Symbolic Verification and Strategy Synthesis for Linearly-Priced Probabilistic Timed Automata

Kwiatkowska, Marta; Norman, Gethin; Parker, David

DOI:
10.1007/978-3-319-63121-9_15

Symbolic Verification and Strategy Synthesis for Linearly-Priced Probabilistic Timed Automata

Marta Kwiatkowska[1], Gethin Norman[2], and David Parker[3]

[1] Department of Computer Science, University of Oxford, Oxford, UK
[2] School of Computing Science, University of Glasgow, Glasgow, UK
[3] School of Computer Science, University of Birmingham, Birmingham, UK

Abstract. Probabilistic timed automata are a formalism for modelling systems whose dynamics includes probabilistic, nondeterministic and timed aspects including real-time systems. A variety of techniques have been proposed for the analysis of this formalism and successfully employed to analyse, for example, wireless communication protocols and computer security systems. Augmenting the model with prices (or, equivalently, costs or rewards) provides a means to verify more complex quantitative properties, such as the expected energy usage of a device or the expected number of messages sent during a protocol's execution. However, the analysis of these properties on probabilistic timed automata currently relies on a technique based on integer discretisation of real-valued clocks, which can be expensive in some cases. In this paper, we propose symbolic techniques for verification and optimal strategy synthesis for priced probabilistic timed automata which avoid this discretisation. We build upon recent work for the special case of expected time properties, using value iteration over a zone-based abstraction of the model.

1 Introduction

Real-time systems are at the heart of application domains such as communication protocols, embedded systems, hardware circuits, autonomous transport, robotics and manufacturing. The presence of hard real-time constraints within a distributed, reactive environment means that their correct functioning depends on the timing pattern of the interaction of the system with its environment, making correctness guarantees difficult.

Timed automata [2] are a powerful formalism for modelling and verification of real-time systems. They are finite-state automata equipped with real-valued clocks which measure the passage of time, and whose transitions are annotated with guards that specify the time constraints that have to be satisfied for the transition to be taken. Since timed automata allow the modelling of dense real-time, the decidability of model checking depends on a number of assumptions.

Several verification approaches have been introduced, see e.g. [1,21,22,32], of which the symbolic *zone-based* approach enables greater scalability compared to the digital clocks method, which assumes an integral model of time as opposed to a dense model of time. Timed automata have been widely used for modelling and analysis of real-world systems; in particular, they are supported by the UPPAAL [31] model checker, the gold standard in computer-aided verification for real-time systems.

When modelling and analysing real-time systems, it is often necessary to consider quantities other than time, for example energy consumption, network bandwidth or number of packets lost. The model of *(linearly) priced timed automata* [3,7] extends timed automata with *prices* (weights) annotating the locations and transitions, thus enabling reasoning about costs or rewards accumulated over time as the execution progresses. This model has good decidability properties and several algorithms have been proposed for its analysis, based on an extension of regions or zones with prices. Priced timed automata are also supported by UPPAAL, and have been used for timing analysis of a range of embedded real-time systems, with several flaws discovered and corrected.

However, many distributed real-time systems also employ *randomisation*, for example random back-off in wireless network protocols. A natural model for such systems is a probabilistic extension of (priced) timed automata called *probabilistic timed automata* (PTAs) [19,29,6]. They can be viewed as timed automata whose transitions are probability distributions over the set of edges, where each such edge specifies a successor location and a set of clocks to reset.

A key property studied here is *expected reachability*, namely the expected time/price until some event occurs. This problem has been found unsuitable for symbolic zone-based methods, including priced zones, since accumulated prices are unbounded. Recently, [25,24] introduced a zone-based symbolic method to compute *minimum and maximum expected time* for PTAs and to synthesise a corresponding strategy. Prior to this, expected reachability properties of PTAs could only be verified using the digital clocks method [28] that can suffer from state-space explosion.

Probabilistic timed automata are supported by the PRISM [27] model checker via the zone-based and digital clocks abstractions (though not yet the method of [25]) and have used been for the analysis of a broad range of real-world protocols, see for example [28,18]. A second tool supporting PTAs is `mcpta` [20], which applies the digital clocks abstraction to translate a subset of the modelling language Modest [15] directly into the PRISM modelling language. The related problem of price-bounded probabilistic reachability [10] (known to be undecidable [9]) can be analysed via a semi-decision procedure using priced zones, implemented in Fortuna [11].

In this paper we study the computation of the *minimum/maximum expected price* for linearly-priced probabilistic timed automata, for which, to the best of our knowledge, no zone-based method exists at present. More specifically, we extend [25], where only the restricted case of expected time is considered. The minimum expected price problem for a related model of priced timed games in stochastic environments was tackled in [16] using *statistical model checking* with Uppaal-SMC. Since this approach is based on simulation, rather than numerical model checking, it gives approximate results with probabilistic guarantees.

As in [25,24], our method relies on an interpretation of the PTA as an uncountable-state Markov decision process (MDP) and employs a representation in terms of an extension of the 'simple' and 'nice' functions of [4]. The optimal prices are computed via a Bellman equation using value iteration, which gives guaranteed eventual convergence to the correct values. Moreover, an ε-optimal strategy can be extracted by stepping backwards and retrieving the locally optimal choices once some convergence criterion has been satisfied. For minimum expected time, it is always optimal to let as little time pass as possible. However, for minimum price, it turns out that this is not always the case, and it can be optimal to let time pass now and accumulate a lower price, as opposed to waiting and accumulating a higher price later. The case of maximum time/price is dual.

Paper structure. In Section 2 we summarise the relevant background, mainly concerning uncountable MDPs and the computation of optimal reward. Section 3 defines the priced extension of probabilistic timed automata (PTAs) and their interpretation as an uncountable MDP under appropriate assumptions. In Section 4, we introduce a representation of the value functions that generalise the simple and nice functions of [4], and present our algorithms for computing optimal expected price and synthesis of an ε-optimal strategy using the backwards zone graph of a PTA.

2 Background

Let $\mathbb{R}$ denote the non-negative reals, $\mathbb{N}$ natural numbers, $\mathbb{Q}$ rationals and $\mathbb{Q}_+$ non-negative rationals. A *discrete probability distribution* over a (possibly uncountable) set S is a function $\mu : S{\rightarrow}[0,1]$ such that $\sum_{s \in S} \mu(s) = 1$ and the set $\{s \in S \mid \mu(s){>}0\}$ is finite. Let $\mathsf{dist}(S)$ denote the set of distributions over S. A distribution $\mu \in \mathsf{dist}(S)$ is a *point distribution* if $\mu(s){=}1$ for some $s \in S$.

In preparation for the sections that follow, we present some background material and known results for the model of Markov decision processes (MDPs).

Definition 1. *An* MDP *is a tuple* $\mathsf{M}{=}(S, s_0, A, \mathit{Prob}_\mathsf{M}, \mathit{Price}_\mathsf{M})$*, where:*

- S *is a (possibly uncountable) set of states and* $s_0 \in S$ *is an initial state;*
- A *is a (possibly uncountable) set of actions;*
- $\mathit{Prob}_\mathsf{M} : S{\times}A \rightarrow \mathsf{dist}(S)$ *is a (partial) probabilistic transition function;*
- $\mathit{Price}_\mathsf{M} : S{\times}A \rightarrow \mathbb{R}$ *is a price function.*

In each state s of an MDP M, there is a set of enabled actions, denoted by $A(s)$, containing those actions a for which $\mathit{Prob}_\mathsf{M}(s,a)$ is defined. In state s, a transition corresponds to first nondeterministically choosing an available action and, assuming action $a \in A(s)$ is chosen, then selecting a successor state randomly according to the distribution $\mathit{Prob}_\mathsf{M}(s,a)$. Taking an a-labelled transition from state s incurs a *price* of $\mathit{Price}_\mathsf{M}(s,a)$. We use the terminology "price" for consistency with the model of priced probabilistic timed automata used later, but these are commonly also referred to as *costs* or, dually, *rewards* for MDPs.

A *path* of an MDP M is given by a finite or infinite sequence of transitions $\omega = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots$ with $\mathit{Prob}_{\mathsf{M}}(s_i, a_i)(s_{i+1}){>}0$ for all $i{\geq}0$. The $(i{+}1)$th state of a path ω and action associated with the $(i{+}1)$th transition are denoted by $\omega(i)$ and $\omega[i]$ respectively. The set of infinite (finite) paths is denoted by $\mathit{IPaths}_{\mathsf{M}}$ ($\mathit{FPaths}_{\mathsf{M}}$) and the last state of a finite path ω by $\mathit{last}(\omega)$.

A *strategy* (also called an adversary, scheduler or policy) of an MDP M represents one resolution of the nondeterminism in M.

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

For a given strategy σ and state s of an MDP M, we can construct a probability measure $\mathcal{P}_s^\sigma$ over the set of infinite paths starting in s [26]. 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 $\omega \in \mathit{FPaths}_{\mathsf{M}}$. The set of all strategies of MDP M is denoted Σ_{M}.

Key quantitative properties for MDPs are the probability of reaching a target and the expected price incurred before doing so. We will refer to these as *probabilistic reachability* and *expected reachability*, respectively. For a strategy σ, state s and set of target states $F \subseteq S$ of an MDP M, these values are given by:

$$\mathbb{P}_{\mathsf{M}}^{\sigma}(s,F) \overset{\text{def}}{=} \mathcal{P}_s^\sigma\{\omega \in \mathit{IPaths}_{\mathsf{M}} \mid \exists k \in \mathbb{N}.\, \omega(k) \in F\}$$
$$\mathbb{E}_{\mathsf{M}}^{\sigma}(s,F) \overset{\text{def}}{=} \textstyle\int_{\omega \in \mathit{IPaths}_{\mathsf{M}}} \mathit{price}(\omega, F)\, \mathrm{d}\mathcal{P}_s^\sigma$$

where for any infinite path ω:

$$\mathit{price}(\omega, F) \overset{\text{def}}{=} \textstyle\sum_{i=0}^{k_F} \mathit{Price}_{\mathsf{M}}(\omega(i), \omega[i])$$

and $k_F = \min\{k{-}1 \mid \omega(k) \in F\}$ if there exists $k \in \mathbb{N}$ such that $\omega(k) \in F$ and $k_F = \infty$ otherwise. As usual we consider the optimal values of these properties, i.e. the minimum and maximum values over all strategies:

$$\mathbb{P}_{\mathsf{M}}^{\min}(s,F) \overset{\text{def}}{=} \inf\nolimits_{\sigma \in \Sigma_{\mathsf{M}}} \mathbb{P}_{\mathsf{M}}^{\sigma}(s,F) \qquad \mathbb{P}_{\mathsf{M}}^{\max}(s,F) \overset{\text{def}}{=} \sup\nolimits_{\sigma \in \Sigma_{\mathsf{M}}} \mathbb{P}_{\mathsf{M}}^{\sigma}(s,F)$$
$$\mathbb{E}_{\mathsf{M}}^{\min}(s,F) \overset{\text{def}}{=} \inf\nolimits_{\sigma \in \Sigma_{\mathsf{M}}} \mathbb{E}_{\mathsf{M}}^{\sigma}(s,F) \qquad \mathbb{E}_{\mathsf{M}}^{\max}(s,F) \overset{\text{def}}{=} \sup\nolimits_{\sigma \in \Sigma_{\mathsf{M}}} \mathbb{E}_{\mathsf{M}}^{\sigma}(s,F)$$

One approach to computing these optimal values is through *Bellman operators* [8] using either *value iteration* or *policy iteration* [12,13]. In the case of expected reachability, the Bellman operators have the following form.

Definition 3. *Let* M *be an MDP with state space* S*,* $F \subseteq S$ *be a target set, and let* $\mathrm{opt} \in \{\min, \max\}$*. The Bellman operator* $T_{\mathsf{M}}^{\mathrm{opt}} : (S{\rightarrow}\mathbb{R}) \rightarrow (S{\rightarrow}\mathbb{R})$ *for optimal expected reachability is defined as follows. For any function* $f : S \rightarrow \mathbb{R}$ *and state* $s \in S$*:*

$$T_{\mathsf{M}}^{\mathrm{opt}}(f)(s) = \begin{cases} 0 & \text{if } s \in F \\ \mathrm{opt}^{\star}_{a \in A(s)} \left\{ \mathit{Price}_{\mathsf{M}}(s,a) + \sum_{s' \in S} \mathit{Prob}_{\mathsf{M}}(s,a)(s') {\cdot} f(s') \right\} & \text{if } s \notin F \end{cases}$$

where $\min^\star = \inf$ *and* $\max^\star = \sup$*.*

Value iteration works by starting with an initial approximation $f_0 : S \to \mathbb{R}$ and repeatedly applying $T^{\mathrm{opt}}_{\mathsf{M}}$ until it converges to the optimal expected reachability value. In practice, an approximate result is obtained by terminating the computation once some convergence criterion is satisfied, for example, by checking that the maximum pointwise difference between $(T^{\mathrm{opt}}_{\mathsf{M}})^n(f_0)$ and $(T^{\mathrm{opt}}_{\mathsf{M}})^{n+1}(f_0)$ is below some threshold $\varepsilon \in \mathbb{R}$. The process also yields an (approximately) optimal strategy for either minimising or maximising expected reachability. Policy iteration starts from a (deterministic and memoryless) strategy, and repeatedly attempts to find an improved (deterministic and memoryless) strategy by computing the expected reachability values for the current strategy and trying to update action choices to optimise expected reachability values.

Below, we state some known results from [23] regarding MDPs and value iteration, which are needed later in the paper (and which were adapted for the case of PTAs in [25]). This requires us to make the following assumptions.

Assumption 1. *For any MDP* $\mathsf{M}{=}(S, s_0, A, \mathit{Prob}_{\mathsf{M}}, \mathit{Price}_{\mathsf{M}})$ *and target set* F*:*

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

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

Theorem 1 ([23]). *If* M *and* F *are an MDP and target set for which Assumption 1 holds, and the minimum expected price values are bounded below, then:*

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

Corollary 1. *If* M *and* F *are an MDP and target set for which Assumption 1 holds and the maximum expected price values are bounded above, then:*

- *there exists a memoryless, deterministic strategy that achieves the maximum expected price of reaching* F*;*
- *the maximum expected price values are the unique solutions to* $T^{\max}_{\mathsf{M}}$*;*
- *value iteration over* $T^{\max}_{\mathsf{M}}$ *converges to the maximum expected price values when starting from any bounded function;*
- *policy iteration converges to the maximum expected price values when starting from any proper, memoryless, deterministic strategy.*

3 Priced Probabilistic Timed Automata

In this section we introduce *probabilistic timed automata* (PTAs) [19,29,6], a formalism for modelling systems whose dynamics includes probabilistic, nondeterministic and timed aspects, and the extended model of *linearly-priced PTAs* [28], which augment PTAs with prices. We will commonly refer to the latter simply as PTAs.

Clocks, clock valuations and zones. We assume we have a finite set $\mathcal{X}$ of real-valued variables called *clocks* which increase at the same, constant rate. A clock valuation is a function $v : \mathcal{X}{\to}\mathbb{R}$ and let $\mathbb{R}^{\mathcal{X}}$ be the set of all clock valuations. We denote by $\mathbf{0}$ the clock valuation that assigns 0 to all clocks. For any subset of clocks R, non-negative real value t and clock valuation v, $v[R]$ is the clock valuation where $v[R](x){=}0$ if $x \in R$ and $v[R](x){=}v(x)$ if $x \in \mathcal{X}\backslash R$, and $v{+}t$ is the clock valuation where $(v{+}t)(x){=}v(x){+}t$ for all $x \in \mathcal{X}$. The set of *zones* over $\mathcal{X}$, written $\mathit{Zones}(\mathcal{X})$, is defined by the syntax:

$$\zeta ::= \mathtt{true} \mid x \leq d \mid c \leq x \mid x{+}c \leq y{+}d \mid \neg\zeta \mid \zeta \wedge \zeta$$

where $x, y \in \mathcal{X}$ and $c, d \in \mathbb{N}$. We can restrict the syntax to *convex* zones by removing negation. For a clock valuation v and zone ζ, we say v satisfies ζ, denoted $v{\models}\zeta$, if ζ is true after substituting each occurrence of each clock x with $v(x)$. The semantics of a zone ζ is the set of clock valuations satisfying it. We require the following zone operations [33], for zone ζ and subset of clocks R:

- $\swarrow\zeta = \{v \in \mathbb{R}^{\mathcal{X}} \mid \exists t \in \mathbb{R}.\, v{+}t \models \zeta\}$;
- $\zeta[R] = \{v[R] \mid v \models \zeta\}$;
- $[R]\zeta = \{v \in \mathbb{R}^{\mathcal{X}} \mid v[R] \models \zeta\}$.

Syntax and semantics of PTAs. We now present the formal syntax and semantics of linearly-priced PTAs.

Definition 4. *A linearly-priced probabilistic timed automaton (PTA)* P *is a tuple* $(L, l_0, \mathcal{X}, \mathit{Act}, \mathsf{enab}, \mathsf{prob}, \mathsf{inv}, \mathsf{price})$ *where:*

- L *is a finite set of locations and* $l_0 \in L$ *is an initial location;*
- $\mathcal{X}$ *is a finite set of clocks;*
- Act *is a finite set of actions;*
- $\mathsf{enab} : L{\times}\mathit{Act} \to \mathit{Zones}(\mathcal{X})$ *is an enabling condition;*
- $\mathsf{prob} : L{\times}\mathit{Act} \to \mathsf{dist}(2^{\mathcal{X}}{\times}L)$ *is a probabilistic transition function;*
- $\mathsf{inv} : L \to \mathit{Zones}(\mathcal{X})$ *is an invariant condition;*
- $\mathsf{price} = (\mathsf{price}_L, \mathsf{price}_{\mathit{Act}})$ *is a price structure where* $\mathsf{price}_L : L \to \mathbb{Q}_+$ *is a location price function and* $\mathsf{price}_{\mathit{Act}} : L{\times}\mathit{Act} \to \mathbb{Q}_+$ *an action price function.*

The underlying semantics of PTA P is an MDP with an infinite set of both states and actions. The states are location-valuation pairs (l, v) such that v satisfies the invariant $\mathsf{inv}(l)$ and the initial state is the initial location with all clocks set to 0. The available actions in state (l, v) are the time-action pairs (t, a) such the invariant $\mathsf{inv}(l)$ remains true while letting t time units pass, after this time the enabling condition $\mathsf{enab}(l,a)$ is satisfied, and the successor location and the clocks that are reset are then chosen according to the distribution $\mathsf{prob}(l,v)$. Furthermore, a price is incurred at rate $\mathsf{price}_L(l)$ while letting the t time units pass and a price $\mathsf{price}_{Act}(l,a)$ is incurred when performing the action a.

Definition 5. *For a PTA* $\mathsf{P}{=}(L,l_0,\mathcal{X},Act,\mathsf{enab},\mathsf{prob},\mathsf{inv},\mathsf{price})$ *the semantics of* P *is given by the MDP* $[\![\mathsf{P}]\!]{=}(S,s_0,\mathbb{R}{\times}Act,Prob_{[\![\mathsf{P}]\!]},Price_{[\![\mathsf{P}]\!]})$ *where:*

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

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

- $Price_{[\![\mathsf{P}]\!]}((l,v),(t,a)) = \mathsf{price}_L(l){\cdot}t{+}\mathsf{price}_{Act}(l,a)$ *for all* $(l,v)\in S$ *and* $(t,a)\in\mathbb{R}{\times}Act$*.*

Expected prices. The property of PTAs on which we focus in this paper is the optimal (minimum or maximum) expected price incurred before reaching a target, which is defined along the same lines as the equivalent property for MDPs defined in Section 2. The differences are that, firstly, the target is now defined as a set $F\subseteq L$ of locations and, secondly, prices are incurred both when time elapses in a location, and when an action is performed. Since the semantics of a PTA is an (infinite-state) MDP, the expected price for a PTA is defined in straightforward fashion in terms of the MDP. For PTA P, target locations F, state (l,v) and opt $\in\{\min,\max\}$, we have:

$$\mathbb{E}^{\mathrm{opt}}_{\mathsf{P}}((l,v),F) \overset{\mathrm{def}}{=} \mathbb{E}^{\mathrm{opt}}_{[\![\mathsf{P}]\!]}((l,v),S_F) \quad\text{where}\quad S_F \overset{\mathrm{def}}{=} \{(l,v)\mid l\in F\wedge v\models\mathsf{inv}(l)\}\,.$$

When computing these values, we make several assumptions about PTAs, similar to those imposed in [25]. Firstly, this will ensure that Assumption 1 holds for the underlying MDP, which allows us to apply Theorem 1 and Corollary 1. Secondly, it makes sure that unrealistic behaviours are discarded.

Assumption 2. *For any PTA* P*, we have:*

(a) *all invariants of* P *are bounded;*
(b) *only non-strict inequalities are allowed in clock constraints, i.e.,* P *is closed;*
(c) *all invariant and enabling conditions of* P *are convex;*
(d) *all location prices of* P *are positive;*
(e) P *is structurally non-zeno [34] (this can be identified syntactically and in a compositional fashion [35] and guarantees time-divergent behaviour).*

The reasons for these assumptions are similar to those given in [25]. The main difference is that, in order to ensure that Assumption 1(c) holds, we require that all location prices are positive (Assumption 2(d)), in addition to the structural non-zeno assumption. More precisely, for any PTA satisfying Assumption 2(d) and (e), if, from some state and under some strategy a target is not reached with probability 1, then from this state and under this strategy the expected price of reaching the target is infinite. Expected time (as in [25]) is a special case of expected price where all action prices are 0 and all location prices are 1, and therefore Assumption 1(d) will always hold in this case.

4 Optimal Expected Price Algorithms for PTAs

In this section, we present our symbolic approach for computing optimal expected reachability prices and for synthesising a corresponding optimal strategy. We first extend the approach of [25] for computing optimal expected times, a key building block of which is an initial backwards exploration of the state space, using the techniques from [30]. Computing expected rewards can then be performed using value iteration over the zone graph constructed during backwards exploration. This process is described in Section 4.1. Next, in Section 4.2, we discuss the use of *rational k-simple functions* and *rational* (r,k)*-nice functions* to represent the prices stored during value iteration. Finally, Section 4.3 presents an example of the process.

To simplify the presentation, for the remainder of this section we will fix a PTA $\mathsf{P}{=}(L, l_0, \mathcal{X}, Act, \mathsf{enab}, \mathsf{prob}, \mathsf{inv}, \mathsf{price})$, target set of locations $F \subseteq L$ and let $[\![\mathsf{P}]\!]{=}(S, s_0, \mathbb{R}{\times}Act, Prob_{[\![\mathsf{P}]\!]}, Price_{[\![\mathsf{P}]\!]})$.

4.1 Computation of Expected Prices and Optimal Strategies

The first step is the construction of a *zone graph* $\mathtt{G}{=}(\mathtt{Z},\mathtt{E})$, whose vertices $\mathtt{Z}$ are *symbolic states*. A symbolic state of P is a location-zone pair (l,ζ) and represents the set of states $\{(l,v) \mid v \in \mathbb{R}^{\mathcal{X}} \wedge v \models \zeta{\wedge}\mathsf{inv}(l)\}$ of $[\![\mathsf{P}]\!]$. If $\mathtt{z}{=}(l,\zeta)$ and $\mathtt{z}'{=}(l,\zeta')$ are symbolic states, then let $\mathtt{z}{\wedge}\mathtt{z}'{=}(l,\zeta{\wedge}\zeta')$, $\mathtt{z} \subseteq \mathtt{z}'$ when $\zeta \subseteq \zeta'$ and $\mathtt{z}{=}\varnothing$ if and only if $\zeta{=}\mathtt{false}$. For any symbolic state $\mathtt{z}{=}(l,\zeta)$, locations l' and l'', action a and set of clocks R we will use the following time and discrete predecessor operations:

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

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

As in [25], we use the backwards reachability algorithm of [30] (adding action labels to the edge tuples) to build a zone graph, shown in Figure 1.

Given a zone graph $\mathtt{G}{=}(\mathtt{Z},\mathtt{E})$, for any $(l,\zeta) \in \mathtt{Z}$ let $\mathtt{E}(l,\zeta) \subseteq 2^{\mathtt{E}}$ represent the following sets of edges: $E \in \mathtt{E}(l,\zeta)$ if and only if there exists $a \in Act$ such that $\mathsf{edges}(l,a){=}\{(R_1,l_1),\ldots,(R_n,l_n)\}$ and:

$$E{=}\{(\mathtt{z},a,(R_1,l_1),\mathtt{z}_1),\ldots,(\mathtt{z},a,(R_n,l_n),\mathtt{z}_n)\}$$

for some $\mathtt{z}_1,\ldots,\mathtt{z}_n \in \mathtt{Z}$.

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

Fig. 1. Backwards reachability algorithm [30]

After building the zone graph, the next step is to find and restrict $[\![\mathsf{P}]\!]$ and $\mathtt{G}$ to include only those states for which the optimal expected price to reach the target is finite, i.e., states for which the maximum probability of reaching the target is 1 in the case of minimum expected prices and for which the minimum probability of reaching the target is 1 in the case of maximum expected prices.

Symbolic (zone-based) algorithms for performing this restriction, which extend the algorithms developed for MDPs [14,17], can be found in [25]. For the remainder of the section we suppose that $S_{\min}$ and $S_{\max}$ are the states of $[\![\mathsf{P}]\!]$ for which the minimum and maximum reachability probability is 1, and $[\![\mathsf{P}]\!]_{\min}$ and $[\![\mathsf{P}]\!]_{\max}$ are the sub-MDPs restricted to these sets of states. We will also assume that $\mathtt{G}_{\min}{=}(\mathtt{Z}_{\min}, \mathtt{E}_{\min})$ and $\mathtt{G}_{\max}{=}(\mathtt{Z}_{\max}, \mathtt{E}_{\max})$ are the restrictions of the zone graph $\mathtt{G}{=}(\mathtt{Z}, \mathtt{E})$ to these sets of states.

It follows that the restricted MDPs $[\![\mathsf{P}]\!]_{\min}$ and $[\![\mathsf{P}]\!]_{\max}$ satisfy Assumption 1, and we can therefore use Theorem 1 in the case of minimum expected pices and Corollary 1 in the case of maximum expected prices. In particular, we can use the fact that value iteration for the Bellman operators $T^{\min}_{[\![\mathsf{P}]\!]_{\min}}$ and $T^{\max}_{[\![\mathsf{P}]\!]_{\max}}$ (see Definition 3) for the target set S_F converges to the minimum and maximum expected prices, respectively, when starting from any bounded function.

Next, we present a value iteration method over the restricted zone graphs $\mathtt{G}_{\min}$ and $\mathtt{G}_{\max}$, based on the function $T^{\mathrm{opt}}_{\mathtt{G}_{\mathrm{opt}}}$, which has a direct correspondence with value iteration over the sub-MDPs $[\![\mathsf{P}]\!]_{\min}$ and $[\![\mathsf{P}]\!]_{\max}$.

Definition 6. *The operator* $T^{\mathrm{opt}}_{\mathtt{G}_{\mathrm{opt}}} : (\mathsf{Z}_{\mathrm{opt}}{\rightarrow}(S_{\mathrm{opt}}{\rightarrow}\mathbb{R})){\rightarrow}(\mathsf{Z}_{\mathrm{opt}}{\rightarrow}(S_{\mathrm{opt}}{\rightarrow}\mathbb{R}))$ *on the zone graph* $\mathtt{G}_{\mathrm{opt}}$ *is such that for* $g : \mathsf{Z}_{\mathrm{opt}}{\rightarrow}(S_{\mathrm{opt}}{\rightarrow}\mathbb{R})$, $\mathsf{z}{=}(l,\zeta) \in \mathsf{Z}_{\mathrm{opt}}$ *and* $s{=}(l,v) \in S_{\mathrm{opt}}$ *where* $s \in \mathsf{tpre}(\mathsf{z})$ *we have* $T^{\mathrm{opt}}_{\mathtt{G}_{\mathrm{opt}}}(g)(\mathsf{z})(s)$ *equals 0 if* $l \in F$ *and otherwise equals:*

$$\underset{\substack{t\in\mathbb{R}\wedge\\ v+t\in\zeta}}{\mathrm{opt}^{\star}} \; \underset{E\in\mathtt{E}(\mathsf{z})}{\mathrm{opt}} \left\{ \mathsf{price}_L(l){\cdot}t + \mathsf{price}_{Act}(l,a) \right.$$
$$\left. + \sum_{(\mathsf{z},a,(R,l'),\mathsf{z}')\in E} \mathsf{prob}(l,a)(R,l'){\cdot}g(\mathsf{z}')(l',(v{+}t)[R]) \right\} .$$

for $\mathrm{opt} \in \{\min,\max\}$, *and where* $\min^{\star} = \inf$ *and* $\max^{\star} = \sup$.

The proof of the following proposition follows directly from the proofs presented in [25] for optimal expected time computation.

Proposition 1. *For* $\mathrm{opt} \in \{\min,\max\}$, *if* $f : S_{\mathrm{opt}}{\rightarrow}\mathbb{R}$ *and* $g : \mathsf{Z}_{\mathrm{opt}}{\rightarrow}(S_{\mathrm{opt}}{\rightarrow}\mathbb{R})$ *are functions such that* $f(s) = g(\mathsf{z})(s)$ *for all* $s \in S_{\mathrm{opt}}$ *and* $\mathsf{z} \in \mathsf{Z}_{\mathrm{opt}}$ *such that* $s \in \mathsf{tpre}(\mathsf{z})$, *then for any* $s \in S_{\mathrm{opt}}$ *and* $n \in \mathbb{N}$ *we have:*

$$(T^{\mathrm{opt}}_{[\![\mathsf{P}]\!]_{\mathrm{opt}}})^n(f)(s) = \mathrm{opt}\{\, (T^{\mathrm{opt}}_{\mathtt{G}_{\mathrm{opt}}})^n(g)(\mathsf{z})(s) \mid \mathsf{z} \in \mathsf{Z}_{\mathrm{opt}} \wedge s \in \mathsf{tpre}(\mathsf{z}) \,\} .$$

Consequently, value iteration, using function $T^{\mathrm{opt}}_{\mathtt{G}_{\mathrm{opt}}}$, converges to the optimal expected reachability price for the original PTA, a result that follows from Theorem 1, Corollary 1 and Proposition 1. The final step is then to synthesise an ε-optimal deterministic, memoryless strategy for expected reachability on the PTA. This can be done by stepping through the backwards graph and selecting the time-action pairs that achieve the results returned by value iteration in each state of the zone graph.

Unlike traditional value iteration for MDPs, which iterates over real-valued vectors over states, the value iteration process for PTAs outlined above uses state vectors whose values are themselves real-valued functions. In the following section, we will show how this can be achieved using classes of functions called rational k-simple functions and rational (r,k)-nice functions.

4.2 Rational Simple Functions and Rational Nice Functions

To simplify the presentation we will assume that $\mathcal{X} = \{x_1,\dots,x_n\}$ and $k \in \mathbb{N}$ is the maximum constant appearing in P. Since P satisfies Assumption 2(a), it is bounded, and therefore all clock values appearing in $[\![\mathsf{P}]\!]$ are bounded by k. We first define polyhedra with rational time bounds.

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

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

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

*Furthermore, 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 the case of expected price reachability computation, [25] introduced the notions of *rational k-simple* and *k-nice* functions to represent the functions encountered during value iteration.

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:*

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

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

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

Definition 9. *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:*

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

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

We now extend these definitions to allow the representation of the value functions encountered when computing optimal expected price reachability using value iteration and either $T^{\min}_{\mathtt{G}_{\min}}$ or $T^{\max}_{\mathtt{G}_{\max}}$ (see Definition 6). We first extend the definition of rational k-simple functions and then consider the different operations performed by $T^{\min}_{\mathtt{G}_{\min}}$ and $T^{\max}_{\mathtt{G}_{\max}}$ and analyse their effect on the extended definition of rational k-simple functions.

Definition 10. *For zone ζ, a function $f : \zeta{\to}\mathbb{R}$ is* rational k-simple *if and only if it can be represented as:*

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

where $c_j, d_l, p_{il} \in \mathbb{Q}$ and C_j, D_l are k-polyhedra for all $1{\leq}i{\leq}n$, $1{\leq}j{\leq}M$ and $1{\leq}l{\leq}N$ for some $M, N \in \mathbb{N}$.

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

The above definition extends the k-simple functions of [25] (see Definition 8) by allowing any linear combination of clock values and allowing negative as well as non-negative rational constants. The first operation we consider for rational k-simple functions is the resetting of clocks.

Definition 11. *If $f : \zeta{\rightarrow}\mathbb{R}$ is a rational k-simple function and $R \subseteq \mathcal{X}$, let $f[R] : [R]\zeta{\rightarrow}\mathbb{R}$ be the function where $f[R](v) = f(v[R])$ for all $v \in \zeta$.*

The following lemma demonstrates that resetting clocks preserves rational simplicity.

Lemma 1. *If $f : \zeta{\rightarrow}\mathbb{R}$ is rational k-simple and $R \subseteq \mathcal{X}$, then $f[R] : [R]\zeta{\rightarrow}\mathbb{R}$ is rational k-simple.*

Proof. For any k-polyhedron C and $R \subseteq \mathcal{X}$, let $[R]C$ be the k-polyhedron $\{v \in \mathbb{R}^{\mathcal{X}} \mid v[R] \in C \wedge v(x){\leq}k \text{ for } x \in \mathcal{X}\}$. Now consider any $R \subseteq \mathcal{X}$ and rational k-simple function $f : \zeta{\rightarrow}\mathbb{R}$ such that for any $v \in \zeta$:

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

where $c_j, d_l, p_{il} \in \mathbb{Q}$ and C_j, D_l are k-polyhedra for all $1{\leq}i{\leq}n$, $1{\leq}j{\leq}M$ and $1{\leq}l{\leq}N$ for some $M, N \in \mathbb{N}$. By Definition 11, for any $v \in [R]\zeta$ we have:

$$\begin{aligned} f[R](v) &= f(v[R]) \\ &= \begin{cases} c_j & \text{if } v[R] \in C_j \\ d_l - \sum_{i=1}^{n} p_{il}{\cdot}v[R](x_i) & \text{if } v[R] \in D_l \end{cases} && \text{(by (1))} \\ &= \begin{cases} c_j & \text{if } v \in [R]C_j \\ d_l - \sum_{i=1}^{n} p_{il}{\cdot}v[R](x_i) & \text{if } v \in [R]D_l \end{cases} && \text{(by definition of } [R]C\text{)} \\ &= \begin{cases} c_j & \text{if } v \in [R]C_j \\ d_l - \sum_{i=1}^{n} p'_{il}{\cdot}v(x_i) & \text{if } v \in [R]D_l \end{cases} \end{aligned}$$

where $p'_{il}{=}0$ if $x_i \in R$ and $p'_{il}{=}p_{il}$ otherwise. It therefore follows that $f[R]$ is rational k-simple as required. □

The next operation performed by $T^{\min}_{\mathtt{G}_{\min}}$ and $T^{\max}_{\mathtt{G}_{\max}}$ builds function of the form $v \mapsto p{\cdot}t{+}p'{+}f(l,\zeta)(l,v{+}t)$. This motivates first demonstrating that adding constants (corresponding to the accumulation of action prices) preserves k-simplicity.

Lemma 2. *If $f : \zeta{\rightarrow}\mathbb{R}$ is rational k-simple and $p' \in \mathbb{Q}_+$, then $f{+}p' : \zeta{\rightarrow}\mathbb{R}$ is also rational k-simple.*

Proof. The proof follows from the definition of k-simple functions (see Definition 10). □

We now extend rational k-nice functions of [25] (see Definition 9) to (p,k)-nice functions, where the additional parameter p corresponds to the current rate at which prices are accumulated as time passes.

Definition 12. *For $p \in \mathbb{Q}_+$ and zone ζ, a function $g : (\zeta{\times}\mathbb{R}){\to}\mathbb{R}$ is* rational (p,k)-nice *if and only if it can be represented as:*

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

where $c_j, d_l, p_{il} \in \mathbb{Q}$ and F_j, G_l are rational k-bipolyhedra for all $1{\leq}i{\leq}n$, $1{\leq}j{\leq}M$ and $1{\leq}l{\leq}N$ for some $M, N \in \mathbb{N}$.

Next we show that rational k-nicety is preserved under taking convex combinations of functions of the form $v \mapsto p{\cdot}t{+}f(l,\zeta)(l, v{+}t)$.

Lemma 3. *A convex combination of rational (p,k)-nice functions is rational (p,k)-nice.*

Proof. It is sufficient to consider a binary convex combination, as any other convex combination can be rewritten as a sequence of binary convex combinations. Therefore, consider any zone ζ, rationals $\lambda, \lambda' \in \mathbb{Q}_+$ and rational (p,k)-nice functions $g, g' : (\zeta{\times}\mathbb{R}){\to}\mathbb{R}$ such that $\lambda{+}\lambda' = 1$ and for any $v \in \zeta$:

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

$$g'(v,t) = \begin{cases} c'_{j'}{+}p{\cdot}t & \text{if } (v,t) \in F'_{j'} \\ d'_{l'} - \sum_{i=1}^n p'_{il'}{\cdot}v(x_i) + (p - \sum_{i=1}^n p'_{il'})\cdot t & \text{if } (v,t) \in G'_{l'} \end{cases}$$

where $c_j, d_l, p_{il}, c'_{j'}, d'_{l'}, p'_{il'} \in \mathbb{Q}$ and $C_j, D_l, C'_{j'}, D'_{l'}$ are k-polyhedra for all $1{\leq}i{\leq}n$, $1{\leq}j{\leq}M$, $1{\leq}l{\leq}N$, $1{\leq}j'{\leq}M'$ and $1{\leq}l'{\leq}N'$ for some $M, M', N, N' \in \mathbb{N}$. Let $h : (\zeta{\times}\mathbb{R}){\to}\mathbb{R}$ be the function such that $h(v,t) = \lambda{\cdot}g(v,t) + \lambda'{\cdot}g'(v,t)$ for all $(v,t) \in \zeta{\times}\mathbb{R}$. Taking any $(v,t) \in \zeta{\times}\mathbb{R}$, we have the following four cases to consider.

- If $(v,t) \in F_j \cap F'_{j'}$ for some j and j', then

  $$h(v,t) = \lambda{\cdot}(c_j{+}p{\cdot}t){+}\lambda'{\cdot}(c'_{j'}{+}p{\cdot}t) = (\lambda{\cdot}c_j{+}\lambda'{\cdot}c'_{j'}) + p{\cdot}t$$

  since $\lambda{+}\lambda' = 1$.
- If $(v,t) \in F_j \cap G'_{l'}$ for some j and l', then

  $$\begin{aligned} h(v,t) &= \lambda{\cdot}(c_j + p{\cdot}t){+}\lambda'{\cdot}\Big(d'_{l'}{-}\textstyle\sum_{i=1}^n p'_{il'}{\cdot}v(x_i){+}(p{-}\sum_{i=1}^n p'_{il'}){\cdot}t\Big) \\ &= (\lambda{\cdot}c_j{+}\lambda'{\cdot}d'_{l'}){-}\textstyle\sum_{i=1}^n(\lambda'{\cdot}p'_{il'}){\cdot}v(x_i) + \Big(\lambda{\cdot}p{+}\lambda'{\cdot}p{-}\sum_{i=1}^n(\lambda'{\cdot}p'_{il'})\Big){\cdot}t \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(rearranging)} \\ &= (\lambda{\cdot}c_j{+}\lambda'{\cdot}d'_{l'}){-}\textstyle\sum_{i=1}^n(\lambda'{\cdot}p'_{il'}){\cdot}v(x_i) + (p{-}\sum_{i=1}^n(\lambda'{\cdot}p'_{il'})){\cdot}t \end{aligned}$$

  since $\lambda{+}\lambda' = 1$.

- If $(v,t) \in G_l \cap F'_{j'}$ for some l and j', then similarly to the above:

$$\begin{aligned} h(v,t) &= \lambda{\cdot}\Big(d_l{-}\textstyle\sum_{i=1}^n p_{il}{\cdot}v(x_i){+}\,(p{-}\textstyle\sum_{i=1}^n p_{il})\,{\cdot}t\Big){+}\lambda'{\cdot}(c'_{j'}+t) \\ &= (\lambda{\cdot}d_l{+}\lambda'{\cdot}c'_{j'}){-}\textstyle\sum_{i=1}^n(\lambda{\cdot}p_{il}){\cdot}v(x_i) + (p{-}\textstyle\sum_{i=1}^n(\lambda{\cdot}p_{il})){\cdot}t\,. \end{aligned}$$

- If $(v,t) \in G_l \cap G'_{l'}$ for some l and l', then using fact $\lambda{+}\lambda' = 1$ we have:

$$\begin{aligned} h(v,t) =\ & \lambda{\cdot}\Big(d_l{-}\textstyle\sum_{i=1}^n p_{il}{\cdot}v(x_i){+}\,(p{-}\textstyle\sum_{i=1}^n p_{il})\,{\cdot}t\Big) \\ &+ \lambda'{\cdot}\Big(d'_{l'}{-}\textstyle\sum_{i=1}^n p'_{il'}{\cdot}v(x_i){+}\,(p{-}\textstyle\sum_{i=1}^n p'_{il'})\,{\cdot}t\Big) \\ =\ & (\lambda{\cdot}d_l{+}\lambda'{\cdot}d'_{l'}){+}\textstyle\sum_{i=1}^n(\lambda{\cdot}p_{il}{+}\lambda'{\cdot}p'_{il'}){\cdot}v(x_i) + (r - \textstyle\sum_{i=1}^n(\lambda{\cdot}p_{il}{+}\lambda'{\cdot}p'_{il'}))\,{\cdot}t\,. \end{aligned}$$

As these are all the cases to consider and the intersection of k-polyhedra is a k-polyhedron, it follows that h is a rational (p,k)-nice function as required. □

After the convex combination, $T^{\min}_{\mathtt{G}_{\min}}$ and $T^{\max}_{\mathtt{G}_{\max}}$ take a minimum or maximum value respectively, and therefore we show that these operations also preserve (p,k)-nicety.

Lemma 4. *The minimum and maximum of rational (p,k)-nice functions are rational (p,k)-nice.*

Proof. We prove the case for the minimum of rational (p,k)-nice functions; the case for maximum follows similarly. Given rational (p,k)-nice functions $g, g' : (\zeta{\times}\mathbb{R}){\to}\mathbb{R}$ such that for any $(v,t) \in \zeta{\times}\mathbb{R}$:

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

$$g'(v,t) = \begin{cases} c_{j'}{+}p{\cdot}t & \text{if } (v,t) \in F'_{j'} \\ d'_{l'}{-}\,\sum_{i=1}^n p'_{il'}{\cdot}v(x_i){+}\,(p{-}\,\sum_{i=1}^n p'_{il'})\,{\cdot}t & \text{if } (v,t) \in G'_{l'} \end{cases}$$

where $c_j, d_l, p_{il}, c'_{j'}, d'_{l'}, p'_{il'} \in \mathbb{Q}$ and $C_j, D_l, C'_{j'}, D'_{l'}$ are k-polyhedra for all $1{\le}i{\le}n$, $1{\le}j{\le}M$, $1{\le}l{\le}N$, $1{\le}j'{\le}M'$ and $1{\le}l'{\le}N'$ for some $M, M'N, N' \in \mathbb{N}$. Letting $h = \min\{g, g'\}$ and considering $h(v,t)$ for any $(v,t) \in \zeta{\times}\mathbb{R}$, we have the following four cases to consider.

- If $(v,t) \in F_j \cap F'_{j'}$ for some j and j', then

$$h(v,t) = \begin{cases} c_j{+}p{\cdot}t & \text{if } (v,t) \in F_j \cap H \\ c_{j'}{+}p{\cdot}t & \text{if } (v,t) \in F'_{j'} \cap H' \end{cases}$$

  where $H = \{(v,t) \in \zeta{\times}\mathbb{R} \mid c_j{+}p{\cdot}t \le c'_{j'}{+}p{\cdot}t\} = \{(v,t) \in \zeta{\times}\mathbb{R} \mid c_j \le c'_{j'}\}$ and similarly $H' = \{(v,t) \in \zeta{\times}\mathbb{R} \mid c'_{j'} \le c_j\}$.

– If $(v,t) \in F_j \cap G'_{l'}$ for some j and l', then

$$h(v,t) = \begin{cases} c_j{+}p{\cdot}t & \text{if } (v,t) \in F_j \cap H \\ d'_{l'}{-}\sum_{i=1}^{n} p'_{il'}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^{n} p'_{il'})\cdot t & \text{if } (v,t) \in G'_{l'} \cap H' \end{cases}$$

where

$$\begin{aligned} H &= \{(v,t) \in \zeta{\times}\mathbb{R} \mid c_j{+}p{\cdot}t \leq d'_{l'}{-}\textstyle\sum_{i=1}^{n} p'_{il'}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^{n} p'_{il'})\cdot t\} \\ &= \{(v,t) \in \zeta{\times}\mathbb{R} \mid \textstyle\sum_{i=1}^{n} p'_{il'}{\cdot}(v(x_i){+}t) \leq d'_{l'}{-}c_j\} && \text{(rearranging)} \\ &= \{(v,t) \in \zeta{\times}\mathbb{R} \mid \textstyle\sum_{i=1}^{n} p'_{il'}{\cdot}(v{+}t)(x_i) \leq d'_{l'}{-}c_j\} && \text{(by definition of } v{+}t) \end{aligned}$$

and similarly $H' = \{(v,t) \in \zeta{\times}\mathbb{R} \mid \sum_{i=1}^{n} {-}p'_{il'}{\cdot}(v{+}t)(x_i) \leq c_j{-}d'_{l'}\}$.

– If $(v,t) \in G_l \cap F'_{j'}$ for some l and j', then

$$h(v,t) = \begin{cases} d_l{-}\sum_{i=1}^{n} p_{il}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^{n} p_{il})\cdot t & \text{if } (v,t) \in G_l \cap H \\ c_{j'}{+}t & \text{if } (v,t) \in F'_{j'} \cap H' \end{cases}$$

and by a similar reduction to the case above we have:

$$\begin{aligned} H &= \{(v,t) \in \zeta{\times}\mathbb{R} \mid \textstyle\sum_{i=1}^{n} {-}p_{il}{\cdot}(v{+}t)(x_i) \leq c_{j'}{-}d_l\} \\ H' &= \{(v,t) \in \zeta{\times}\mathbb{R} \mid \textstyle\sum_{i=1}^{n} p_{il}{\cdot}(v{+}t)(x_i) \leq d_l{-}c_{j'}\}\,. \end{aligned}$$

– If $(v,t) \in G_l \cap G'_{l'}$ for some l and l', then

$$h(v,t) = \begin{cases} d_l{-}\sum_{i=1}^{n} p_{il}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^{n} p_{il})\cdot t & \text{if } (v,t) \in G_l \cap H \\ d'_{l'}{-}\sum_{i=1}^{n} p'_{il'}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^{n} p'_{il'})\cdot t & \text{if } (v,t) \in G'_{l'} \cap H' \end{cases}$$

where

$$\begin{aligned} H = \{(v,t) \in \zeta{\times}\mathbb{R} \mid\ & d_l{-}\textstyle\sum_{i=1}^{n} p_{il}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^{n} p_{il})\cdot t \\ & \leq d'_{l'}{-}\textstyle\sum_{i=1}^{n} p'_{il'}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^{n} p'_{il'})\cdot t\} \\ = \{(v,t) \in \zeta{\times}\mathbb{R} \mid\ & \textstyle\sum_{i=1}^{n}(p'_{il'}{-}p_{il}){\cdot}v(x_i){+}\sum_{i=1}^{n}(p'_{il'}{-}p_{il}){\cdot}t \leq d'_{l'}{-}d_l\} \\ & \text{(rearranging)} \\ = \{(v,t) \in \zeta{\times}\mathbb{R} \mid\ & {-}\textstyle\sum_{i=1}^{n}(p'_{il'}{-}p_{il}){\cdot}(v(x_i){+}t) \leq d'_{l'}{-}d_l\} \quad \text{(rearranging again)} \\ = \{(v,t) \in \zeta{\times}\mathbb{R} \mid\ & {-}\textstyle\sum_{i=1}^{n}(p'_{il'}{-}p_{il}){\cdot}(v{+}t)(x_i) \leq d'_{l'}{-}d_l\} \end{aligned}$$

by definition of $v{+}t$ and similarly we have:

$$H' = \{(v,t) \in \zeta{\times}\mathbb{R} \mid {-}\textstyle\sum_{i=1}^{n}(p_{il}{-}p'_{il'}){\cdot}(v{+}t)(x_i) \leq d_l{-}d'_{l'}\}\,.$$

Since in each case H and H' are k-bipolyhedra, if follows from Definition 12 that the lemma holds. □

The final operations performed by $T^{\min}_{\mathtt{G}_{\min}}$ and $T^{\max}_{\mathtt{G}_{\max}}$ concern taking the infimum or supremum over t of a function of the form $v \mapsto p{\cdot}t{+}f(l,\zeta)(l,v{+}t)$. Hence, we now show that performing either of these operations on a rational (p,k)-nice function returns a rational k-simple function.

Lemma 5. *For any zone ζ, if $g : (\zeta{\times}\mathbb{R}){\to}\mathbb{R}$ is rational (p,k)-nice, then the functions $f_1 : \zeta{\to}\mathbb{R}$ and $f_2 : \zeta{\to}\mathbb{R}$ where $f_1(v) = \inf_{t\in\mathbb{R}} g(v,t)$ and $f_2(v) = \sup_{t\in\mathbb{R}} g(v,t)$ for $v \in \zeta$ are rational k-simple.*

Proof. We prove the case for f_1; the case for f_2 follows similarly (swapping Δ^- and Δ^+). Consider any zone ζ and rational (p,k)-nice function $g : (\zeta{\times}\mathbb{R}){\to}\mathbb{R}$. By Definition 12, for any $(v,t) \in \zeta{\times}\mathbb{R}$, we have:

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

where $c_j, d_l, p_{il} \in \mathbb{Q}$ and

$$F_j = \{(v,t) \mid v \in C_j \wedge v{+}t \in C'_j\} \quad \text{and} \quad G_l = \{(v,t) \mid v \in D_l \wedge v{+}t \in D'_l\}$$

for some k-polyhedra C_j, C'_j, D_l and D'_l for all $1{\le}i{\le}n$, $1{\le}j{\le}M$ and $1{\le}l{\le}N$ for some $M, N \in \mathbb{N}$.

For any k-polyhedron C, let

$$\Delta^-(v,C) \stackrel{\text{def}}{=} \inf\{t \mid v{+}t \in C\} \quad \text{and} \quad \Delta^+(v,C) \stackrel{\text{def}}{=} \sup\{t \mid v{+}t \in C\}\,.$$

Following the arguments of [4], it follows that the functions $\Delta^-(\cdot,C) : \zeta{\to}\mathbb{R}$ and $\Delta^+(\cdot,C) : \zeta{\to}\mathbb{R}$ are both k-simple over k-polyhedra. If $f_1(v) = \inf_{t\in\mathbb{R}} g(v,t)$, for any $v \in \zeta$ we have $f_1(v)$ equals:

$$\begin{cases} c_j & \text{if } v \in C_j \cap C'_j \\ c_j + p{\cdot}\Delta^-(v,C'_j) & \text{if } v \in C_j \setminus C'_j \\ d_l{-}\sum_{i=1}^n p_{il}{\cdot}v(x_i) & \text{if } v \in D_l \cap D'_l \text{ and } p{-}\sum_{i=1}^n p_{il}{\ge}0 \\ d_l{-}\sum_{i=1}^n p_{il}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^n p_{il})\cdot\Delta^-(v,D'_l) & \text{if } v \in D_l \setminus D'_l \text{ and } p{-}\sum_{i=1}^n p_{il}{\ge}0 \\ d_l{-}\sum_{i=1}^n p_{il}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^n p_{il})\cdot\Delta^+(v,D'_l) & \text{if } v \in D_l \text{ and } p{-}\sum_{i=1}^n p_{il}{<}0 \end{cases}$$

In all except the final two cases, since $\Delta^-(\cdot,C) : \zeta{\to}\mathbb{R}$ is k-simple, it follows that f_1 is rational k-simple. Considering the penultimate case, by definition of k-simple functions we have the following two cases to consider.

- if $\Delta^-(v,D'_l) = d'_l$ for some $d'_l \in \mathbb{Q}_+$, then for any $v \in D_l \setminus D'_l$:

$$\begin{aligned} f_1(v) &= d_l{-}\textstyle\sum_{i=1}^n p_{il}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^n p_{il})\cdot\Delta^-(v,D'_l) \\ &= \big(d_l + (p{-}\textstyle\sum_{i=1}^n p_{il})\cdot d'_l\big){-}\textstyle\sum_{i=1}^n p_{il}{\cdot}v(x_i) && \text{(rearranging)} \end{aligned}$$

  which is rational k-simple, since g is rational (p,k)-nice.
- if $\Delta^-(v,D'_l) = d'_l{-}v(x_{i'_l})$ for some $d'_l \in \mathbb{Q}_+$ and $1{\le}i'_l{\le}n$, then for any $v \in D_l \setminus D'_l$:

$$\begin{aligned} f_1(v) &= d_l{-}\textstyle\sum_{i=1}^n p_{il}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^n p_{il})\cdot\Delta^-(v,D'_l) \\ &= d_l{-}\textstyle\sum_{i=1}^n p_{il}{\cdot}v(x_i){+}\,(p{-}\sum_{i=1}^n p_{il})\cdot(d'_l{-}v(x_{i'_l})) && \text{(rearranging)} \\ &= \Big(d_l + (p{-}\textstyle\sum_{i=1}^n p_{il})\cdot d'_l\Big){-}\textstyle\sum_{i=1}^n p'_{il}{\cdot}v(x_i) \end{aligned}$$

  where $p'_{il} = p_{il}{+}(p{-}\sum_{i=1}^n p_{il})$ if $i{=}i'_l$ and $p'_{il}{=}p_{il}$ otherwise.

The final case follows similarly to the penultimate using the fact $\varDelta^+(v, D'_l)$ is a k-simple function. Therefore, we can conclude that f_1 is rational k-simple as required. □

In the related proof of [25] we see that, for minimum expected time computation, it is always optimal to let as little time pass as possible in the current polyhedron and, for maximum expected time computation, it is always optimal to let as much time pass as possible. However, for prices, we see that this is not always the case, e.g., $\varDelta^+(v, C)$ is used in the computation of minimum expected prices. This is due to the fact that price rates in locations reached at a later stage can be higher, and in such cases it can be optimal to let time pass now and accumulate a lower price, as opposed to waiting and accumulating a higher price later.

We now combine the above results and show that rational k-simple functions are a suitable representation for value functions when computing optimal expected time using value iteration and either $T^{\min}_{\mathtt{G}_{\min}}$ or $T^{\max}_{\mathtt{G}_{\max}}$.

Proposition 2. *For* opt $\in \{\min, \max\}$*, if* $f : \mathsf{Z}_{\text{opt}} \to (S_{\text{opt}} \to \mathbb{R})$ *is a rational k-simple function, then* $T^{\text{opt}}_{\mathtt{G}_{\text{opt}}}(f)$ *is rational k-simple.*

Proof. We only the consider when opt $=$ min, the case when opt $=$ max follows similarly. Consider any rational k-simple function, $\mathsf{z}{=}(l, \zeta) \in \mathsf{Z}_{\min}$ and $E \in \mathtt{E}(\mathsf{z})$. For any $v \in \mathbb{R}^{\mathcal{X}}$ and $t \in \mathbb{R}$ and letting $r = \mathsf{price}_L(l)$ and $p' = \mathsf{price}_{Act}(l, a)$:

$$
\begin{aligned}
& p{\cdot}t + p' + \textstyle\sum_{(\mathsf{z},a,(R,l'),\mathsf{z}_{(R,l')})\in E} \mathsf{prob}(l,a)(R,l')\cdot f(\mathsf{z}_{(R,l')})(l',(v{+}t)[R]) \\
&\quad = p{\cdot}t + p' + \textstyle\sum_{(\mathsf{z},a,(R,l'),\mathsf{z}_{(R,l')})\in E} \mathsf{prob}(l,a)(R,l')\cdot f[R](\mathsf{z}_{(R,l')})(l',v{+}t) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by Definition 11)} \\
&\quad = \textstyle\sum_{(\mathsf{z},a,(R,l'),\mathsf{z}_{(R,l')})\in E} \mathsf{prob}(l,a)(R,l')\cdot \big(p{\cdot}t + p' + f[R](\mathsf{z}_{(R,l')})(l',v{+}t)\big) \qquad (2)
\end{aligned}
$$

since $\mathsf{prob}(l, a)$ is a distribution. By construction, f is rational k-simple, and hence for any $(\mathsf{z}, a, (R, l'), \mathsf{z}_{(R,l')}) \in E$ using Lemma 1 and Lemma 2 we have that $p' + f[R]$ is also rational k-simple. Using Definition 12 it follows that:

$$(v,t) \mapsto p{\cdot}t + p' + f[R](\mathsf{z}_{(R,l')})(l', v{+}t)$$

is rational (p, k)-nice. Thus, since $(\mathsf{z}, a, (R, l'), \mathsf{z}_{(R,l')}) \in E$ was arbitrary, using Lemma 3 and (2) we have that:

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

is also rational (p, k)-nice. Since $E \in \mathtt{E}(\mathsf{z})$ was arbitrary and $\mathtt{E}(\mathsf{z})$ is finite, Lemma 4 tells us:

$$(v,t) \mapsto \min_{E\in\mathtt{E}(\mathsf{z})} \left\{ p{\cdot}t{+}p'{+} \sum_{(\mathsf{z},a,(R,l'),\mathsf{z}_{(R,l')})\in E} \mathsf{prob}(l,a)(R,l')\cdot f(\mathsf{z}_{(R,l')})(l',(v{+}t)[R]) \right\}$$

is again rational (p, k)-nice. Finally, using Definition 6 and Lemma 5, it follows that $T_{\mathtt{G}}(f)(\mathsf{z})$ is rational k-simple as required. □

Proposition 2 tells us that value iteration over a zone graph to compute expected prices, as specified in Definition 6, can be performed using rational k-simple functions (and rational (p, k)-nice functions).

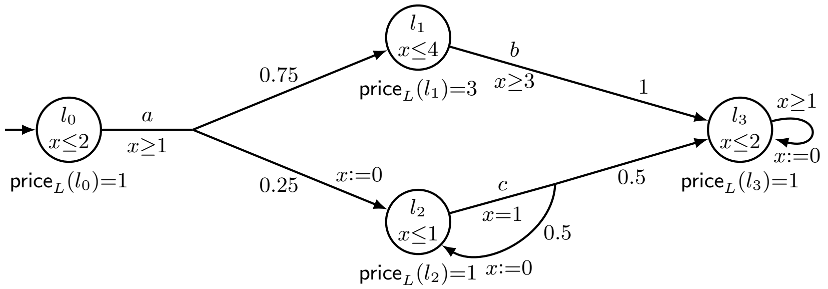

Fig. 2. Example PTA P

4.3 Example

Figure 2 shows an example of a linearly-priced PTA. Location prices are indicated next to each location; all action prices are zero so they are omitted from the figure. For this example, we consider the target set $F = \{l_2\}$ and compute both the minimum and maximum expected price of reaching F. For this PTA, all states reach the target with minimum (and maximum) probability 1, and therefore the zone graphs used for minimum and maximum expected price computation are the same and equal that constructed using the algorithm presented in Figure 1. This zone graph is shown in Figure 3.

In the case of the minimum expected price, performing value iteration over the zone graph $\mathtt{G}$ of Figure 3 gives, for $n{\geq}\ 3$:

$$\begin{aligned}
(T^{\min}_{\mathtt{G}})^n(f_0)(\mathsf{z}_0)(l_0,v) &= \Big(2 + 0.5\cdot \Big(3 + \textstyle\sum_{i=0}^{n-3} 0.25^i\Big)\Big) - v(x) \\
(T^{\min}_{\mathtt{G}})^n(f_0)(\mathsf{z}_1)(l_1,v) &= \begin{cases} 9{-}3{\cdot}v(x) & \text{if } v(x){\leq}3 \\ 0 & \text{if } 3{\leq}v(x){\leq}4 \end{cases} \\
(T^{\min}_{\mathtt{G}})^n(f_0)(\mathsf{z}_2)(l_2,v) &= \textstyle\sum_{i=0}^{n-2} 0.25^i - v(x) \\
(T^{\min}_{\mathtt{G}})^n(f_0)(\mathsf{z}_3)(l_3,v) &= 0
\end{aligned}$$

It then follows that the minimum expected price to reach the target from the initial state equals 4.166667. On the other hand, for the maximum expected price, performing value iteration yields for $n{\geq}3$:

$$\begin{aligned}
(T^{\max}_{\mathtt{G}})^n(f_0)(\mathsf{z}_0)(l_0,v) &= \begin{cases} \Big(1 + 0.5\cdot \Big(9 + \sum_{i=0}^{n-3} 0.25^i\Big)\Big) - v(x) & \text{if } x{\leq}1 \\ 0.5\cdot \Big(12 + \sum_{i=0}^{n-3} 0.25^i\Big) - 3{\cdot}v(x) & \text{if } 1{\leq}x{\leq}2 \end{cases} \\
(T^{\max}_{\mathtt{G}})^n(f_0)(\mathsf{z}_1)(l_1,v) &= 12{-}3{\cdot}v(x) \\
(T^{\max}_{\mathtt{G}})^n(f_0)(\mathsf{z}_2)(l_2,v) &= \textstyle\sum_{i=0}^{n-2} 0.25^i - v(x) \\
(T^{\max}_{\mathtt{G}})^n(f_0)(\mathsf{z}_3)(l_3,v) &= 0
\end{aligned}$$

and hence the maximum expected price for the initial state is 6.166667.

The optimal strategy for the minimum expected price is to always perform an action as soon as it is enabled. The choices of the optimal strategy for the

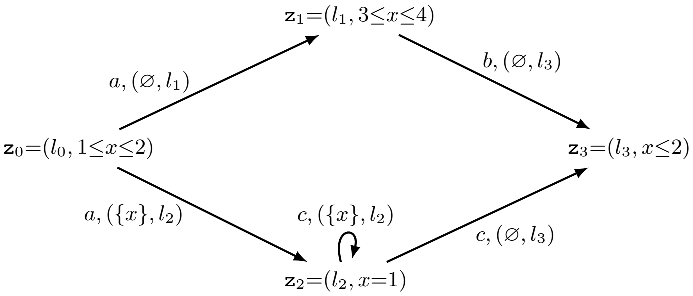
$\mathbf{z}_1{=}(l_1, 3{\le}x{\le}4)$
$a, (\varnothing, l_1)$
$b, (\varnothing, l_3)$
$\mathbf{z}_0{=}(l_0, 1{\le}x{\le}2)$
$\mathbf{z}_3{=}(l_3, x{\le}2)$
$a, (\{x\}, l_2)$
$c, (\{x\}, l_2)$
$c, (\varnothing, l_3)$
$\mathbf{z}_2{=}(l_2, x{=}1)$

Fig. 3. Backwards zone graph $\mathtt{G}$ for PTA of Figure 2 and target set $\{l_3\}$

maximum expected price are to leave l_0 as soon as the action a is enabled, as this allows it to remain longer in l_1, yielding a higher overall expected price.

5 Conclusions

We have extended the techniques of [25] for the symbolic computation of optimal expected time and strategy synthesis to expected prices for linearly-priced probabilistic timed automata. The approach involves building the backwards zone graph of the PTA under study and then performing value iteration over this graph. We have demonstrated that an extension of simple and nice functions over rational valued polyhedra provide an effective representation of the value functions required for this computation. One restriction that we impose on the linearly-priced PTAs we consider is that all location prices are positive. We note that it should be possible to remove this restriction by extending the algorithms of [17] for removing zero-priced end components for finite state MDPs to linearly-priced PTAs.

As already mentioned in [25], an important next step is to perform a rigorous investigation into the advantages and disadvantages of our approach in comparison with the digital clocks method [28]. This will require implementing the algorithms introduced here, for example using the Parma Polyhedra Library [5], which includes efficient ways of manipulating convex polyhedra and has already been used effectively to implement a number of real-time verification algorithms. Finally, we also plan to investigate policy iteration since it converges to optimal expected prices (see Theorem 1 and Corollary 1).

Acknowledgments. This work was partly supported by the EPSRC Mobile Autonomy Programme Grant EP/M019918/1 and the PRINCESS project, funded by the DARPA BRASS programme.

References

1. R. Alur, C. Courcoubetis, and D. Dill. Model checking in dense real time. *Information and Computation*, 104(1):2–34, 1993.

2. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.
3. R. Alur, S. La Torre, and G. Pappas. Optimal paths in weighted timed automata. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Proc. 4th Int. Workshop Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *LNCS*, pages 49–62. Springer, 2001.
4. E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In F. Vaandrager and J. van Schuppen, editors, *Proc. 2nd Int. Workshop Hybrid Systems: Computation and Control (HSCC'99)*, volume 1569 of *LNCS*, pages 19–30. Springer, 1999.
5. R. Bagnara, P. 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.
6. D. Beauquier. On probabilistic timed automata. *Theoretical Computer Science*, 292(1):65–84, 2003.
7. G. Behrmann, A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for linearly priced timed automata. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Proc. 4th Int. Workshop Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *LNCS*, pages 147–162. Springer, 2001.
8. R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.
9. J. Berendsen, T. Chen, and D. Jansen. Undecidability of cost-bounded reachability in priced probabilistic timed automata. In J. Chen and S. Cooper, editors, *Proc. 6th Int. Conf. Theory and Applications of Models of Computation (TAMC'09)*, volume 5532 of *LNCS*, pages 128–137. Springer, 2009.
10. J. Berendsen, D. Jansen, and J.-P. Katoen. Probably on time and within budget – On reachability in priced probabilistic timed automata. In *Proc. 3rd Int. Conf. Quantitative Evaluation of Systems (QEST'06)*, pages 311–322. IEEE Press, 2006.
11. J. Berendsen, D. Jansen, and F. Vaandrager. Fortuna: Model checking priced probabilistic timed automata. In *Proc. 7th Int. Conf. Quantitative Evaluation of Systems (QEST'10)*, pages 273–281. IEEE Press, 2010.
12. D. Bertsekas. *Dynamic Programming and Optimal Control*, Volumes 1 and 2. Athena Scientific, 1995.
13. D. Bertsekas and J. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.
14. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In P. Thiagarajan, editor, *Proc. 15th Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS'95)*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.
15. H. Bohnenkamp, P. D'Argenio, H. Hermanns, and J.-P. Katoen. Modest: A compositional modeling formalism for hard and softly timed systems. *IEEE Trans. Software Engineering*, 32(10):812–830, 2006.
16. A. David, P. Jensen, K. Larsen, A. Legay, D. Lime, M. Sørensen, and J. Taankvist. On time with minimal expected cost! In F. Cassez and J. Raskin, editors, *Proc. 12th Int. Symp. Automated Technology for Verification and Analysis (ATVA'14)*, volume 8837 of *LNCS*, pages 129–145. Springer, 2014.
17. L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In J. Baeten and S. Mauw, editors, *Proc. 10th Int. Conf. Concurrency Theory (CONCUR'99)*, volume 1664 of *LNCS*, pages 66–81. Springer, 1999.

18. M. Duflot, M. Kwiatkowska, G. Norman, and D. Parker. A formal analysis of Bluetooth device discovery. *Int. Journal on Software Tools for Technology Transfer*, 8(6):621–632, 2006.
19. H. Gregersen and H. Jensen. Formal design of reliable real time systems. Master's thesis, Department of Mathematics and Computer Science, Aalborg University, 1995.
20. A. Hartmanns and H. Hermanns. A modest approach to checking probabilistic timed automata. In *Proc. 6th International Conference on Quantitative Evaluation of Systems (QEST'09)*, pages 187–196. IEEE Press, 2009.
21. T. Henzinger, Z. Manna, and A. Pnueli. What good are digital clocks? In W. Kuich, editor, *Proc. 19th Int. Coll. Automata, Languages and Programming (ICALP'92)*, volume 623 of *LNCS*, pages 545–558. Springer, 1992.
22. T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.
23. H. James and E. Collins. An analysis of transient Markov decision processes. *Journal of Applied Probability*, 43 (3):603–621, 2006.
24. A. Jovanovic, M. Kwiatkowska, and G. Norman. Symbolic minimum expected time controller synthesis for probabilistic timed automata. In S. Sankaranarayanan and E. Vicario, editors, *Proc. 13th Int. Conf. Formal Modeling and Analysis of Timed Systems (FORMATS'15)*, volume 9268 of *LNCS*, pages 140–155. Springer, 2015.
25. A. Jovanovic, M. Kwiatkowska, G. Norman, and Q. Peyras. Symbolic optimal expected time reachability computation and controller synthesis for probabilistic timed automata. *Theoretical Computer Science*, 669:1–21, 2017.
26. J. Kemeny, J. Snell, and A. Knapp. *Denumerable Markov Chains.* Springer, 1976.
27. M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proc. 23rd Int. Conf. Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.
28. M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston. Performance analysis of probabilistic timed automata using digital clocks. *Formal Methods in System Design*, 29:33–78, 2006.
29. M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 282:101–150, 2002.
30. M. Kwiatkowska, G. Norman, J. Sproston, and F. Wang. Symbolic model checking for probabilistic timed automata. *Information and Computation*, 205(7):1027–1077, 2007.
31. K. Larsen, P. Pettersson, and W. Yi. Uppaal in a Nutshell. *International Journal on Software Tools for Technology Transfer*, 1:134–152, 1997.
32. K. Larsen and P. P. W. Yi. Model-checking for real-time systems. In H. Reichel, editor, *Proc. 10th Int. Conf. Fundamentals of Computation Theory (FCT95)*, volume 965 of *LNCS*, pages 62–88. Springer, 1995.
33. S. Tripakis. *The analysis of timed systems in practice.* PhD thesis, Université Joseph Fourier, Grenoble, 1998.
34. S. Tripakis. Verifying progress in timed systems. In J.-P. Katoen, editor, *Proc. 5th Int. AMAST Workshop Real-Time and Probabilistic Systems (ARTS'99)*, volume 1601 of *LNCS*, pages 299–314. Springer, 1999.
35. S. Tripakis, S. Yovine, and A. Bouajjani. Checking timed Büchi automata emptiness efficiently. *Formal Methods in System Design*, 26(3):267–292, 2005.