Concavely-Priced Timed Automata
(Extended Abstract)

Marcin Jurdziński and Ashutosh Trivedi

Department of Computer Science, University of Warwick, UK

Abstract. Concavely-priced timed automata, a generalization of linearly-priced timed automata, are introduced. Computing the minimum value of a number of cost functions—including reachability price, discounted price, average time, average price, price-per-time average, and price-per-reward average—is considered in a uniform fashion for concavely-priced timed automata. All the corresponding decision problems are shown to be PSPACE-complete. This paper generalises the recent work of Bouyer et al. on deciding the minimum reachability price and the minimum ratio-price for linearly-priced timed automata.

A new type of a region graph—the boundary region graph—is defined, which generalizes the corner-point abstraction of Bouyer et al. A broad class of cost functions—concave-regular cost functions—is introduced, and the boundary region graph is shown to be a correct abstraction for deciding the minimum value of concave-regular cost functions for concavely-priced timed automata.

1 Introduction

A system is *real-time*, if the correctness of some of its operations critically depend on the time at which they are performed. Numerous safety-critical systems are real-time, including medical systems such as heart pacemakers, and industrial process controllers such as nuclear reactor protective systems. Ensuring the correctness of real-time systems is of paramount importance. Timed automata [4] are a popular and well-established formalism for modelling real-time systems.

A timed automaton is a finite automaton accompanied by a finite set of real-valued variables called *clocks*. The states of a timed automaton are traditionally called *locations*, and the configurations of a timed automaton, consisting of a location and a valuation of clock variables, are called states. A state is called a *corner state*, if the values of all clock variables are integers. Clock variables may appear in guards of transitions of a timed automaton, where they can be compared against integers. The syntax of timed automata also allows clock values to be reset to zero after executing a transition.

Given a timed automaton and an initial state, the reachability problem is to decide whether there exists a run starting from the initial state and leading to a final state of the timed automaton. The safety-violation property of a real-time system can be modeled as a reachability problem on a timed automaton. The reachability problem for timed automata is in PSPACE (in fact, it is PSPACE-complete) by a reduction to the non-emptiness problem for a finite state automaton, called the *region graph*, whose size is exponential in the size of a timed automaton [4,2]. A natural optimization problem over timed automata is to minimize time to reach a final state. Minimum- and maximum-time

F. Cassez and C. Jard (Eds.): FORMATS 2008, LNCS 5215, pp. 48–62, 2008.

reachability was shown to be decidable in [14]. It was shown to be PSPACE-complete in [3,16]. A more general problem of two-player reachability-time games was shown to be decidable in [6] and proved EXPTIME-complete in [15,13]. An efficient algorithm to solve minimum-time reachability problem on timed automata appeared in [19], where the initial state was restricted to be a corner state.

The *linearly-priced* timed automata [7] (LPTA), also known as weighted timed automata, are an extension of timed automata, and are quite useful in modeling scheduling problems [11,1] for real-time systems. A linearly-priced timed automaton augments a timed automaton with price information, such that the price of waiting in a location is proportional to the waiting time, hence the name linearly-priced timed automata. The problem of finding a minimum price (time) schedule can be modeled as a minimum reachability-price problem over a linearly-priced timed automaton. This problem is known to be a PSPACE-complete [10] if the start state is a corner state. Alur et al. [5] give an EXPTIME algorithm to solve the problem with an arbitrary initial state by giving a non-trivial extension of the region graph. Larsen et al. [17,7] give a symbolic algorithm to solve the problem, although with some restrictions on the initial state (a corner state with all clocks set to zero). Note that PSPACE-hardness results hold for timed automata with at least three clocks. For timed automata with one clock, reachability-time and reachability-price problems are known to be NL-complete, while the complexity of these problems for two-clock timed automata remains open.

On the other hand, the problem of finding a minimum-price (or minimum-time) infinite schedule can be modelled by the minimum average-price problem on a priced timed automaton. Since the total price of an infinite run can be unbounded, a natural measure of optimality is average price per transition or average price per time unit. Bouyer et al. [11] show that a more general problem of average reward per transition over multi-priced timed automata is PSPACE-complete if the start state is a corner state.

In this paper we introduce the *concavely-priced* timed automata, a generalization of the linearly-priced timed automata, which arguably can be used to model a larger class of scheduling problems. The definition of a concavely-priced timed automaton is such that it allows price functions which are concave in a certain sense. In this paper, we show that deciding the minimum value of the reachability price, discounted price, average time, average price, price-per-time average, and price-per-reward average is PSPACE-complete for arbitrary start states (i.e., including non-corner states).

2 Optimization Problems on Finite Priced Graphs

Consider a finite graph $G = (S, E, F)$, where S is a set of states, $E \subseteq S \times S$ is a set of directed edges, such that every state has at least one outgoing edge, and $F \subseteq S$ is a set of final states. An *run* (path) in G is a sequence $\langle s_0, s_1, s_2, \ldots \rangle \in S^\omega$, such that for all $i \geq 1$, we have $(s_{i-1}, s_i) \in E$. We write Runs and Runs_{fin} for the sets of infinite and finite runs, respectively, and we write $\text{Runs}(s)$ and $\text{Runs}_{\text{fin}}(s)$ for the sets of infinite and finite runs starting from state $s \in S$, respectively. For a run $r = \langle s_0, s_1, s_2, \ldots \rangle$, we define $\text{Stop}(r) = \inf_{i \geq 0}\{i \,:\, s_i \in F\}$.

Let $\text{Cost} : \text{Runs} \to \mathbb{R}$ be a cost function that for every run $r \in \text{Runs}$ determines its cost $\text{Cost}(r)$. We then define the *minimum cost* function $\text{Cost}_* : S \to \mathbb{R}$, by $\text{Cost}_*(s) =$

$\inf_{r \in \text{Runs}(s)} \text{Cost}(r)$. The *minimization problem* for that cost function Cost is: given a state $s \in S$ and a number $D \in \mathbb{Q}$, determine whether $\text{Cost}_*(s) \leq D$.

A *priced graph* (G, π) consists of a graph G and a price function $\pi : E \to \mathbb{R}$; and a *price-reward graph* (G, π, ϱ) consists of a graph G and price and reward functions $\pi, \varrho : E \to \mathbb{R}$, respectively. Let $r = \langle s_0, s_1, s_2, \ldots \rangle \in \text{Runs}$, and for every $n \geq 1$, let $\pi_n(r) = \sum_{i=1}^{n} \pi(s_{i-1}, s_i)$ and $\varrho_n(r) = \sum_{i=1}^{n} \varrho(s_{i-1}, s_i)$. We further assume that the graph is reward-diverging i.e. $|\varrho_n(r)| \to \infty$ as $n \to \infty$.

The following list of cost functions gives rise to a number of corresponding minimization problems.

1. *Reachability:* $\text{Reach}(r) = \pi_N(r)$ if $N = \text{Stop}(r) < \infty$, and $\text{Reach}(r) = \infty$ otherwise.
2. *Discounted:* $\text{Discounted}(\lambda)(r) = (1 - \lambda) \sum_{i=1}^{\infty} \lambda^{i-1} \pi(s_{i-1}, s_i)$, where $\lambda \in (0, 1)$ is the *discount factor*.
3. *Average price:* $\text{AvgPrice}(R) = \limsup_{n \to \infty} \pi_n(r)/n$.
4. *Price-per-reward average:* $\text{PriceRewardAvg}(r) = \limsup_{n \to \infty} \pi_n(r)/\varrho_n(r)$.

A *positional strategy* is a function $\sigma : S \to S$, such that for every $s \in S$, we have $(s, \sigma(s)) \in E$. We write Σ for the set of positional strategies. A run from state $s \in S$ according to strategy σ is the unique run $\text{Run}(s, \sigma) = \langle s_0, s_1, s_2, \ldots \rangle$, such that $s_0 = s$, and for every $i \geq 1$, we have $\sigma(s_{i-1}) = s_i$. A positional strategy is *optimal* for a cost function $\text{Cost} : \text{Runs} \to \mathbb{R}$ if for every state $s \in S$, we have $\text{Cost}_*(s) = \text{Cost}(\text{Run}(s, \sigma))$. Observe that existence of an optimal positional strategy for a cost function means that, from every starting state, there is a run that minimizes the cost, and that is a simple path leading to a simple cycle.

Theorem 1. *For every finite priced graph, and for each of the reachability, discounted, and average price cost functions, there is an optimal positional strategy.*

For every finite price-reward graph, there is an optimal positional strategy for the price-per-reward average cost function.

3 Concavely-Priced Timed Automata

We assume that, wherever appropriate, sets $\mathbb{N}$ of non-negative integers and $\mathbb{R}$ of reals contain a maximum element ∞, and we write $\mathbb{N}_+$ for the set of positive integers and $\mathbb{R}_\oplus$ for the set of non-negative reals. For $n \in \mathbb{N}$, we write $[\![n]\!]_\mathbb{N}$ for the set $\{0, 1, \ldots, n\}$, and $[\![n]\!]_\mathbb{R}$ for the set $\{r \in \mathbb{R} \,:\, 0 \leq r \leq n\}$ of non-negative reals bounded by n. For a real number $r \in \mathbb{R}$, we write $|r|$ for its absolute value, we write $\lfloor r \rfloor$ for its integer part, i.e., the largest integer $n \in \mathbb{N}$, such that $n \leq r$, and we write $\wr r \int$ for its fractional part, i.e., we have $\wr r \int = r - \lfloor r \rfloor$. For sets X and Y, we write $[X \to Y]$ for the set of functions $F : X \to Y$, and $[X \rightharpoondown Y]$ for the set of partial functions $F : X \rightharpoondown Y$. For a function $f : X \to Y$ we write $\text{dom}(f)$ for the domain of function f.

For a point $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, we define its norm by $\|x\| = \max_{i=1}^{n} |x_i|$. For a point $x \in \mathbb{R}^n$, the open ball $B(x, r)$, with the center x and the radius $r > 0$, is defined by $B(x, r) = \{y \,:\, y \in \mathbb{R}^n \text{ and } \|x - y\| < r\}$. An $x \in \mathcal{D} \subseteq \mathbb{R}^n$ is an *interior* point of $\mathcal{D}$ if there is an $r > 0$, such that $B(x, r) \subseteq \mathcal{D}$. The set of all interior points of $\mathcal{D}$ is called the *interior* of $\mathcal{D}$, and it is denoted by $\text{int}(\mathcal{D})$. A set $\mathcal{D} \subseteq \mathbb{R}^n$ is *open* if $\text{int}(\mathcal{D}) = \mathcal{D}$. A set $\mathcal{D} \subseteq \mathbb{R}^n$ is *closed* if its complement $\mathbb{R}^n \setminus \mathcal{D}$ is open. The *closure* of a set $\mathcal{D} \subseteq \mathbb{R}^n$ is defined as $\text{clos}(\mathcal{D}) = \mathbb{R}^n \setminus \text{int}(\mathbb{R}^n \setminus \mathcal{D})$. We sometimes denote closure of a set $\mathcal{D}$ by $\overline{\mathcal{D}}$.

3.1 Concave and Quasi-Concave Functions

A set $\mathcal{D} \subseteq \mathbb{R}^n$ is *convex* if for all $x, y \in \mathcal{D}$ and $\theta \in [0, 1]$, we have $\theta x + (1 - \theta)y \in \mathcal{D}$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is *concave* (on its domain $\text{dom}(f) \subseteq \mathbb{R}^n$), if $\text{dom}(f) \subseteq \mathbb{R}^n$ is a convex set, and for all $x, y \in \mathbb{R}^n$ and $\theta \in [0, 1]$, we have $f(\theta x + (1 - \theta)y) \geq \theta f(x) + (1 - \theta)f(y)$. A function f is *convex* if the function $-f$ is concave. A function is *affine* if it is both convex and concave. The *α-superlevel set* of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as $S^\alpha(f) = \{x \in \text{dom}(f) : f(x) \geq \alpha\}$, and the *$\alpha$-sublevel set* of f is defined as $S_\alpha(f) = \{x \in \text{dom}(f) \ : \ f(x) \leq \alpha\}$.

Proposition 2 ([12]). *If a function is concave then its superlevel sets are convex; and if it is convex then its sublevel sets are convex.*

The following properties of concave functions are of interest in this paper.

Lemma 3. *1. (Non-negative weighted sum) If $f_1, f_2, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$ are concave and $w_1, w_2, \ldots, w_k \geq 0$, then their w-weighted sum $\mathbb{R}^n \ni x \mapsto \sum_{i=1}^k w_i \cdot f_i$ is concave (on its domain $\bigcap_{i=1}^k dom(f_i)$).*

2. (Composition with an affine map) If $f : \mathbb{R}^n \to \mathbb{R}$ is concave, $A \in \mathbb{R}^{n \times n}$, and $b \in \mathbb{R}^n$, then the function $\mathbb{R}^n \ni x \mapsto f(Ax + b)$ is concave (on its domain $\{x : Ax + b \in dom(f)\}$).

3. (Pointwise minimum and infimum) If functions $f_1, f_2, \ldots f_k : \mathbb{R}^n \to \mathbb{R}$ are concave, then their pointwise minimum *$\mathbb{R}^n \ni x \mapsto \min_{i=1}^k f_i(x)$ is concave (on its domain $\bigcap_{i=1}^k dom(f_i)$).*
Let $f : \mathbb{R}^n \times Z \to \mathbb{R}$, where Z is an arbitrary (infinite) set. If for all $z \in Z$, the function $\mathbb{R}^n \ni x \mapsto f(x, z)$ is concave, then the function $\mathbb{R}^n \ni x \mapsto \inf_{z \in Z} f(x, z)$ is concave (on its domain $\bigcap_{z \in Z} dom(f(\cdot, z))$).

A function $f : \mathbb{R}^n \to \mathbb{R}$ is *quasi-concave* (on its domain $\text{dom}(f) \subseteq \mathbb{R}^n$), if $\text{dom}(f)$ is a convex set, and for all $x, y \in \text{dom}(f)$ and $\theta \in [0, 1]$, we have $f(\theta x + (1 - \theta)y) \geq \min\{f(x), f(y)\}$. A function f is *quasi-convex* if the function $-f$ is quasi-concave.

Proposition 4 ([12]). *A function is quasi-concave if and only if its superlevel sets are convex; and it is quasi-convex if and only if its sublevel sets are convex.*

The following properties of quasi-concave functions are of interest in this paper.

Lemma 5 ([18]). *For $h_1, h_2 : \mathbb{R}^n \to \mathbb{R}$, the function $\mathbb{R}^n \ni x \mapsto h_1(x)/h_2(x)$ is quasi-concave (on its domain $dom(h_1) \cap dom(h_2)$) if at least one of the following holds:*

1. *h_1 is nonnegative and convex, and h_2 is positive and convex;*
2. *h_1 is nonpositive and convex, and h_2 is negative and convex;*
3. *h_1 is nonnegative and concave, and h_2 is positive and concave;*
4. *h_1 is nonpositive and concave, and h_2 is negative and concave;*
5. *h_1 is affine and h_2 is non-zero and affine;*
6. *h_1 is concave, and h_2 is positive and affine;*
7. *h_1 is convex, and h_2 is negative and affine.*

3.2 Lipschitz-Continuous Functions

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz-continuous on its domain $\mathrm{dom}(f)$, if there exists a constant $K \geq 0$, called a Lipschitz constant of f, such that $\|f(x) - f(y)\| \leq K\|x-y\|$ for all $x, y \in \mathrm{dom}(f)$; we then also say that f is K-continuous. The following properties of Lipschitz-continuous functions are of interest in this paper.

Lemma 6. *1. If for every $i = 1, 2, \ldots, k$, the function $f_i : \mathbb{R}^n \to \mathbb{R}^m$ is K_i-continuous and $w_i \in \mathbb{R}$, then the function $\mathbb{R}^n \ni x \mapsto \sum_{i=1}^{n} w_i f_i(x)$ is K-continuous for $K = \sum_{i=1}^{k} |w_i| K_i$.*

2. If $f_1 : \mathbb{R}^n \to \mathbb{R}^m$ and $f_2 : \mathbb{R}^m \to \mathbb{R}^k$ are K_1-continuous and K_2-continuous, respectively, then their composition, $\mathbb{R}^n \ni x \mapsto f_2(f_1(x))$, is K-continuous for $K = K_1 K_2$.

3. Let $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$ be K_1-continuous and K_2-continuous, respectively; let f_1 and f_2 be bounded, i.e., there is a constant $M \geq 0$, such that for all $x \in dom(f_1) \cap dom(f_2)$, we have $|f_1(x)|, |f_2(x)| \leq M$; and let f_2 be bounded from below, i.e., there is a constant $N > 0$, such that for all $x \in dom(f_2)$, we have $f_2(x) \geq N$. Then the function $\mathbb{R}^n \ni x \mapsto f_1(x)/f_2(x)$ is K-continuous for $K = (NK_1 + MK_2)/N^2$.

3.3 Timed Automata

Clock Valuations, Regions, and Zones. Fix a constant $k \in \mathbb{N}$ for the rest of this paper. Let C be a finite set of *clocks*. A (k-bounded) *clock valuation* is a function $\nu : C \to [\![k]\!]_{\mathbb{R}}$; we write V for the set $[C \to [\![k]\!]_{\mathbb{R}}]$ of clock valuations. If $\nu \in V$ and $t \in \mathbb{R}_{\oplus}$ then we write $\nu + t$ for the clock valuation defined by $(\nu + t)(c) = \nu(c) + t$, for all $c \in C$. For a set $C' \subseteq C$ of clocks and a clock valuation $\nu : C \to \mathbb{R}_{\oplus}$, we define $\mathrm{Reset}(\nu, C')(c) = 0$ if $c \in C'$, and $\mathrm{Reset}(\nu, C')(c) = \nu(c)$ if $c \notin C'$.

Note 7. Clocks in timed automata are usually allowed to take arbitrary non-negative real values, while we restrict them to be bounded by some constant k, i.e., we consider only *bounded* timed automata models. We can make this restriction for technical convenience and without significant loss of generality.

The set of *clock constraints* over the set of clocks C is the set of conjunctions of *simple clock constraints*, which are constraints of the form $c \bowtie i$ or $c - c' \bowtie i$, where $c, c' \in C$, $i \in [\![k]\!]_{\mathbb{N}}$, and $\bowtie \in \{ <, >, =, \leq, \geq \}$. There are finitely many simple clock constraints. For every clock valuation $\nu \in V$, let $\mathrm{SCC}(\nu)$ be the set of simple clock constraints which hold in $\nu \in V$. A *clock region* is a maximal set $P \subseteq V$, such that for all $\nu, \nu' \in P$, $\mathrm{SCC}(\nu) = \mathrm{SCC}(\nu')$. In other words, every clock region is an equivalence class of the indistinguishability-by-clock-constraints relation, and vice versa. Note that ν and ν' are in the same clock region iff all clocks have the same integer parts in ν and ν', and if the partial orders of the clocks, determined by their fractional parts in ν and ν', are the same. For all $\nu \in V$, we write $[\nu]$ for the clock region of ν.

A *clock zone* is a convex set of clock valuations, which is a union of a set of clock regions. Note that a set of clock valuations is a zone iff it is definable by a clock constraint. For $W \subseteq V$, we write $\overline{W}$ for the smallest closed set in V which contains W. Observe that for every clock zone W, the set $\overline{W}$ is also a clock zone.

Let L be a finite set of *locations*. A *configuration* is a pair (ℓ, ν), where $\ell \in L$ is a location and $\nu \in V$ is a clock valuation; we write Q for the set of configurations. If $s = (\ell, \nu) \in Q$ and $c \in C$, then we write $s(c)$ for $\nu(c)$. A *region* is a pair (ℓ, P), where ℓ is a location and P is a clock region. If $s = (\ell, \nu)$ is a configuration then we write $[s]$ for the region $(\ell, [\nu])$. We write $\mathcal{R}$ for the set of regions. A set $Z \subseteq Q$ is a *zone* if for every $\ell \in L$, there is a clock zone W_ℓ (possibly empty), such that $Z = \{(\ell, \nu) : \ell \in L \text{ and } \nu \in W_\ell\}$. For a region $R = (\ell, P) \in \mathcal{R}$, we write $\overline{R}$ for the zone $\{(\ell, \nu) : \nu \in \overline{P}\}$.

Timed Automata. A *timed automaton* $T = (L, C, S, A, E, \delta, \xi, F)$ consists of a finite set of locations L, a finite set of clocks C, a set of *states* $S \subseteq Q$, a finite set of *actions* A, an *action enabledness function* $E : A \to 2^S$, a *transition function* $\delta : L \times A \to L$, a *clock reset function* $\xi : A \to 2^C$, and a set of *final states* $F \subseteq S$. We require that S, F, and $E(a)$ for all $a \in A$, are zones.

Clock zones, from which zones S, F, and $E(a)$, for all $a \in A$, are built, are typically specified by clock constraints. Therefore, when we consider a timed automaton as an input of an algorithm, its size should be understood as the sum of sizes of encodings of L, C, A, δ, and ξ, and the sizes of encodings of clock constraints defining zones S, F, and $E(a)$, for all $a \in A$. Our definition of a timed automaton may appear to differ from the usual ones [4,9]. The differences are, however, superficial and mostly syntactic.

For a configuration $s = (\ell, \nu) \in Q$ and $t \in \mathbb{R}_\oplus$, we define $s + t$ to be the configuration $s' = (\ell, \nu + t)$ if $\nu + t \in V$, and we then write $s \to_t s'$. For an action $a \in A$, we define $\mathrm{Succ}(s, a)$ to be the configuration $s' = (\ell', \nu')$, where $\ell' = \delta(\ell, a)$ and $\nu' = \mathrm{Reset}(\nu, \xi(a))$, and we then write $s \xrightharpoonup{a} s'$. We write $s \xrightarrow{a} s'$ if $s \xrightharpoonup{a} s'$; $s, s' \in S$; and $s \in E(a)$. For technical convenience, and without loss of generality, we will assume throughout that for every $s \in S$, there exists $a \in A$, such that $s \xrightarrow{a} s'$.

For $s, s' \in S$, we say that s' is in the future of s, or equivalently, that s is in the past of s', if there is $t \in \mathbb{R}_\oplus$, such that $s \to_t s'$; we then write $s \to_* s'$. For $R, R' \in \mathcal{R}$, we say that R' is in the future of R, or that R is in the past of R', if for all $s \in R$, there is $s' \in R'$, such that s' is in the future of s; we then write $R \to_* R'$. We say that R' is the *time successor* of R if $R \to_* R'$, $R \neq R'$, and for every $R'' \in \mathcal{R}$, we have that $R \to_* R'' \to_* R'$ implies $R'' = R$ or $R'' = R'$; we then write $R \to_{+1} R'$ or $R' \leftarrow_{+1} R$. Similarly, for $R, R' \in \mathcal{R}$, we write $R \xrightarrow{a} R'$ if there is $s \in R$, and there is $s' \in R'$, such that $s \xrightarrow{a} s'$.

We say that a region $R \in \mathcal{R}$ is *thin* if for every $s \in R$ and every $\varepsilon > 0$, we have that $[s] \neq [s + \varepsilon]$; other regions are called *thick*. We write $\mathcal{R}_{\mathrm{Thin}}$ and $\mathcal{R}_{\mathrm{Thick}}$ for the sets of thin and thick regions, respectively. Note that if $R \in \mathcal{R}_{\mathrm{Thick}}$ then for every $s \in R$, there is an $\varepsilon > 0$, such that $[s] = [s + \varepsilon]$. Observe also, that the time successor of a thin region is thick, and vice versa.

A *timed action* is a pair $\tau = (t, a) \in \mathbb{R}_\oplus \times A$. For $s \in Q$, we define $\mathrm{Succ}(s, \tau) = \mathrm{Succ}(s, (t, a))$ to be the configuration $s' = \mathrm{Succ}(s + t, a)$, i.e., such that $s \rightharpoonup_t s'' \xrightharpoonup{a} s'$, and we then write $s \xrightharpoonup{a}_t s'$. We write $s \xrightarrow{a}_t s'$ if $s \to_t s'' \xrightarrow{a} s'$, and we then say that $(s, (t, a), s')$ is a *transition* of the timed automaton. If $\tau = (t, a)$ then we write $s \xrightharpoonup{\tau} s'$ instead of $s \xrightharpoonup{a}_t s'$, and $s \xrightarrow{\tau} s'$ instead of $s \xrightarrow{a}_t s'$.

The Reachability Problem. A finite run is a sequence $\langle s_0, \tau_1, s_1, \tau_2, \ldots, \tau_n, s_n \rangle \in S \times ((\mathbb{R}_\oplus \times A) \times S)^*$, such that for all i, $1 \leq i \leq n$, we have that (s_{i-1}, τ_i, s_i) is a transition, i.e., that $s_{i-1} \xrightarrow{\tau_i} s_i$. For a finite run $r = \langle s_0, \tau_1, s_1, \tau_2, \ldots, \tau_n, s_n \rangle$, we define $\mathrm{Length}(r) = n$, and we define $\mathrm{Last}(r) = s_n$ to be the state in which the run ends. We write $\mathrm{Runs_{fin}}$ for the set of finite runs, and $\mathrm{Runs_{fin}}(s)$ for the set of finite runs starting from state $s \in S$. An infinite run of a timed automaton is a sequence $r = \langle s_0, \tau_1, s_1, \tau_2, \ldots \rangle$, such that for all $i \geq 1$, we have $s_{i-1} \xrightarrow{\tau_i} s_i$. For an infinite run r, we define $\mathrm{Length}(r) = \infty$. For a run $r = \langle s_0, \tau_1, s_1, \tau_2, \ldots \rangle$, we define $\mathrm{Stop}(r) = \inf\{i : s_i \in F\}$. We write Runs for the set of infinite runs, and $\mathrm{Runs}(s)$ for the set of infinite runs starting from state $s \in S$.

The *reachability problem* for timed automata is the following: given a timed automaton $\mathcal{T}$ and an initial state $s \in S$, decide whether there is a run $r \in \mathrm{Runs}(s)$, such that $\mathrm{Stop}(r) < \infty$. The following is a well known result.

Theorem 8 ([4]). *The reachability problem for timed automata is PSPACE-complete.*

Strategies. A *strategy* is a function $\sigma : \mathrm{Runs_{fin}} \to \mathbb{R}_{\oplus} \times A$, such that if $\mathrm{Last}(r) = s \in S$ and $\sigma(r) = \tau$ then $s \xrightarrow{\tau} s'$. We write Σ for the set of strategies. A run according to a strategy σ from a state $s \in S$ is the unique run $\mathrm{Run}(s, \sigma) = \langle s_0, \tau_1, s_1, \tau_2, \ldots \rangle$, such that $s_0 = s$, and for every $i \geq 1$, we have $\sigma(\mathrm{Run}_i(s, \sigma)) = \tau_{i+1}$, where $\mathrm{Run}_i(s, \sigma) = \langle s_0, \tau_1, s_1, \ldots, s_{i-1}, \tau_i, s_i \rangle$.

Since for every run $r \in \mathrm{Runs}(s)$, there is a strategy $\sigma \in \Sigma$, such that $\mathrm{Run}(s, \sigma) = r$, the reachability problem can be equivalently stated in terms of strategies: given a timed automaton $\mathcal{T}$ and an initial state $s \in S$, decide whether there is a strategy $\sigma \in \Sigma$, such that $\mathrm{Stop}(\mathrm{Run}(s, \sigma)) < \infty$.

We say that a strategy σ is *positional* if for all finite runs $r, r' \in \mathrm{Runs_{fin}}$, we have that $\mathrm{Last}(r) = \mathrm{Last}(r')$ implies $\sigma(r) = \sigma(r')$. A positional strategy can be then represented as a function $\sigma : S \to \mathbb{R}_{\oplus} \times A$, which uniquely determines the strategy $\sigma^{\infty} \in \Sigma$ as follows: $\sigma^{\infty}(r) = \sigma(\mathrm{Last}(r))$, for all finite runs $r \in \mathrm{Runs_{fin}}$. We write Π for the sets of positional strategies.

3.4 Priced Timed Automata

A *priced timed automaton* $(\mathcal{T}, \pi)$ consists of a timed automaton $\mathcal{T}$ and a price function $\pi : S \times \mathbb{R}_{\oplus} \times A \to \mathbb{R}$ that, for every state $s \in S$ and a timed move $(t, a) \in \mathbb{R}_{\oplus} \times A$, determines the price $\pi(s, t, a)$ of taking the timed move (t, a) from state s, i.e., of the transition $(s, (t, a), \mathrm{Succ}(s, (t, a)))$. In a *linearly-priced* timed automaton [7], the price function is represented as a function $p : L \cup A \to \mathbb{R}$, that gives a *price rate* $p(\ell)$ to every location $\ell \in L$, and a price $p(a)$ to every action a; the price of taking the timed move (a, t) from state $s = (\ell, \nu)$ is then defined by $\pi(s, (t, a)) = p(\ell) \cdot t + p(a)$.

In this paper we consider *concavely-priced* timed automata, a generalization of linearly-priced timed automata. Unlike for linearly-priced timed automata, we do not specify explicitly how the price function $\pi : S \times \mathbb{R}_{\oplus} \times A$ is represented; for conceptual simplicity it is convenient to think of it as a black box. We do, however, require that there is a constant $K > 0$, given as a part of the input, such that for all actions $a \in A$ and for all regions $R, R' \in \mathcal{R}$, the function $\pi^a_{R,R'} : (s, t) \mapsto \pi(s, t, a)$ is concave and K-continuous on $D_{R,R'} = \{(s, t) \in S \times \mathbb{R}_{\oplus} : s \in R \text{ and } (s + t) \in R'\}$.

Notice that every linearly-priced timed automaton is a concavely-priced timed automaton. In the rest of the paper we reserve the term priced timed automata to refer to concavely-priced timed automata.

We also consider *concave price-reward timed automata* $(\mathcal{T}, \pi, \varrho)$, where the *price* and *reward* functions $\pi, \varrho : S \times \mathbb{R}_{\oplus} \times A \to \mathbb{R}$ satisfy the following properties: there is a constant $K > 0$, given as a part of the input, such that for all actions $a \in A$ and for all regions $R, R' \in \mathcal{R}$, the functions $(s,t) \mapsto \pi(s,t,a)$ and $(s,t) \mapsto \varrho(s,t,a)$ are K-continuous, and concave and convex, respectively, on $\{(s,t) \in S \times \mathbb{R}_{\oplus} : s \in R \text{ and } (s+t) \in R'\}$. Moreover, for technical convenience we require that the timed automaton is *structurally non-Zeno* with respect to ϱ, i.e., for every run $r = \langle s_0, \tau_1, s_1, \tau_2, \ldots, \tau_n, s_n \rangle \in \text{Runs}_{\text{fin}}$, such that $s_0 = (\ell_0, \nu_0)$, $s_n = (\ell_n, \nu_n)$, and $\ell_0 = \ell_n$ (i.e., such that the run r forms a cycle in the finite graph of the locations and transitions of the timed automaton), we have that $\sum_{i=1}^{n} \varrho(s_{i-1}, \tau_i) \geq 1$.

4 Optimization Problems on Priced Timed Automata

The fundamental reachability problem for timed automata can be, in a natural way, generalized to a number of optimization problems on priced timed automata. Let Cost : Runs $\to \mathbb{R}$ be a cost function that for every run $r \in$ Runs determines its cost Cost(r). We then define the *minimum cost* function $\text{Cost}_* : S \to \mathbb{R}$, by

$$\text{Cost}_*(s) = \inf_{r \in \text{Runs}(s)} \text{Cost}(r) = \inf_{\sigma \in \Sigma} \text{Cost}(\text{Run}(s, \sigma)).$$

The *minimization problem* for that cost function Cost is: given a state $s \in S$ and a number $D \in \mathbb{Q}$, determine whether $\text{Cost}_*(s) \leq D$.

The following list of cost functions gives rise to a number of corresponding minimization problems. Let $r = \langle s_0, \tau_1, s_1, \tau_2, \ldots \rangle \in$ Runs, where $\tau_i = (t_i, a_i)$ for all $i \geq 1$. Moreover, for π and ϱ, the price and reward functions, respectively, of a priced (or price-reward) timed automaton, and for every $n \geq 1$, let: $T_n(r) = \sum_{i=1}^{n} t_i$, $\pi_n(r) = \sum_{i=1}^{n} \pi(s_{i-1}, \tau_i)$, and $\varrho_n(r) = \sum_{i=1}^{n} \varrho(s_{i-1}, \tau_i)$.

1. *Reachability*: $\text{Reach}(r) = \pi_N(r)$ if $N = \text{Stop}(r) < \infty$, and we define $\text{Reach}(r) = \infty$ otherwise.
2. *Discounted*: $\text{Discounted}(\lambda)(r) = (1-\lambda) \sum_{i=1}^{\infty} \lambda^{i-1} \pi(s_{i-1}, \tau_i)$, where $\lambda \in (0,1)$ is the *discount factor*.
3. *Average time*: $\text{AvgTime}(r) = \limsup_{n \to \infty} T_n/n$.
4. *Average price*: $\text{AvgPrice}(r) = \limsup_{n \to \infty} \pi_n(r)/n$.
5. *Price-per-time average*: $\text{TimeAvgPrice}(r) = \limsup_{n \to \infty} \pi_n(r)/T_n(r)$.
6. *Price-per-reward average*: $\text{PriceRewardAvg}(r) = \limsup_{n \to \infty} \pi_n(r)/\varrho_n(r)$.

The following is the main result of the paper.

Theorem 9. *The minimization problems for reachability, discounted, average time, average price, price-per-time average, and price-per-reward average cost functions, for concavely-priced and concave price-reward timed automata, as appropriate, are PSPACE-complete.*

The reachability problem for timed automata can be easily reduced, in logarithmic space, to the minimization problems discussed above so, by Theorem 8, they are all PSPACE-hard. In Sections 5 and 6 we prove that they are all in PSPACE, and hence we establish the main Theorem 9.

5 Region Graphs

We say that a run $r = \langle s_0, (t_1, a_1), s_1, (t_2, a_2), \ldots \rangle$ of a timed automaton $\mathcal{T}$ is of type $\Lambda(r) = \langle R_0, (R_1', a_1), R_1, (R_2', a_2), \ldots \rangle$, if for all $i \in \mathbb{N}$, we have $[s_i] = R_i$ and $[s_i + t_{i+1}] = R_{i+1}'$. We write Types for the set of run types, and we write Types(R) for the set of run types starting from region $R \in \mathcal{R}$.

For $\Lambda = \langle R_0, (R_1', a_1), R_1, (R_2', a_2), \ldots \rangle \in$ Types, $s \in R_0$, and $\overline{t} = \langle t_1, t_2, \ldots \rangle \in \mathbb{R}_{\oplus}^{\omega}$, we define $\mathrm{PreRun}_s^{\Lambda}(\overline{t}) = \langle (s_0, R_0), (R_1', t_1, a_1), (s_1, R_1), (R_2', t_2, a_2), \ldots \rangle$, where $s_0 = s$, and for $i \in \mathbb{N}$, we have $(s_i + t_{i+1}) \xrightarrow{a_{i+1}} s_{i+1}$. For $s, s' \in S$, $R, R'', R' \in \mathcal{R}$, $t \in \mathbb{R}_{\oplus}$, and $a \in A$, we also say that $\big((s, R), (R'', t, a), (s', R')\big)$ is a pre-transition if $(s + t) \xrightarrow{a} s'$.

5.1 Region Graph $\widetilde{\mathcal{T}}$

Let $\mathcal{T}$ be a timed automaton. We define the *region graph* $\widetilde{\mathcal{T}}$ to be the finite edge-labelled graph $(\mathcal{R}, \widetilde{\mathcal{M}})$, where the set $\mathcal{R}$ of $\mathcal{T}$ is the set of vertices, and the labelled edge relation $\widetilde{\mathcal{M}} \subseteq \mathcal{R} \times \mathcal{R} \times A \times \mathcal{R}$ is defined by $\widetilde{\mathcal{M}} = \{(R, R'', a, R') \,:\, R \to_* R'' \xrightarrow{a} R'\}$.

Let $\widetilde{S} = \{(s, R) \in S \times \mathcal{R} \,:\, s \in R\}$ be the set of states of $\widetilde{\mathcal{T}}$. For $(s, R), (s', R') \in \widetilde{S}$ and $(R'', t, a) \in \mathcal{R} \times \mathbb{R}_{\oplus} \times A$, we say that $\big((s, R), (R'', t, a), (s', R')\big)$ is a transition in $\widetilde{\mathcal{T}}$ if: it is a pre-transition, $(s + t) \in R''$, and $(R, R'', a, R') \in \widetilde{\mathcal{M}}$. We then also say that there is an (R'', t, a)-transition from state (s, R) in $\widetilde{\mathcal{T}}$.

A run of $\widetilde{\mathcal{T}}$ is a sequence $\langle (s_0, R_0), (R_1', t_1, a_1), (s_1, R_1), (R_2', t_2, a_2), \ldots \rangle$, such that for all $i \in \mathbb{N}$, we have that $\big((s_i, R_i), (R_{i+1}'', t_{i+1}, a_{i+1}), (s_{i+1}', R_{i+1}')\big)$ is a transition in $\widetilde{\mathcal{T}}$. We write $\mathrm{Runs}^{\widetilde{\mathcal{T}}}$ for the set of runs of $\widetilde{\mathcal{T}}$, and for $(s, R) \in \widetilde{S}$, we write $\mathrm{Runs}^{\widetilde{\mathcal{T}}}(s, R)$ for the set of runs of $\widetilde{\mathcal{T}}$ whose initial state is (s, R).

The timed automaton $\mathcal{T}$ and the region graph $\widetilde{\mathcal{T}}$ are equivalent in the following sense.

Proposition 10. *For every $s \in S$ and $(t, a) \in \mathbb{R}_{\oplus} \times A$, there is a (t, a)-transition from s in $\mathcal{T}$ if and only if there is a $([s + t], t, a)$-transition from $(s, [s])$ in $\widetilde{\mathcal{T}}$.*

Let $(\mathcal{T}, \pi)$ be a concavely-priced timed automaton. We define the price function $\widetilde{\pi} : \widetilde{S} \times (\mathcal{R} \times \mathbb{R}_{\oplus} \times A) \to \mathbb{R}$ in the following way. For $(s, R) \in \widetilde{S}$ and $(R'', t, a) \in \mathcal{R} \times \mathbb{R}_{\oplus} \times A$, such that there is a (R'', t, a)-transition from (s, R) in $\widetilde{\mathcal{T}}$, we define $\widetilde{\pi}\big((s, R), (R'', t, a)\big) = \pi(s, t, a)$. For a concave price-reward automaton $(\mathcal{T}, \pi, \varrho)$, we define functions $\widetilde{\pi}$ and $\widetilde{\varrho}$ in an analogous way.

5.2 Boundary Region Graph $\widehat{\mathcal{T}}$

Define the finite set of *boundary timed actions* $\mathcal{A} = [\![k]\!]_{\mathbb{N}} \times C \times A$. For $s \in Q$ and $\alpha = (b, c, a) \in \mathcal{A}$, we define $t(s, \alpha) = b - s(c)$; and we define $\mathrm{Succ}(s, \alpha)$ to be the state

$s' = \mathrm{Succ}(s, \tau(\alpha))$, where $\tau(\alpha) = (t(s,\alpha), a)$; we then write $s \xrightarrow{\alpha} s'$. We also write $s \xrightarrow{\alpha} s'$ if $s \xrightarrow{\tau(\alpha)} s'$. Note that if $\alpha \in \mathcal{A}$ and $s \xrightarrow{\alpha} s'$ then $[s'] \in \mathcal{R}_{\mathrm{Thin}}$. Observe that for every thin region $R' \in \mathcal{R}_{\mathrm{Thin}}$, there is a number $b \in [\![k]\!]_{\mathbb{N}}$ and a clock $c \in C$, such that for every $R \in \mathcal{R}$ in the past of R', we have that $s \in R$ implies $(s + (b - s(c)) \in R'$; we then write $R \to_{b,c} R'$. For $\alpha = (b, c, a) \in \mathcal{A}$ and $R, R' \in \mathcal{R}$, we write $R \xrightarrow{\alpha} R'$ or $R \xrightarrow{a}_{b,c} R'$, if $R \to_{b,c} R'' \xrightarrow{a} R'$, for some $R'' \in \mathcal{R}_{\mathrm{Thin}}$.

Let $\mathcal{T}$ be a timed automaton. We define the *boundary region graph* $\widehat{\mathcal{T}}$ to be the finite edge-labelled graph $(\mathcal{R}, \widehat{\mathcal{M}})$, where the set $\mathcal{R}$ of $\mathcal{T}$ is the set of vertices, and the labelled edge relation $\widehat{\mathcal{M}} \subseteq \mathcal{R} \times \mathcal{R} \times \mathcal{A} \times \mathcal{R}$ is defined in the following way. For $\alpha = (b, c, a) \in \mathcal{A}$ and $R, R'', R' \in \mathcal{R}$, we have $(R, R'', \alpha, R') \in \widehat{\mathcal{M}}$ if one of the following conditions holds:

- $R \to_{b,c} R'' \xrightarrow{a} R'$; or
- there is an $R''' \in \mathcal{R}$, such that $R \to_{b,c} R''' \to_{+1} R'' \xrightarrow{a} R'$; or
- there is an $R''' \in \mathcal{R}$, such that $R \to_{b,c} R''' \leftarrow_{+1} R'' \xrightarrow{a} R'$.

Let $\widehat{S} = \{(s, R) \in S \times \mathcal{R} : s \in \overline{R}\}$ be the set of states of $\widehat{\mathcal{T}}$. For $(s, R), (s', R') \in \widehat{S}$ and $(R'', t, a) \in \mathcal{R} \times \mathbb{R}_{\oplus} \times A$, we say that $\big((s, R), (R'', t, a), (s', R')\big)$ is a transition in $\widehat{\mathcal{T}}$ if: it is a pre-transition, and there is an $\alpha = (b, c, a)$, such that $t = b - s(c)$, $(s + t) \in \overline{R''}$, and $(R, R'', \alpha, R') \in \widehat{\mathcal{M}}$.

A run of $\widehat{\mathcal{T}}$ is a sequence $\langle (s_0, R_0), (R'_1, t_1, a_1), (s_1, R_1), (R'_2, t_2, a_2), \ldots \rangle$, such that for all $i \in \mathbb{N}$, we have that $\big((s_i, R_i), (R''_{i+1}, t_{i+1}, a_{i+1}), (s'_{i+1}, R'_{i+1})\big)$ is a transition in $\widehat{\mathcal{T}}$. We write $\mathrm{Runs}^{\widehat{\mathcal{T}}}$ for the set of runs of $\widehat{\mathcal{T}}$, and for $(s, R) \in \widehat{S}$, we write $\mathrm{Runs}^{\widehat{\mathcal{T}}}(s, R)$ for the set of runs of $\widehat{\mathcal{T}}$ whose initial state is (s, R).

Let $(\mathcal{T}, \pi)$ be a concavely-priced timed automaton. We define the price function $\widehat{\pi} : \widehat{S} \times (\mathcal{R} \times \mathbb{R}_{\oplus} \times A)$ in the following way. Recall that for $a \in A$ and $R, R'' \in \mathcal{R}$, the function $\pi^a_{R,R''} : (s, t) \mapsto \pi(s, a, t)$ defined on the set $D_{R,R''} = \{(s, t) : s \in R \text{ and } (s+t) \in R''\}$ is continuous. We write $\overline{\pi^a_{R,R''}}$ for the unique continuous extension of $\pi^a_{R,R''}$ to the closure $\overline{D_{R,R''}}$ of the set $D_{R,R''}$. For $(s, R) \in \widehat{S}$ and $(R'', t, a) \in \mathcal{R} \times \mathbb{R}_{\oplus} \times A$, such that there is an (R'', t, a)-transition from (s, R) in $\widehat{\mathcal{T}}$, we define $\widehat{\pi}\big((s, R), (R'', t, a)\big) = \overline{\pi^a_{R,R''}}(s, t)$. For a concave price-reward automaton $(\mathcal{T}, \pi, \varrho)$, we define functions $\widehat{\pi}$ and $\widehat{\varrho}$ in an analogous way.

Proposition 11. *If $r \in Runs^{\widetilde{\mathcal{T}}} \cap Runs^{\widehat{\mathcal{T}}}$ then $\widetilde{\pi}(r) = \widehat{\pi}(r)$.*

Thanks to the above proposition we can, and sometimes will, abuse notation by writing $\pi(r)$ instead of $\widetilde{\pi}(r)$ or $\widehat{\pi}(r)$ for $r \in \mathrm{Runs}^{\widetilde{\mathcal{T}}}$ or $r \in \mathrm{Runs}^{\widehat{\mathcal{T}}}$, respectively.

5.3 Optimization Problems on the Region Graphs $\widetilde{\mathcal{T}}$ and $\widehat{\mathcal{T}}$

For a cost function $\mathrm{Cost} : \mathrm{PreRuns} \to \mathbb{R}$, we define the *minimum cost* functions $\mathrm{Cost}^{\widetilde{\mathcal{T}}}_* : \widetilde{S} \to \mathbb{R}$ and $\mathrm{Cost}^{\widehat{\mathcal{T}}}_* : \widehat{S} \to \mathbb{R}$, by:

$$\mathrm{Cost}^{\widetilde{\mathcal{T}}}_*(s, R) = \inf_{r \in \mathrm{Runs}^{\widetilde{\mathcal{T}}}(s,R)} \mathrm{Cost}(r), \quad \text{and} \quad \mathrm{Cost}^{\widehat{\mathcal{T}}}_*(s, R) = \inf_{r \in \mathrm{Runs}^{\widehat{\mathcal{T}}}(s,R)} \mathrm{Cost}(r).$$

The corresponding *minimization problems* are: given a state $s \in S$ and a number $D \in \mathbb{Q}$, determine whether $\mathrm{Cost}_*^{\widetilde{\mathcal{T}}}(s, [s]) \le D$ and $\mathrm{Cost}_*^{\widehat{\mathcal{T}}}(s, [s]) \le D$, respectively.

The following list of cost functions gives rise to a number of minimization problems. Let $r = \langle (s_0, R_0), (R'_1, t_1, a_1), (s_1, R_1), (R'_2, t_2, a_2), \ldots \rangle$ be a run of $\widetilde{\mathcal{T}}$ or $\widehat{\mathcal{T}}$. For all $n \in \mathbb{N}$, define $T_n(r) = \sum_{i=1}^{n} t_i$; $\pi_n(r) = \sum_{i=1}^{n} \pi((s_{i-1}, R_{i-1}), (R'_i, t_i, a_i))$; and $\varrho_n(r) = \sum_{i=1}^{n} \varrho((s_{i-1}, R_{i-1}), (R'_i, t_i, a_i))$. With those notations, we define the reachability, discounted, average time, average price, price-per-time average, and price-per-reward average cost functions, on the sets of runs of $\widetilde{\mathcal{T}}$ and $\widehat{\mathcal{T}}$, in exactly the same way as for runs of the timed automaton $\mathcal{T}$; see Section 4.

The following is an easy corollary of Proposition 10.

Proposition 12. *If Cost is any of the reachability, discounted, average time, average price, price-per-time average, or price-per-reward average cost functions, then for all $s \in S$, we have $\mathit{Cost}_*^{\mathcal{T}}(s) = \mathit{Cost}_*^{\widetilde{\mathcal{T}}}(s, [s])$.*

The following theorem is one of the main technical results of the paper.

Theorem 13. *If Cost is any of the reachability, discounted, average time, average price, price-per-time average, or price-per-reward average cost functions, then for all $s \in S$, we have $\mathit{Cost}_*^{\widetilde{\mathcal{T}}}(s, [s]) = \mathit{Cost}_*^{\widehat{\mathcal{T}}}(s, [s])$.*

Observe that for every state $s \in S$, the number of states reachable from s in the boundary region graph $\widehat{\mathcal{T}}$ is at most proportional to the size of $\widehat{\mathcal{T}}$, and hence finite. By Theorem 1, it follows that optimal positional strategies exist in $\widehat{\mathcal{T}}$ for all above-mentioned cost functions. Therefore, and since a run from a state according to a positional strategy in $\widehat{\mathcal{T}}$ can be guessed, and its cost computed, in PSPACE (with respect to the size of the input, i.e., a timed automaton $\mathcal{T}$), it suffices to prove Theorem 13 in order to obtain the main Theorem 9. We dedicate Section 6 to the proof of Theorem 13.

6 Correctness of the Bounded Region Graph Abstraction

6.1 Approximations of Cost Functions

For $n \in \mathbb{N}$, we write $\mathrm{Runs}^{\widetilde{\mathcal{T}}}(n)$ and $\mathrm{Runs}^{\widehat{\mathcal{T}}}(n)$ for the sets of runs of $\widetilde{\mathcal{T}}$ and $\widehat{\mathcal{T}}$, respectively, of length n. Also, for a run $r \in \mathrm{Runs}^{\widetilde{\mathcal{T}}}$ or $r \in \mathrm{Runs}^{\widehat{\mathcal{T}}}$, and $n \in \mathbb{N}$, we write $\mathrm{Prefix}(r, n)$ for the finite run consisting of the first n transitions of r. For $r \in \mathrm{Runs}^{\widetilde{\mathcal{T}}}$, we sometimes abuse notation—for the sake of brevity—by writing $\mathrm{Cost}_n(r)$ instead of $\mathrm{Cost}_n(\mathrm{Prefix}(r, n))$; the same applies to runs in $\mathrm{Runs}^{\widehat{\mathcal{T}}}$.

We say that a sequence of functions $\langle \mathrm{Cost}_n : \mathrm{PreRuns}(n) \to \mathbb{R} \rangle_{n \in \mathbb{N}}$ approximates a cost function $\mathrm{Cost} : \mathrm{Runs}^{\widetilde{\mathcal{T}}} \to \mathbb{R}$ or $\mathrm{Cost} : \mathrm{Runs}^{\widehat{\mathcal{T}}} \to \mathbb{R}$, respectively, if for all $r \in \mathrm{Runs}^{\widetilde{\mathcal{T}}}$, or for all $r \in \mathrm{Runs}^{\widehat{\mathcal{T}}}$, respectively, we have that $\mathrm{Cost}(r) = \limsup_{n \to \infty} \mathrm{Cost}_n(r)$.

6.2 Cost Functions and Finite Run Types

Let $\Lambda = \langle R_0, (R'_1, a_1), R_1, (R'_2, a_2), R_2, \ldots \rangle$ be a run type. For a state $s \in R_0$ and $(t_1, t_2, \ldots, t_n) \in \mathbb{R}^n_\oplus$, we define $\text{PreRun}^\Lambda_{n,s}(t_1, t_2, \ldots, t_n) = \text{Prefix}(\text{PreRun}^\Lambda_s(\overline{t}), n)$, where the first n elements of $\overline{t} \in \mathbb{R}^\omega_\oplus$ are $t_1, t_2, \ldots, t_n$. We define $\Delta^\Lambda_{n,s} \subseteq \mathbb{R}^n_\oplus$ to consist of the tuples $(t_1, t_2, \ldots, t_n) \in \mathbb{R}^n_\oplus$, such that $\text{PreRun}^\Lambda_{n,s}(t_1, t_2, \ldots, t_n) \in \text{Runs}^{\widetilde{\mathcal{T}}}(n)$.

Proposition 14. *For every state $s \in S$, a run type $\Lambda \in Types([s])$, and $n \in \mathbb{N}$, the set $\Delta^\Lambda_{n,s}$ is a polytope.*

Proposition 15. *Let $R \in \mathcal{R}$, $\Lambda \in Types(R)$, $s \in R$, and $n \in \mathbb{N}$. There is a 1-to-1 correspondence between runs—starting from s, of type Λ, and of length n—in $\widehat{\mathcal{T}}$, and vertices of $\overline{\Delta^\Lambda_{n,s}}$.*

More precisely, $r = \langle (s_0, R_0), (R'_1, t_1, a_1), (s_1, R_1), \ldots, (R'_n, t_n, a_n), (s_n, R_n) \rangle$ is a run (of type Λ) in $\widehat{\mathcal{T}}$ if and only if there is a vertex $(t_1, t_2, \ldots, t_n)$ of $\overline{\Delta^\Lambda_{n,s}}$, such that $r = PreRun^\Lambda_{n,s}(t_1, t_2, \ldots, t_n)$.

The following is a well-known result [8].

Proposition 16. *Let $f : \Delta \to \mathbb{R}$ be a continuous quasi-concave function, where $\Delta \subseteq \mathbb{R}^n$ is a polytope. Let $\overline{f}$ be the unique continuous extension of f to the closure $\overline{\Delta}$ of Δ.*

- *There exists a vertex v of $\overline{\Delta}$, such that $\overline{f}(v) = \inf_{x \in \Delta} f(x)$.*
- *For every $\varepsilon > 0$, there exists $x \in \Delta$, such that $f(x) \leq \overline{f}(v) + \varepsilon$.*

Let a sequence $\langle \text{Cost}_n \rangle_{n \in \mathbb{N}}$ approximate a cost function Cost. We define the function $\text{Cost}^\Lambda_{n,s} : \Delta^\Lambda_{n,s} \to \mathbb{R}$ by $\text{Cost}^\Lambda_{n,s}(t_1, t_2, \ldots, t_n) = \text{Cost}_n(\text{PreRun}^\Lambda_{n,s}(t_1, t_2, \ldots, t_n))$.

The following can be derived from Propositions 15 and 16.

Corollary 17. *Let $Cost^\Lambda_{n,s}$ be quasi-concave on $\Delta^\Lambda_{n,s}$.*

1. *For every run $\widetilde{r} \in Runs^{\widetilde{\mathcal{T}}}(s)$ of type Λ, and for every $n \in \mathbb{N}$, there is a run $\widehat{r} \in Runs^{\widehat{\mathcal{T}}}(s)$ of type Λ, such that $Cost_n(\widehat{r}) \leq Cost_n(\widetilde{r})$.*
2. *For every run $\widehat{r} \in Runs^{\widehat{\mathcal{T}}}(s)$, and for every $\varepsilon > 0$, there is a run $\widetilde{r} \in Runs^{\widetilde{\mathcal{T}}}(s)$ of type Λ, such that $Cost_n(\widetilde{r}) \leq Cost_n(\widehat{r}) + \varepsilon$.*

Consider pre-runs $r = \langle (s_0, R_0), (R'_1, t_1, a_1), (s_1, R_1), (R'_2, t_2, a_2), \ldots \rangle$ and $r' = \langle (s'_0, R_0), (R'_1, t'_1, a_1), (s'_1, R_1), (R'_2, t'_2, a_2), \ldots \rangle$ of the same type. We define $r - r' = (s_0 - s'_0, t_1 - t'_1, s_1 - s'_1, t_2 - t'_2, \ldots)$, where for all $i \in \mathbb{N}$, the expression $(s_i - s'_i)$ stands for the finite sequence $\langle s_i(c) - s'_i(c) \rangle_{c \in C}$. For a sequence $\overline{x} = \langle x_i \rangle_{i \in \mathbb{N}} \in \mathbb{R}^\omega$ of reals, we define $\|\overline{x}\| = \sup_{i \in \mathbb{N}} |x_i|$.

Proposition 18. *For every run $\widehat{r} \in Runs^{\widehat{\mathcal{T}}}(s)$, and for every $\varepsilon > 0$, and there is a run $\widetilde{r} \in Runs^{\widetilde{\mathcal{T}}}(s)$, such that $\|\widehat{r} - \widetilde{r}\| \leq \varepsilon$.*

Proof. Let $\widehat{r} = \langle (s_0, R_0), (R'_1, t_1, a_1), (s_1, R_1), (R'_2, t_2, a_2), \ldots \rangle \in \text{Runs}^{\widehat{\mathcal{T}}}$. Note that since $\widehat{r} \in \text{Runs}^{\widehat{\mathcal{T}}}$, for every $i \in \mathbb{N}$, there are $b_i \in [\![k]\!]_\mathbb{N}$ and $c_i \in C$, such that $t_i = b_i - s_{i-1}(c_i)$. Let $\widetilde{r} = \langle (s'_0 = s_0, R_0), (R'_1, t'_1, a_1), (s'_1, R_1), (R'_2, t'_2, a_2), \ldots \rangle \in \text{Runs}^{\widetilde{\mathcal{T}}}$ (of the same type as $\widehat{r}$) be such that for all $i \in \mathbb{N}$, we choose $t'_i \in \mathbb{R}_\oplus$ so that $|t'_i - (b_i - s'_{i-1}(c_i))| < \varepsilon - \|s_{i-1} - s'_{i-1}\|$. It then follows that for all $i \in \mathbb{N}$, we have $\|s_i - s'_i\| < \varepsilon$, and hence $\|\widehat{r} - \widetilde{r}\| \leq \varepsilon$. $\square$

6.3 Concave-Regular Cost Functions

A cost function Cost : PreRuns $\to \mathbb{R}$ is *concave-regular* if it satisfies the following properties.

1. *(Quasi-concavity).* For every region $R \in \mathcal{R}$ and for every run type $\Lambda \in$ Types(R), there is $N \in \mathbb{N}$, such that for every state $s \in R$ and for every $n \geq N$, the function $\text{Cost}^{\Lambda}_{n,s}$ is quasi-concave on $\Delta^{\Lambda}_{n,s}$.
2. *(Regular Lipschitz-continuity).* There is a constant $K \geq 0$, such that for every region $R \in \mathcal{R}$ and for every positional run type $\Lambda \in$ Types(R), there is $N \in \mathbb{N}$, such that for every state $s \in R$ and for every $n \geq N$, the function $\text{Cost}^{\Lambda}_{n,s}$ is K-continuous on $\Delta^{\Lambda}_{n,s}$.
3. *(Uniform convergence).* There is $\chi : \mathbb{N} \to \mathbb{R}$, such that $\lim_{n\to\infty} \chi(n) = 0$, and for every state $s \in S$, run $\widehat{r} \in \text{Runs}^{\widehat{\mathcal{T}}}(s, [s])$, and $n \in \mathbb{N}$, we have $\text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s]) \leq \text{Cost}_n(\widehat{r}) + \chi(n)$.

Theorem 19. *If Cost : PreRuns $\to \mathbb{R}$ is concave-regular then for all states $s \in S$, we have $\text{Cost}^{\widetilde{\mathcal{T}}}_{*}(s, [s]) = \text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s])$.*

Proof. First we prove that for all $s \in S$, we have $\text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s]) \leq \text{Cost}^{\widetilde{\mathcal{T}}}_{*}(s, [s])$. It suffices to show that for every run $\widetilde{r} \in \text{Runs}^{\widetilde{\mathcal{T}}}(s, [s])$, we have $\text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s]) \leq \text{Cost}(\widetilde{r})$.

Let $\widetilde{r} \in \text{Runs}^{\widetilde{\mathcal{T}}}(s, [s])$ be a run in $\widetilde{\mathcal{T}}$ of type Λ. By the quasi-concavity property of Cost, there is $N \in \mathbb{N}$, such that for all $n \geq N$, the function $\text{Cost}^{\Lambda}_{n,s}$ is quasi-concave on $\Delta^{\Lambda}_{n,s}$. Hence—by the first part of Corollary 17—for every $n \geq N$, there is a run $\widehat{r_n} \in \text{Runs}^{\widehat{\mathcal{T}}}(s, [s])$ of type Λ, such that $\text{Cost}_n(\widehat{r_n}) \leq \text{Cost}_n(\widetilde{r})$.

By the uniform convergence property of Cost, there is a function $\chi : \mathbb{N} \to \mathbb{R}$, such that $\lim_{n\to\infty} \chi(n) = 0$ and for all $n \in \mathbb{N}$, we have $\text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s]) \leq \text{Cost}_n(\widehat{r_n}) + \chi(n)$. Hence—combining the last two inequalities—we get $\text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s]) \leq \text{Cost}_n(\widetilde{r}) + \chi(n)$, for $n \geq N$. Taking the limit supremum of both sides of the last inequality yields:

$$\text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s]) \leq \limsup_{n\to\infty} \big(\text{Cost}_n(\widetilde{r}) + \chi(n)\big) = \text{Cost}(\widetilde{r}).$$

Next we prove that for all $s \in S$, we have $\text{Cost}^{\widetilde{\mathcal{T}}}_{*}(s, [s]) \leq \text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s])$. It suffices to argue that for every $s \in S$ and $\varepsilon > 0$, there is a run $\widetilde{r} \in \text{Runs}^{\widetilde{\mathcal{T}}}(s, [s])$, such that $|\text{Cost}(\widetilde{r}) - \text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s])| \leq \varepsilon$.

Let $\varepsilon > 0$ and let $\widehat{r} \in \text{Runs}^{\widehat{\mathcal{T}}}(s, [s])$, so that $\text{Cost}(\widehat{r}) \leq \text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s]) + \varepsilon/2$, and hence $|\text{Cost}(\widehat{r}) - \text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s])| \leq \varepsilon/2$. Let $\widetilde{r} \in \text{Runs}^{\widetilde{\mathcal{T}}}$ be such that $\|\widetilde{r} - \widehat{r}\| \leq \varepsilon'$, for some $\varepsilon' > 0$ to be chosen later; existence of such $\widetilde{r} \in \text{Runs}^{\widetilde{\mathcal{T}}}(s, [s])$ follows from Proposition 18.

By the regular Lipschitz-continuity of Cost, there is $K \geq 0$ and $N \in \mathbb{N}$, such that for all $n \geq N$, we have: $|\text{Cost}_n(\widetilde{r}) - \text{Cost}_n(\widehat{r})| \leq K\|\widetilde{r} - \widehat{r}\| \leq K\varepsilon'$. Hence—by choosing $\varepsilon' > 0$ so that $\varepsilon' \leq \varepsilon/(2K)$—we obtain that:

$$|\text{Cost}_n(\widetilde{r}) - \text{Cost}_n(\widehat{r})| \leq \varepsilon/2,$$

for $n \geq N$. Recall, however, that we have chosen $\widehat{r} \in \text{Runs}^{\widehat{\mathcal{T}}}(s, [s])$ so that:

$$|\text{Cost}(\widehat{r}) - \text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s])| \leq \varepsilon/2.$$

From the last two inequalities it follows that $|\text{Cost}(\widetilde{r}) - \text{Cost}^{\widehat{\mathcal{T}}}_{*}(s, [s])| \leq \varepsilon$. □

Theorem 20. *Reachability, discounted, average time, average price, price-per-time average, and price-per-reward average cost functions are concave-regular for concavely-priced (or concave price-reward, as appropriate) timed automata.*

Note that the key Theorem 13 follows immediately from Theorems 19 and 20.

References

1. Abdeddaïm, Y., Maler, O.: Job-shop scheduling using timed automata. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 478–492. Springer, Heidelberg (2001)
2. Alur, R., Courcoubetis, C., Dill, D.: Model-checking in dense real-time. Information and Computation 104(1), 2–34 (1993)
3. Alur, R., Courcoubetis, C., Henzinger, T.A.: Computing accumulated delays in real-time systems. Formal Methods in System Design 11(2), 137–155 (1997)
4. Alur, R., Dill, D.: A theory of timed automata. In: Theoretical Computer Science, vol. 126, pp. 183–235 (1994)
5. Alur, R., La Torre, S., Pappas, G.J.: Optimal paths in weighted timed automata. Theoretical Computer Science 318(3), 297–322 (2004)
6. Asarin, E., Maler, O.: As soon as possible: Time optimal control for timed automata. In: Vaandrager, F.W., van Schuppen, J.H. (eds.) HSCC 1999. LNCS, vol. 1569, pp. 19–30. Springer, Heidelberg (1999)
7. Behrmann, G., Fehnker, A., Hune, T., Larsen, K.G., Pettersson, P., Romijn, J., Vaandrager, F.W.: Minimum-cost reachability for priced timed automata. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 147–161. Springer, Heidelberg (2001)
8. Bertsekas, D.P., Nedić, A., Ozdaglar, A.E.: Convex Analysis and Optimization. Athena Scientific (2003)
9. Bouyer, P.: Weighted timed automata: Model-checking and games. Electr. Notes Theor. Comput. Sci. 158, 3–17 (2006)
10. Bouyer, P., Brihaye, T., Bruyère, V., Raskin, J.: On the optimal reachability problem on weighted timed automata. Formal Methods in System Design 31(2), 135–175 (2007)
11. Bouyer, P., Brinksma, E., Larsen, K.G.: Optimal infinite scheduling for multi-priced timed automata. Formal Methods in System Design 32(1), 3–23 (2008)
12. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
13. Brihaye, T., Henzinger, T.A., Prabhu, V.S., Raskin, J.: Minimum-time reachability in timed games. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 825–837. Springer, Heidelberg (2007)
14. Courcoubetis, C., Yannakakis, M.: Minimum and maximum delay problems in real-time systems. In: FMSD 1992, vol. 1, pp. 385–415. Kluwer, Dordrecht (1992)
15. Jurdziński, M., Trivedi, A.: Reachability-time games on timed automata. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 838–849. Springer, Heidelberg (2007)

16. Kesten, Y., Pnueli, A., Sifakis, J., Yovine, S.: Decidable integration graphs. Information and Computation 150(2), 209–243 (1999)
17. Larsen, K.G., Behrmann, G., Brinksma, E., Fehnker, A., Hune, T., Pettersson, P., Romijn, J.: As cheap as possible: Efficient cost-optimal reachability for priced timed automata. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 493–505. Springer, Heidelberg (2001)
18. Mangasarian, O.L.: Nonlinear Programming. McGraw-Hill Series in Systems Science. McGraw-Hill, New York (1969)
19. Niebert, P., Tripakis, S., Yovine, S.: Minimum-time reachability for timed automata. In: MED 2000. IEEE Comp. Soc. Press, Los Alamitos (2000)