. A. Jovanović, M. Kwiatkowska, and G. Norman. Symbolic minimum expected time controller synthesis for probabilistic timed automata. Technical Report CS-RR-15-04, Oxford University, 2015.
16. J. Kemeny, J. Snell, and A. Knapp. *Denumerable Markov Chains.* Springer, 1976.
17. M. Kwiatkowska, G. Norman, and D. Parker. Stochastic games for verification of probabilistic timed automata. In *Proc. FORMATS'09*, volume 5813 of *LNCS*. Springer, 2009.
18. M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV'11*, volume 6806 of *LNCS*. Springer, 2011.
19. M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston. Performance analysis of probabilistic timed automata using digital clocks. *FMSD*, 29:33–78, 2006.
20. M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *TCS*, 282:101–150, 2002.
21. M. Kwiatkowska, G. Norman, J. Sproston, and F. Wang. Symbolic model checking for probabilistic timed automata. *Information and Computation*, 205(7):1027–1077, 2007.
22. K. Larsen, G. Berhmann, E. Brinksma, A. Fehnker, T. Hune, P. Pettersson, and J. Romijn. As cheap as possible: Efficient cost-optimal reachability for priced timed automata. In *Proc. CAV'02*, volume 2102 of *LNCS*. Springer, 2001.
23. K. G. Larsen, P. Pettersson, and W. Yi. Uppaal in a Nutshell. *Int. Journal on Software Tools for Technology Transfer*, 1:134–152, 1997.
24. S. Tripakis. *The analysis of timed systems in practice.* PhD thesis, Université Joseph Fourier, Grenoble, 1998.
25. S. Tripakis. Verifying progress in timed systems. In *Proc. ARTS'99*, volume 1601 of *LNCS*. Springer, 1999.
26. S. Tripakis, S. Yovine, and A. Bouajjan. Checking timed Büchi automata emptiness efficiently. *Formal Methods in System Design*, 26(3):267–292, 2005.