On the optimal reachability problem of weighted timed automata⋆

Patricia Bouyer[1], Thomas Brihaye[2], Véronique Bruyère[2], and Jean-François Raskin[3]

[1] LSV - CNRS & ENS de Cachan
Avenue du Président Wilson 61, F-94230 Cachan, France

[2] Faculté des Sciences, Université de Mons-Hainaut,
Avenue du Champ de Mars 6, B-7000 Mons, Belgium

[3] Département d'Informatique, Université Libre de Bruxelles,
Boulevard du Triomphe CP 212, B-1050-Bruxelles, Belgium

Abstract. We study the cost-optimal reachability problem for weighted timed automata such that positive and negative costs are allowed on edges and locations. By optimality, we mean an infimum cost as well as a supremum cost. We show that this problem is PSPACE-COMPLETE. Our proof uses techniques of linear programming, and thus exploits an important property of optimal runs : their time-transitions use a time τ which is arbitrarily close to an integer. We then propose an extension of the region graph, the weighted discrete graph, whose structure gives light on the way to solve the cost-optimal reachability problem. We also give an application of the cost-optimal reachability problem in the context of timed games.

1 Introduction

Timed automata are a well-established formalism for the modeling and analysis of timed systems. Timed automata augment finite state automata with clocks and clock constraints [AD94]. The reachability problem for a timed automaton $\mathcal{A}$ asks, given a location l of $\mathcal{A}$, if there exists a run of $\mathcal{A}$ that visits the location l. This basic problem has been shown PSPACE-COMPLETE in the seminal paper of Alur and Dill [AD94]. The verification of more complex properties like properties expressed in the timed extension of the CTL logic, known as TCTL, is also a PSPACE-COMPLETE problem [ACD93]. On the other hand, some problems have been shown undecidable on the model of timed automata. For example, the universality problem that asks if a given timed automaton accepts the language of all timed words, has been shown undecidable in [AD94]. As a direct consequence, the language inclusion problem between two timed automata is also undecidable. Not only a large number of important and interesting theoretical results have been obtained on timed automata, but efficient verification tools have also been implemented and successfully applied to industrially relevant case studies [HHWT95,LPY97].

Recently, a useful extension of timed automata has been proposed: *weighted timed automata*[1] [ALP01,BFH$^+$01]. Weighted timed automata are natural models for embedded systems where, often, resources consumptions have to be modeled. They extend classical timed automata with a cost function $\mathcal{C}$ that maps every location and every edge to a nonnegative integer (or rational) number. For a location l, $\mathcal{C}(l)$ represents the cost per time unit for staying in location l. For an edge e, $\mathcal{C}(e)$ represents the cost of crossing the edge. As a consequence, an accumulated cost can be associated to each run of a weighted timed automata and optimization problems can be defined. The *cost-optimal reachability problem* for a weighted timed automaton $\mathcal{A}$ asks, given a location l of $\mathcal{A}$, what is the minimal accumulated cost of a run that visits l in $\mathcal{A}$?

Two different algorithmic solutions have been proposed independently to solve the cost-optimal reachability problem. First, in [ALP01], Alur et al. propose a non-trivial extension of the region automaton to solve the cost-optimal reachability problem. This construction is the basis for an EXPTIME solution to the problem. The optimality of the proposed solution is not studied there. Second, in [BFH$^+$01], Larsen et al. propose a symbolic algorithm that manipulates priced (weighted) extensions of zones. This second solution does not provide a complexity result: the termination

⋆ Supported by the FRFC project "Centre Fédéré en Vérification" funded by the Belgian National Science Foundation (FNRS) under grant nr 2.4530.02

[1] The terminology of "priced timed automata" is also used.

of the algorithm is ensured by a well-quasi order for which the length of descending chains is not studied. The decidability of the cost-optimal reachability problem can also be derived from a paper by Kesten et al. [KPSY99] where some subclasses of integration graphs are shown decidable. In particular, weighted timed automata are integration graphs with a single integrator test along each run (when entering the location l).

In this paper, we further study the cost-optimal reachability problem. Our results are threefold. *First*, we show that the cost-optimal reachability problem can be solved for a more general class of weighted timed automata: positive as well as negative costs on edges and locations can be handled simultaneously. As a consequence, we study the computation of the infimum and the supremum of costs for reachability. This extension is of practical interest. In fact, assume that a weighted timed automaton $\mathcal{A}$ models the behaviors of an embedded controller and its environment. Assume that the objective of the controller is to force the system to reach a given location with an optimal cost whatever does the environment. To measure the quality of a fixed controller, one can consider the worst-case cost, that is, the supremum cost of runs performed by this controller over all possible behaviours of the environment. The smaller is this worst-case cost, the better is the controller. Our method does not find the optimal controller (which is impossible because of the results of [BBR05]), but allows evaluating and comparing controllers. *Second*, we settle the exact complexity of the cost-optimal reachability problem in weighted timed automata with positive and negative costs. We show that this problem is PSPACE-COMPLETE. *Third*, our solution comes in the form of an extension of the region graph which is simpler than the one proposed initially in [ALP01]. Our construction exploits an important property of optimal runs: optimal runs only contain time-transitions with a time τ arbitrarily close to an integer.

Our optimal algorithm relies on two main ingredients. First, we study a simpler version of the cost-optimal reachability problem: the *cost-optimal path reachability problem*. In this problem, a sequence of locations of the underlying timed automaton $\mathcal{A}$ is fixed a priori. Then the problem asks for the optimal time-transitions to switch between the locations of the sequence. We show that this problem is closely related to a linear programming problem. We study the structure of this linear programming problem and show that the associated polyhedron has vertices with integer coordinates. As a consequence, we gain an *important knowledge*: only time-transitions with a time τ arbitrarily close to an integer have to be considered. This important property allows us to propose and justify a simple extension of the classical notion of region called *ε-region*. This notion of ε-region is at the heart of a finite *weighted discrete graph* whose optimal paths are related to optimal runs in the original weighted timed automaton $\mathcal{A}$. The justifications for the correctness of our construction are not straightforward. Indeed, we show that there is no reasonable simulation relation between the states of the weighted discrete graph and the transition graph associated with the weighted timed automaton $\mathcal{A}$. Finally, to obtain an optimal PSPACE algorithm, we show that the construction of the entire weighted discrete graph can be avoided and that this graph can be analyzed without being explicitly constructed.

Our approach easily extends to weighted timed automata with a more general cost function $\mathcal{C}$, for instance when the cost of staying a time τ in location l is computed as $\mathcal{C}(l) \cdot \ln(\tau)$ instead of $\mathcal{C}(l) \cdot \tau$. Indeed, the linear programming problem related to the cost-optimal path reachability problem can still be solved in the more general case of concave and convex cost functions. Moreover, since the notion of ε-region proposed in this paper is only dependent on the fact that the associated polyhedron has vertices with integer coordinates, the weighted discrete graph can be easily adapted to more general cost functions under mild hypotheses.

Other related works. In [ACH93], the authors study the reachability problem for timed automata augmented with costs. Timed automata augmented with costs are a simple class of hybrid automata. The decidability border for hybrid automata has been extensively studied (for surveys see [Hen96,Ras05]). Among the numerous results about this problem, let us mention the following ones. The important class of *initialized rectangular automata* has a decidable reachability problem; however several slight generalizations of these automata lead to an undecidable reachability problem, in particular for timed automata augmented with one stopwatch [HKPV95]. The reachability problem is also undecidable for the simple class of *constant slope hybrid systems* which are timed automata augmented with integrators; the reachability problem becomes decidable when the integrators are used as *observers* (they are neither reset nor tested) [KPSY99].

The optimal reachability problem has also recently been studied in a game setting. In this setting, we are interested in synthesizing optimal strategies for reachability objectives in weighted timed automata. In [ABM04], Alur et al. show that optimal strategies for reachability in less than k transitions can be computed. In [BCFL04], the authors show that optimal strategies for reachability can be computed for a restricted class of weighted timed automata that respect the condition of *strong non-zenoness of cost*. Recently in [BBR05], it is shown that, in the general case, optimal strategies can not be constructed algorithmically. The interesting subcase of time-optimal strategies is solved in [AM99].

In [LR05], Larsen and Rasmussen consider the problem of determining the minimal cost of reaching a given target location, with respect to some primary cost variable, while respecting upper bound constraints on the remaining (secondary) cost variables. The proposed algorithm is an extension of the algorithm presented in [BFH$^+$01].

In [BBL04], the optimal way of staying into a designated set of safe locations is studied. The construction proposed in [BBL04], called corner point abstraction, shares several ideas with the construction proposed here for the weighted discrete graph.

Organization of the paper. In Sect. 2, we recall the notion of timed automaton, region graph and weighted timed automaton.

In Sect. 3, we introduce the cost-optimal reachability problem and we announce our main result that it is PSPACE-COMPLETE. We also introduce the simpler problem of cost-optimal path reachability. We show that solving this problem reduces in solving a linear programming problem. When studying further the related linear programming problem, we deduce the important observation that optimal runs have time-transitions with a time τ arbitrarily close to an integer.

In Sect. 4, we prove that the cost-optimal reachability problem is PSPACE-COMPLETE. PSPACE-HARDNESS is straightforward. The proof of PSPACE-EASYNESS needs several steps. First, due to the previous observation, we refine the classical notion of region with the concept of ε-region. We therefore define the ε-region graph. Second, while there is no natural simulation between states of the ε-region graph and the underlying weighted timed automaton, we are able to relate them in a weaker way (this relation is not straightforward). Third we propose the notion of weighted discrete graph where the cost-optimal reachability problem can be reformulated and solved with a PSPACE-complexity.

In Sect. 5, we show that some assumptions made at the beginning of the paper can be discarded without loss of generality. In Sect. 6, we illustrate the interest of computing infimum and supremum costs in the context of timed games. Finally we give a conclusion in the last section.

2 Preliminaries

In this section, we recall the notions of timed automaton and region graph [AD94]. We introduce the concept of weighted timed automaton [ALP01,BFH$^+$01].

2.1 Timed automaton

Notations. Throughout the paper, we denote by $X = \{x_1, \ldots, x_n\}$ a set of n *clocks*. A *clock valuation* is a map $\nu : X \to \mathbb{R}^+$, where $\mathbb{R}^+$ denotes the set of non-negative real numbers. For $i \in \{1, ..., n\}$, we denote by ν_i the image of the clock x_i by ν, *i.e.* $\nu(x_i) = \nu_i$. Given a clock valuation ν, when no confusion is possible, we also denote by ν the n-tuple of clock values $(\nu_1, ..., \nu_n)$. Let ν be a clock valuation and $\tau \in \mathbb{R}^+$, $\nu + \tau$ is the clock valuation defined by $(\nu_1 + \tau, \ldots, \nu_n + \tau)$. A *guard* is any finite conjunction of expressions of the form $x_i \sim c$ or $x_i - x_j \sim c$ where x_i, x_j are clocks, $c \in \mathbb{N}$ is an integer constant, and $\sim$ is one of the symbols in $\{<, \leq, =, >, \geq\}$. We denote by $\mathcal{G}$ the set of guards. Let g be a guard and ν be a clock valuation, notation $\nu \models g$ means that $(\nu_1, ..., \nu_n)$ satisfies g. A *reset* $Y \in 2^X$ indicates which clocks are reset to 0.

Definition 1. *A* timed automaton $\mathcal{A} = (L, X, E, \mathcal{I})$ *has the following components:* (i) L *is a finite set of* locations, (ii) X *is a set of* clocks, (iii) $E \subseteq L \times \mathcal{G} \times 2^X \times L$ *is a finite set of* edges *and* (iv) $\mathcal{I} : L \to \mathcal{G}$ *assigns an* invariant *to each location.*

The *semantics* of a timed automaton $\mathcal{A}$ is given by its transition system $T_{\mathcal{A}}$.

Definition 2. *A timed automaton* $\mathcal{A} = (L, X, E, \mathcal{I})$ *generates a* transition system $T_{\mathcal{A}} = (Q, \to)$ *with a set of* states *Q equal to*

$$\{(l, \nu) \mid l \in L, \nu \in (\mathbb{R}^+)^n, \nu \models \mathcal{I}(l)\}$$

and a transition relation

$$\rightarrow \; = \bigcup_{\tau \in \mathbb{R}^+} \xrightarrow{\tau} \; \cup \; \bigcup_{e \in E} \xrightarrow{e}$$

defined by

- time-transition $(l,\nu) \xrightarrow{\tau} (l',\nu')$: *if* $l = l'$ *and* $\nu' = \nu + \tau$,
- switch-transition $(l,\nu) \xrightarrow{e} (l',\nu')$: *if* $e = (l,g,Y,l') \in E$, $\nu \models g$ *and* $\nu'_i = 0$ *if* $x_i \in Y$, $\nu'_i = \nu_i$ *otherwise.*

A time-transition corresponds to an *elapse of time* at a location l, and a switch-transition corresponds to an instantaneous switch from a location l to a location l'.

Remark 1. Let us notice that notation $(l,\nu) \rightarrow (l',\nu')$ is ambiguous in some very particular cases, since it can represent both a time-transition and a switch-transition. Indeed, one could have both $(l,\nu) \xrightarrow{\tau} (l,\nu')$ with $\tau = 0$ and $(l,\nu) \xrightarrow{e} (l,\nu')$ for some $e \in E$. However we use it in order to avoid a too heavy notation.

Remark 2. In this paper, we only consider bounded and diagonal-free timed automata. A timed automaton is *diagonal-free* if the guards used in the edges and the invariants contain no expression of the form $x_i - x_j \sim c$, with x_i, x_j being clocks, $c \in \mathbb{N}$ and $\sim \in \{<,\leq,=,\geq,>\}$. A timed automaton $\mathcal{A}$ is *bounded* if for each location l, the invariant $\mathcal{I}(l)$ is upper bounded on all clocks. In other words, there exists a constant M such that each state (l,ν) of $T_{\mathcal{A}}$ satisfies $\nu_i \leq M$ for all $i \in \{1,\ldots,n\}$. In Sect. 5, we explain why these two hypotheses are not restrictions.

The states (l,ν) of $T_{\mathcal{A}}$ are shortly denoted by q. Given $q = (l,\nu) \in Q$ and $\tau \in \mathbb{R}^+$, we denote by $q + \tau$ the state $(l, \nu + \tau)$.

A *run* ρ of $T_{\mathcal{A}}$ is a finite path

$$\rho = q_0 \rightarrow q_1 \rightarrow \cdots \rightarrow q_m.$$

It is also shortly denoted $\rho = q_0 \rightsquigarrow q_m$. The run ρ is called *initialized* if q_0 is of the form $(l,0)$ with all the clock values being null. We say that ρ is *canonical* if it is of the form $q_0 \xrightarrow{\tau_1} q_1 \xrightarrow{e_1} q_2 \xrightarrow{\tau_2} q_3 \xrightarrow{e_2} q_4 \cdots$ where time-transitions and switch-transitions alternate.

Remark 3. A canonical (initialized) run can be associated with any (initialized) run $\rho = q_0 \rightarrow \cdots \rightarrow q_m$. Indeed any two consecutive time-transitions $q_k \xrightarrow{\tau} q_{k+1} \xrightarrow{\tau'} q_{k+2}$ can be replaced by the time-transition $q_k \xrightarrow{\tau+\tau'} q_{k+2}$, and time-transition $q_k \xrightarrow{\tau} q_{k+1}$ with $\tau = 0$ is allowed in Definition 2.

Remark 4. Let ρ be the following canonical initialized run

$$q'_0 = (l_0, 0) \xrightarrow{\tau_1} q_1 \xrightarrow{e_1} q'_1 \xrightarrow{\tau_2} q_2 \xrightarrow{e_2} q'_2 \cdots \xrightarrow{\tau_k} q_k \xrightarrow{e_k} q'_k \cdots .$$

Given $q_k = (l_k, \nu^k)$ a state of ρ, the clock values $(\nu^k_1, \ldots, \nu^k_n)$ at q_k depend on $\{\tau_1, ..., \tau_k\}$ as follows : the value ν^k_i of the clock x_i at state q_k is equal to

$$\nu^k_i = \tau_{h+1} + \tau_{h+2} + \cdots + \tau_{k-1} + \tau_k$$

with $0 \leq h \leq k$ such that $q_h \xrightarrow{e_h} q'_h$ is the last transition of ρ where the clock x_i has been reset[2].

2.2 Region graph

In this section, we define the region graph of a timed automaton $\mathcal{A} = (L, X, E, \mathcal{I})$. We first recall the usual equivalence on clock valuations and its extension to the states of $T_{\mathcal{A}}$. For every clock x_i, let c_i be the largest constant that x_i is compared with in any guard of E and any invariant of $\mathcal{I}$. For $\tau \in \mathbb{R}^+$, $\lfloor \tau \rfloor$ denotes its integral part and $\bar{\tau}$ denotes its fractional part.

Definition 3. *Two clock valuations ν and ν' are equivalent, $\nu \approx \nu'$, iff the following conditions hold*

[2] We notice that h depends on i.

– $\lfloor \nu_i \rfloor = \lfloor \nu'_i \rfloor$ *or* $\nu_i, \nu'_i > c_i$, *for all* $i \in \{1, \ldots, n\}$;
– $\bar{\nu}_i \leq \bar{\nu}_j$ *iff* $\bar{\nu}'_i \leq \bar{\nu}'_j$, *for all* $i \neq j \in \{1, \ldots, n\}$ *with* $\nu_i \leq c_i, \nu_j \leq c_j$;
– $\bar{\nu}_i = 0$ *iff* $\bar{\nu}'_i = 0$, *for all* $i \in \{1, \ldots, n\}$ *with* $\nu_i \leq c_i$.

The equivalence relation $\approx$ *is extended to the states of* $T_{\mathcal{A}}$ *as follows*

$$q = (l, \nu) \ \approx \ q' = (l', \nu') \quad \textit{iff} \quad l = l' \textit{ and } \nu \approx \nu'.$$

We use $[\nu]$ (resp. $[q]$) to denote the equivalence class to which ν (resp. q) belongs. A *region* is an equivalence class $[q]$. The set of all the regions is denoted by R. A region $[q]$ is *closed* if $q + \tau \not\approx q$ for any $\tau > 0$, otherwise it is *open*. A region $[q]$ is *unbounded* if it satisfies $q = (l, \nu)$ with $\nu_i > c_i$ for some $i \in \{1, \ldots, n\}$, otherwise it is *bounded*.

We notice that since timed automata are supposed to be bounded (see Remark 2), the states of any run of $T_{\mathcal{A}}$ never belong to an unbounded region.

Remark 5. A nice representation of the regions has been introduced in [ACH93]. A region is fully specified by a location l, the integral parts of the clock values $(\nu_1, \ldots, \nu_n)$, and the ordering of their fractional parts for the clocks x_i such that $\nu_i \leq c_i$. The representation proposed in [ACH93] consists in visualizing this ordering. For example, the ordering $0 < \bar{\nu}_1 < \bar{\nu}_2 < \cdots < \bar{\nu}_n < 1$ is depicted on Fig. 1.

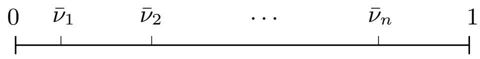

Fig. 1. The ordering of the fractional parts of the clock values in a region

We now define the region graph of a timed automaton $\mathcal{A}$ which is nothing else than the quotient of $T_{\mathcal{A}}$ by $\approx$.

Definition 4. *Let* $\mathcal{A}$ *be a timed automaton. The* region graph $R_{\mathcal{A}} = (R, \rightarrow)$ *is the finite graph given by* $T_{\mathcal{A}}/_{\approx}$. *Its vertex set is equal to* R. *Its edge set is composed of the edges* $r \rightarrow r'$, *with* $r, r' \in R$, *such that there exist two states* $q \in r$, $q' \in r'$, *and a transition* $q \rightarrow q'$ *in* $T_{\mathcal{A}}$. *The edge* $r \rightarrow r'$ *is called a* switch-edge *(resp.* time-edge*) if* $q \rightarrow q'$ *is a switch-transition (resp. time-transition).*

Given two distinct bounded regions $r = [q]$, $r' = [q']$, we say that r' is a *successor* of r, written $r' = \mathrm{succ}(r)$, if $\exists \tau \in \mathbb{R}^+$, $q + \tau \in r'$, and $\forall \tau' < \tau$, $q + \tau' \in r \cup r'$.

Given a run $\rho = q_0 \rightarrow q_1 \rightarrow \cdots \rightarrow q_m$ of $T_{\mathcal{A}}$, we denote by $[\rho]$ the corresponding path $[q_0] \rightarrow [q_1] \rightarrow \cdots \rightarrow [q_m]$ in $R_{\mathcal{A}}$. Notice that due to Remark 2, each region $[q_k]$ with $k \in \{0, \ldots, m\}$, is bounded. We say that a path ρ_R in $R_{\mathcal{A}}$ is *canonical* (resp. *initialized*) if $\rho_R = [\rho]$ for some canonical (resp. initialized) run ρ of $T_{\mathcal{A}}$. We use the notation $\rho_R = r \rightsquigarrow r'$ for a path in $R_{\mathcal{A}}$ starting with the region r and ending with the region r'. Let us notice that we only consider finite paths of $R_{\mathcal{A}}$ in this paper.

Remark 6. We recall [AD94] that the size $|R_{\mathcal{A}}|$ of the region graph, i.e. its number of regions and edges, is in $\mathcal{O}((|L| + |E|)2^{|\delta(\mathcal{A})|})$ where $\delta(\mathcal{A})$ is the binary encoding of the constants (guards and costs) appearing in $\mathcal{A}$. Thus $|R_{\mathcal{A}}|$ is in $\mathcal{O}(2^{|\mathcal{A}|})$ where $|\mathcal{A}|$ takes into account the locations, edges and constants of $\mathcal{A}$.

2.3 Weighted timed automaton

We now introduce the notion of weighted timed automaton[3], which is an extension of timed automaton with costs on both locations and edges.

Definition 5. *A* weighted timed automaton *is a timed automaton* $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{C})$ *augmented with a* cost function $\mathcal{C} : L \cup E \rightarrow \mathbb{Z}$ *which assigns an integer cost to both locations and edges.*

[3] This model differs from the one used in [ALP01,BFH$^+$01] since it allows negative costs.

The semantics of a weighted timed automaton $\mathcal{A}$ associates a *cost* with each run of $T_{\mathcal{A}}$ in the following way.

Definition 6. *Let $\mathcal{A}$ be a weighted timed automaton and $\rho = q'_0 \xrightarrow{\tau_1} q_1 \xrightarrow{e_1} q'_1 \xrightarrow{\tau_2} q_2 \xrightarrow{e_2} q'_2 \cdots \xrightarrow{\tau_m} q_m \xrightarrow{e_m} q'_m$ be a canonical run of $T_{\mathcal{A}}$. Let l_k be the location of q_k (and q'_{k-1}) for each k. Then the* cost[4] *$\mathcal{C}(\rho)$ of ρ is equal to $\mathcal{C}_d(\rho) + \mathcal{C}_s(\rho)$ with*

$$\mathcal{C}_d(\rho) = \sum_{k \in \{1,\ldots,m\}} \mathcal{C}(l_k) \cdot \tau_k, \qquad \mathcal{C}_s(\rho) = \sum_{k \in \{1,\ldots,m\}} \mathcal{C}(e_k).$$

In the previous definition, $\mathcal{C}_d(\rho)$ is called the *duration cost* of ρ, and $\mathcal{C}_s(\rho)$ the *switch cost* of ρ.

Example 1. Let $\mathcal{A}$ be the weighted timed automaton pictured on Fig. 2. The cost of each location is indicated on the figure and the cost of each edge is null. The invariant $(x_1 \leq 4) \wedge (x_2 \leq 2)$ is assigned to each location, showing that $\mathcal{A}$ is bounded.

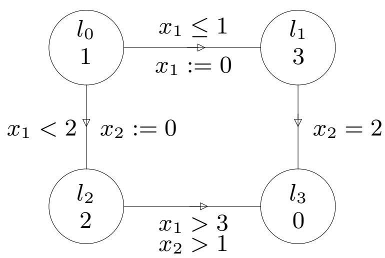

Fig. 2. A weighted timed automaton

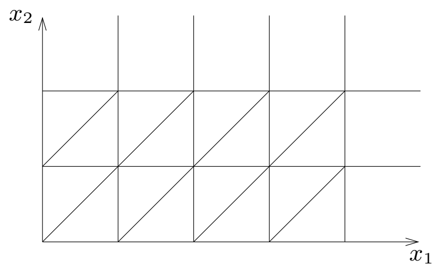

Fig. 3. Its equivalence relation $\approx$

The canonical run

$$\rho = (l_0, 0, 0) \xrightarrow{0.5} (l_0, 0.5, 0.5) \rightarrow (l_1, 0, 0.5) \xrightarrow{1.5} (l_1, 1.5, 2) \rightarrow (l_3, 1.5, 2)$$

has a cost equal to $\mathcal{C}_d(\rho) = \mathcal{C}(l_0) \cdot 0.5 + \mathcal{C}(l_1) \cdot 1.5 = 5$.

3 Cost-optimal reachability problem

In this section, we define the cost-optimal reachability problem for weighted timed automata [BFH$^+$01].[5]

Definition 7. *Let $\mathcal{A}$ be a weighted timed automaton. Given two regions r, r' of $R_{\mathcal{A}}$, the* optimal cost $\mathsf{OptCost}(r, r')$ *of reaching r' from r is the infimum (resp. supremum) of the costs of the runs $\rho = q \rightsquigarrow q'$ of $T_{\mathcal{A}}$ such that $q \in r$ and $q' \in r'$.*

Moreover, we say that $\mathsf{OptCost}(r, r')$ *is* realizable *if there exists such a run ρ such that $\mathcal{C}(\rho) = \mathsf{OptCost}(r, r')$.*

Remark 7. In the previous definition, suppose that the infimum cost is considered. By convention $\mathsf{OptCost}(r, r') = +\infty$ in the case there is no run $\rho = q \rightsquigarrow q'$ such that $q \in r$ and $q' \in r'$. Otherwise, $\mathsf{OptCost}(r, r') \in \mathbb{R}$ or $\mathsf{OptCost}(r, r') = -\infty$. Symmetric observations hold when the supremum cost is considered.

Problem 1. (Cost-optimal reachability problem) Given $\mathcal{A}$ a weighted timed automaton, and two regions r, r' of $R_{\mathcal{A}}$, compute the optimal cost $\mathsf{OptCost}(r, r')$.

Our main result is the following one. The rest of the paper is devoted to its proof.

[4] In the case ρ ends with a time-transition, *i.e.* there is an additional transition $q'_m \xrightarrow{\tau_{m+1}} q_{m+1}$, then there is an additional term $\mathcal{C}(l_{m+1}) \cdot \tau_{m+1}$ in both $\mathcal{C}(\rho)$ and $\mathcal{C}_d(\rho)$.

[5] In this paper, by cost-optimality we mean both infimum cost and supremum cost, while only infimum cost is studied in [BFH$^+$01].

Theorem 1. *The cost-optimal reachability problem is* PSPACE-COMPLETE.

Remark 8. In the sequel, we make two *assumptions* for solving Problem 1. First, we suppose that the region r given in Problem 1 is composed of a unique state of the form $(l, 0)$ such that all the clock values are null. Second, we focus only on the computation of the infimum cost. Indeed these two assumptions can be discarded with little effort (see Sect. 5).

Remark 9. Problem 1 refers to the computation of $\mathsf{OptCost}(r, r')$ for two regions r, r' of $R_{\mathcal{A}}$. An alternative problem is the computation of $\mathsf{OptCost}(q, q')$ where $q = (l, \nu), q' = (l, \nu')$ are two given states of $T_{\mathcal{A}}$. When q, q' have rational clocks values ν, ν', the optimal cost $\mathsf{OptCost}(q, q')$ can be computed by using our method for Problem 1. The arguments are the following ones. Let $\lambda \in \mathbb{N}$ be such that $\lambda \cdot \nu$, $\lambda \cdot \nu'$ are integers. Let $\mathcal{A}_\lambda$ be the automaton obtained from the weighted timed automaton $\mathcal{A}$ by replacing

- each constant c in each guard and invariant of $\mathcal{A}$ by $\lambda \cdot c$;
- each cost $\mathcal{C}(e)$ of each edge e by $\lambda \cdot \mathcal{C}(e)$.

In this way the "granularity" of time has been modified, such that $(l, \nu) \rightsquigarrow (l', \nu')$ is a run of $\mathcal{A}$ with cost κ iff $(l, \lambda \cdot \nu) \rightsquigarrow (l', \lambda \cdot \nu')$ is a run of $\mathcal{A}_\lambda$ with cost $\lambda \cdot \kappa$ (see also [AD94]). Therefore computing $\mathsf{OptCost}(q, q')$ in $\mathcal{A}$ is equivalent to computing $\frac{1}{\lambda}\mathsf{OptCost}(r, r')$ in $\mathcal{A}_\lambda$ where the region r (resp. r') of $R_{\mathcal{A}_\lambda}$ is composed of the unique state $(l, \lambda \cdot \nu)$ (resp. $(l', \lambda \cdot \nu')$).

The next example indicates how the cost-optimal reachability problem is related to a linear programming problem (see the book [NW88] for details on linear programming).

Example 2. We consider again the weighted timed automaton of Fig. 2. We are interested in runs from l_0 to l_3.[6] There are mainly two families of such runs, the runs going through l_1, and the runs going through l_2. The first family can be described by the following parameterized run[7]

$$\rho_1(t_1, t_2) = (l_0, 0, 0) \xrightarrow{t_1} (l_0, t_1, t_1) \rightarrow (l_1, 0, t_1) \xrightarrow{t_2} (l_1, t_2, t_1 + t_2) \rightarrow (l_3, t_2, t_1 + t_2).$$

The parameters t_1, t_2 represent the time elapsed at locations l_0, l_1 respectively. They are constrained by the next linear inequalities

$$0 \leq t_1 \leq 1, t_2 \geq 0 \text{ and } t_1 + t_2 = 2. \tag{1}$$

The cost of the parameterized run $\rho_1(t_1, t_2)$ is given by $t_1 + 3 \cdot t_2$. Therefore to find the infimum cost with respect to the first family of runs reduces in computing the infimum value of the function $t_1 + 3 \cdot t_2$ under the constraints (1). This is a linear programming problem for which it is known that the optimal solution is given by one of the vertices of the polyhedron defined by (1), here the point $(1, 1)$ leading to the infimum cost 4. On Fig. 4, the bold line represents this polyhedron, and the dashed line represents the situation of an optimal cost $t_1 + 3 \cdot t_2 = 4$. Note that the optimum cost 4 is a minimum cost since it is realized by the run $\rho_1(t_1, t_2)$ with $t_1 = t_2 = 1$.

Similarly the second family of runs is described by the following parameterized run

$$\rho_2(t_1, t_2) = (l_0, 0, 0) \xrightarrow{t_1} (l_0, t_1, t_1) \rightarrow (l_2, t_1, 0) \xrightarrow{t_2} (l_2, t_1 + t_2, t_2) \rightarrow (l_3, t_1 + t_2, t_2).$$

In this case, parameters t_1, t_2 are constrained by the linear inequalities

$$0 \leq t_1 < 2, t_2 > 1 \text{ and } t_1 + t_2 > 3. \tag{2}$$

The cost with respect to $\rho_2(t_1, t_2)$ is given by $t_1 + 2 \cdot t_2$. on Fig. 5, the shaded zone represents the polyhedron defined by (2), and the dashed line represents the situation of the infimum cost $t_1 + 2 \cdot t_2 = 4$. This infimum cost is not a

[6] In this example, we work with locations, instead of regions as indicated in Definition 7.

[7] We can suppose that this run is canonical by Remark 3 and that it is initialized by Remark 8. Moreover we can assume that this run ends with a switch-transition since we consider the infimum cost to reach l_3. We also notice the form of the clock values as described in Remark 4.

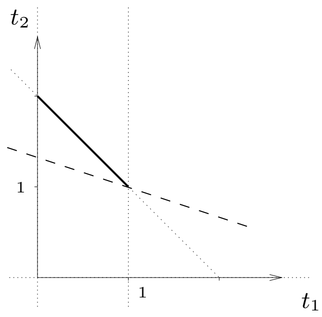

Fig. 4. Optimizing the cost of $\rho_1(t_1, t_2)$

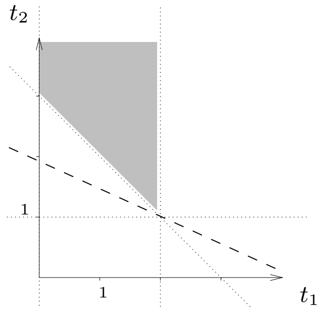

Fig. 5. Optimizing the cost of $\rho_2(t_1, t_2)$

minimum cost since no run realizes it. Indeed the value 4 is achieved at the vertex $(2, 1)$ of the polyhedron, a point that does not belong to it.

Therefore in this simple example, the infimum cost of reaching location l_3 from location l_1 is equal to 4, and it is realizable. This value has been obtained by solving a linear programming problem for the two parameterized runs $\rho_1(t_1, t_2)$ and $\rho_2(t_1, t_2)$.

In order to solve the cost-optimal reachability problem, we first study an easier problem: the *cost-optimal path reachability problem*. It is related to a given path in the region graph $R_{\mathcal{A}}$ of a weighted timed automaton $\mathcal{A}$. We define this simpler problem in Section 3.1 below. We show in Section 3.2 that solving the cost-optimal path reachability problem reduces in solving a linear programming problem. In Sections 3.3 and 3.4, we investigate further the approach by linear programming. The obtained results will be a first step toward the solution of Problem 1 given in Sect. 4.

3.1 Cost-optimal path reachability problem

Definition 8. *Let $\mathcal{A}$ be a weighted timed automaton. Given a canonical initialized path ρ_R in $R_{\mathcal{A}}$, the* optimal cost $\mathsf{OptCost}(\rho_R)$ associated with ρ_R *is the infimum of the costs $\mathcal{C}(\rho)$ among the runs ρ of $T_{\mathcal{A}}$ such that $[\rho] = \rho_R$.*

Moreover, we say that $\mathsf{OptCost}(\rho_R)$ is realizable *if there exists such a run ρ such that $\mathcal{C}(\rho) = \mathsf{OptCost}(\rho_R)$.*

Remark 10. In the previous definition, we can suppose that ρ_R is canonical and initialized due to Remarks 3 and 8.

Problem 2. (Cost-optimal path reachability problem) Given $\mathcal{A}$ a weighted timed automaton, and ρ_R a canonical initialized path in $R_{\mathcal{A}}$, compute the optimal cost $\mathsf{OptCost}(\rho_R)$ associated with ρ_R.

Remark 11. We notice that given ρ_R a path of $R_{\mathcal{A}}$, we have $\mathcal{C}_s(\rho) = \mathcal{C}_s(\rho')$ whenever $[\rho] = [\rho'] = \rho_R$. Hence the cost-optimal path reachability problem reduces in computing the optimal duration cost $\mathcal{C}_d$.

3.2 A linear programming problem

In this section we show that solving Problem 2 reduces in solving a linear programming problem. This idea was already illustrated in Example 2. Before we formalize this idea, we go further with this example.

Example 3. We come back to the weighted timed automaton of Fig. 2 and its equivalence relation $\approx$ given on Fig. 3. We consider the following path ρ_R in $R_{\mathcal{A}}$

$$\rho_R = r'_0 \to r_1 \to r'_1 \to r_2 \to r'_2$$

with the regions

$$\begin{aligned} r'_0 &= (l_0, 0, 0), \\ r_1 &= (l_0, 0 < x_1 = x_2 < 1), \\ r'_1 &= (l_1, x_1 = 0, 0 < x_2 < 1), \\ r_2 &= (l_1, 1 < x_1 < 2, x_2 = 2), \\ r'_2 &= (l_3, 1 < x_1 < 2, x_2 = 2). \end{aligned}$$

Each run ρ of $T_{\mathcal{A}}$ such that $[\rho] = \rho_R$ can be parameterized as

$$\rho(t_1, t_2) = (l_0, 0, 0) \xrightarrow{t_1} (l_0, t_1, t_1) \to (l_1, 0, t_1) \xrightarrow{t_2} (l_1, t_2, t_1 + t_2) \to (l_3, t_2, t_1 + t_2)$$

with the two parameters t_1, t_2 constrained by the next linear inequalities

$$0 < t_1 < 1, 1 < t_2 < 2 \text{ and } t_1 + t_2 = 2. \tag{3}$$

These constraints are obtained as follows. We have $r_1 = [(l_0, t_1, t_1)]$ justifying the first inequality, and $r_2 = [(l_1, t_2, t_1 + t_2)]$ justifying the second and third inequalities.

In the same way it has been done in Example 2, we compute $\mathsf{OptCost}(\rho_R)$ as equal to 4. Indeed, it is equal to the infimum value of the cost $\mathcal{C}(\rho(t_1, t_2)) = t_1 + 3 \cdot t_2$ under the constraints (3). This optimal cost is not realizable. However it can be approximated by $\rho(1 - \varepsilon, 1 + \varepsilon)$ with $\varepsilon > 0$ arbitrarily small.

We now generalize arguments of Example 3 to any canonical initial path ρ_R of Definition 8. We suppose that it has the following form with the last edge being a switch-edge[8]:

$$\rho_R = r'_0 \to r_1 \to r'_1 \to r_2 \cdots \to r_m \to r'_m. \tag{4}$$

In this path, each region r_k (resp. r'_k) is bounded since the timed automata studied in this paper are supposed to be bounded (see Remark 2).

We recall the basic fact [AD94] that each region r of $\mathcal{A}$ can be described by a location and a finite set of linear constraints of the form

$$x_i - x_j \sim c \;\text{ or }\; x_i \sim c \tag{5}$$

where x_i, x_j are clocks, $c \in \mathbb{Z}$ and $\sim \in \{<, \leq, =, \geq, >\}$. We denote this set of linear constraints by $r(x_1, ..., x_n)$.

All runs ρ of $T_{\mathcal{A}}$ such that $[\rho] = \rho_R$ can be parameterized as

$$\rho(t_1, t_2, \ldots, t_m) = q'_0 \xrightarrow{t_1} q_1 \xrightarrow{e_1} q'_1 \xrightarrow{t_2} q_2 \xrightarrow{e_2} \cdots \xrightarrow{t_m} q_m \xrightarrow{e_m} q'_m \tag{6}$$

where

- the first state is of the form $q'_0 = (l_1, 0)$ such that $r'_0 = [(l_1, 0)]$,
- each other state can be written as $q_k = (l_k, x^k) = (l_k, x^k_1, x^k_2, \ldots, x^k_n)$ (resp. $q'_k = (l_{k+1}, x'^k)$) such that each clock x^k_i (resp. x'^k_i) depends on the parameters $t_1, t_2, \ldots, t_k$.

For state q_k, this dependence $x^k_i = x^k_i(t_1, \ldots, t_k)$ is given in Remark 4:

$$x^k_i(t_1, \ldots, t_k) = t_{h+1} + t_{h+2} + \cdots + t_{k-1} + t_k \tag{7}$$

with $0 \leq h \leq k$ such that $q_h \xrightarrow{e_h} q'_h$ is the last transition of $\rho(t_1, \ldots, t_m)$ where the clock x_i has been reset. For state q'_k, with $e_k = (l_k, g_k, Y_k, l_{k+1})$, we have

$$\begin{aligned} x'^k_i(t_1, \ldots, t_k) &= 0 \text{ if } x^k_i \in Y_k \\ &= x^k_i \text{ otherwise.} \end{aligned} \tag{8}$$

Since $[\rho(t_1, \ldots, t_m)] = \rho_R$, we have $r_k = [q_k]$ for all $k \in \{1, \ldots, m\}$, this shows that the parameters $t_1, \ldots, t_m$ are constrained by the following set of inequalities

$$\mathsf{Constr}(\rho_R) = \bigcup_{k \in \{1, \ldots, m\}} r_k(t_1, \ldots, t_k) \tag{9}$$

$r_k(t_1, \ldots, t_k) = r_k(x^k_1(t_1, \ldots, t_k), \ldots, x^k_n(t_1, \ldots, t_k))$.

Therefore for all runs ρ of $T_{\mathcal{A}}$ such that $[\rho] = \rho_R$, we can write $\rho = \rho(\tau_1, \ldots, \tau_m)$ such that $(\tau_1, \ldots, \tau_m) \in (\mathbb{R}^+)^m$ satisfy the constraints of $\mathsf{Constr}(\rho_R)$.

[8] The case where the last edge is a time-edge can be treated similarly. All the results of Sect. 3.2 remain valid.

Example 4. Let us illustrate the previous notation on the path ρ_R of Example 3. The set $\mathsf{Constr}(\rho_R)$ is composed of the following linear constraints

– region $r_1 : 0 < t_1 < 1$,
– region $r_2 : 1 < t_2 < 2, t_1 + t_2 = 2$.

They have been obtained as follows. From $r_1 = (l_0, 0 < x_1 = x_2 < 1)$ with the two clocks equal to t_1, we obtain the first constraint $0 < t_1 < 1$. From $r_2 = (l_1, 1 < x_1 < 2, x_2 = 2)$ with the two clocks x_1, x_2 respectively equal to t_2 and $t_1 + t_2$, we obtain the second and third constraints $1 < t_2 < 2$ and $t_1 + t_2 = 2$.

We now define the two following subsets of $(\mathbb{R}^+)^m$:

$$A(\rho_R) = \{(\tau_1, \ldots, \tau_m) \in (\mathbb{R}^+)^m \mid [\rho(\tau_1, \ldots, \tau_m)] = \rho_R\},$$
$$B(\rho_R) = \{(\tau_1, \ldots, \tau_m) \in (\mathbb{R}^+)^m \mid (\tau_1, \ldots, \tau_m) \models \mathsf{Constr}(\rho_R)\}.$$

This allows us to formulate the next lemma.

Lemma 1. $A(\rho_R) = B(\rho_R)$.

Proof. From above we have $A(\rho_R) \subseteq B(\rho_R)$. For the other inclusion, consider $(\tau_1, ..., \tau_m) \models \mathsf{Constr}(\rho_R)$, we have to prove that $\rho = \rho(\tau_1, \ldots, \tau_m)$ is a run of $T_{\mathcal{A}}$ satisfying $[\rho(\tau_1, \ldots, \tau_m)] = \rho_R$. The proof is by induction on k with $k \in \{0, \ldots, m\}$.

For $k = 0$, we have $q'_0 = (l_1, \nu'^0) = (l_1, 0)$ and $[q'_0] = r'_0$. For correctly starting the induction, we also need a fictitious state $q_0 = (l_0, \nu^0) = (l_0, 0)$ and a fictitious edge $e_0 = (l_0, g_0, Y_0, l_1)$ with $g = \mathsf{true}$ and $Y_0 = X$.

Consider the case $k > 0$. Let e_{k-1} be the edge $(l_{k-1}, g_{k-1}, Y_{k-1}, l_k)$.

By induction, we can suppose that $q'_{k-1} = (l_k, \nu'^{k-1})$ with ν'^{k-1} satisfying (8) and (7), that is

$$\begin{aligned} \nu_i'^{k-1} &= 0 && \text{if the clock } x_i \text{ belongs to } Y_{k-1} \\ &= \nu_i^{k-1} && \text{otherwise} \end{aligned}$$

where

$$\nu_i^{k-1} = \tau_{h+1} + \tau_{h+2} + \cdots + \tau_{k-1}$$

with $0 \leq h \leq k-1$ such that $q_h \xrightarrow{e_h} q'_h$ is the last transition of ρ where the clock x_i has been reset. Moreover, $[q'_{k-1}] = r'_{k-1}$.

Let us now study the form of the states q_k and q'_k.

By definition of a time-transition, we have $q_k = (l_k, \nu^k)$ with

$$\begin{aligned} \nu_i^k &= \tau_k && \text{if } \nu_i'^{k-1} = 0 \\ &= \tau_{h+1} + \tau_{h+2} + \cdots + \tau_{k-1} + \tau_k && \text{otherwise.} \end{aligned}$$

This shows that ν^k satisfies (7). By hypothesis, $\tau_1, \ldots, \tau_k$ satisfy the subset of constraints $r_k(t_1, \ldots, t_k)$ of $\mathsf{Constr}(\rho_R)$. It follows that the transition $q'_{k-1} \xrightarrow{\tau_k} q_k$ is a time-transition of $T_{\mathcal{A}}$ such that $[q_k] = r_k$.

Let e_k be the edge (l_k, g_k, Y_k, l_{k+1}). By definition of a switch-transition, we have $q'_k = (l_{k+1}, \nu'^k)$ with

$$\begin{aligned} \nu_i'^k &= 0 && \text{if the clock } x_i \text{ belongs to } Y_k \\ &= \nu_i^k && \text{otherwise.} \end{aligned}$$

Then we have a switch-transition $q_k \xrightarrow{e_k} q'_k$ such that $[q'_k] = r'_k$ and ν'^k satisfies (8). This ends the case $k > 0$ of the induction. □

In Remark 11, we notice that solving the cost-optimal path reachability problem reduces in computing the optimal duration cost $\mathcal{C}_d$. Looking at the parameterized run $\rho(t_1, \ldots, t_m)$ (see (6)), its duration cost is equal to

$$\mathcal{C}_d(\rho(t_1, \ldots, t_m)) = \sum_{k \in \{1, \ldots, m\}} \mathcal{C}(l_k) \cdot t_k. \tag{10}$$

Thus by Lemma 1, the optimal cost $\mathsf{OptCost}(\rho_R)$ can be obtained by computing the infimum value of $\mathcal{C}_d(\rho(t_1,\ldots,t_m))$ under the set of constraints $\mathsf{Constr}(\rho_R)$.

The set $\mathsf{Constr}(\rho_R)$ defines an m-dimensional *polyhedron* $\mathsf{Pol}(\rho_R)$ equal to

$$\mathsf{Pol}(\rho_R) = \{(\tau_1, ..., \tau_m) \in (\mathbb{R}^+)^m \mid (\tau_1, ..., \tau_m) \models \mathsf{Constr}(\rho_R)\} \tag{11}$$

Notice that this polyhedron is bounded since the set of constraints $\mathsf{Constr}(\rho_R)$ is constructed from bounded regions.

We also define the *closure* of the polyhedron $\mathsf{Pol}(\rho_R)$, denoted by $\overline{\mathsf{Pol}(\rho_R)}$. This polyhedron is obtained by considering the set $\mathsf{Constr}(\rho_R)$ where each constraint (see (5)) of the form $x_i - x_j < c$ or $x_i < c$ (resp. $x_i - x_j > c$ or $x_i > c$) is replaced by $x_i - x_j \leq c$ or $x_i \leq c$ (resp. $x_i - x_j \geq c$ or $x_i \geq c$).[9] Looking at (7), we notice that the constraints of $\mathsf{Constr}(\rho_R)$ have the form

$$t_i + t_{i+1} + \cdots + t_{j-1} + t_j \sim c$$

with $i, j \in \{1, \ldots, m\}$, $c \in \mathbb{Z}$ and $\sim \in \{<, \leq, =, \geq, >\}$. It follows that $\overline{\mathsf{Pol}(\rho_R)}$ can be defined by constraints of the form

$$M \cdot t \leq d, \qquad t \geq 0 \tag{12}$$

where M is a $(p \times m)$ matrix with integer coefficients (for some p), t is the column vector $(t_1, \ldots, t_m)$ such that $t_i \geq 0$ for all $i \in \{1, \ldots, m\}$, and d is a column vector of p integer constants.

As the duration cost is a linear function with integer coefficients (see (10)), the optimum value of $\mathcal{C}_d(\rho(t_1,\ldots,t_m))$ is obtained at one of the vertices of the polyhedron $\overline{\mathsf{Pol}(\rho_R)}$. Due to the form of (12), this can be computed by the *Simplex Method*, a well-known method in linear programming (see [NW88]). In this way, we have shown how to solve Problem 2.

Corollary 1. *Problem 2 is decidable.*

Notice that this problem is in PTIME (in p and m). We recall that m is the length of ρ_R and p is related to the number of constraints of $\mathsf{Constr}(\rho_R)$ defined in (9).

With the linear programming approach, we can also decide whether the optimal cost $\mathsf{OptCost}(\rho_R)$ is realizable.

Corollary 2. *It is decidable whether the optimal cost* $\mathsf{OptCost}(\rho_R)$ *is realizable.*

Proof. Suppose that the minimum value of $\mathcal{C}_d(\rho(\tau_1,\ldots,\tau_m))$ computed by the Simplex Method is equal to b. Recall the form of $\mathcal{C}_d(\rho(\tau_1,\ldots,\tau_m))$ given in (10). Then $\mathsf{OptCost}(\rho_R)$ is realizable if and only if the intersection between

$$\{(\tau_1, ..., \tau_m) \in (\mathbb{R}^+)^m \mid \sum_{k\in\{1,\ldots,m\}} \mathcal{C}(l_k) \cdot \tau_k = b\}$$

and $\mathsf{Pol}(\rho_R)$ is non empty. □

Remark 12. It is important to note that Corollary 1 remains true in the case of more general duration cost functions. For instance, if $\mathcal{C}_d(\rho(t_1,\ldots,t_m))$ is a concave function, then its minimum value is obtained at one of the vertices of the polyhedron $\overline{\mathsf{Pol}(\rho_R)}$ (see [Roc70]). We recall that a function $f(t) = f(t_1,\ldots,t_m)$ is *concave* if

$$f(\lambda t + (1-\lambda)t') \geq \lambda f(t) + (1-\lambda) f(t')$$

with $\lambda \in [0,1]$. Since every $t \in \overline{\mathsf{Pol}(\rho_R)}$ can be written as $t = \sum_k \lambda_k v_k$ with $\sum_k \lambda_k = 1$ and the v_k's being the vertices of $\overline{\mathsf{Pol}(\rho_R)}$, we have

$$f(t) = f(\sum_k \lambda_k v_k) \geq \sum_k \lambda_k f(v_k) \geq \sum_k \lambda_k \min_k\{f(v_k)\} = \min_k\{f(v_k)\}.$$

This shows that the minimum value of $\mathcal{C}_d(\rho(t_1,\ldots,t_m))$ is obtained at the vertex v_l of $\overline{\mathsf{Pol}(\rho_R)}$ such that $f(v_l) = \min_k\{f(v_k)\}$.

Symmetrically, if $\mathcal{C}_d(\rho(t_1,\ldots,t_m))$ is a convex function, then its maximum value is obtained at one of the vertices of $\overline{\mathsf{Pol}(\rho_R)}$ (see [Roc70]). A function $f(t)$ is *convex* if $-f(t)$ is concave.

[9] This definition corresponds to the notion of closure from the topological point of view.

3.3 3-Block matrices

Let $\mathcal{A}$ be a weighted timed automaton, and ρ_R be a canonical initial path in $R_{\mathcal{A}}$. In this section we investigate in more details the form of the polyhedron $\overline{\mathsf{Pol}(\rho_R)}$, and in particular its vertices. This study leads to the nice results given in Corollaries 4 and 5.

Coming back to the form of the matrix M given in (12), we observe that each row of M is composed of three blocks (possibly empty) : a first block of 0's, a second block of 1's (resp. -1's) and a third block of 0's, that is

$$(0,\ldots,0,1,\ldots,1,0,\ldots,0) \text{ or } (0,\ldots,0,-1,\ldots,-1,0,\ldots,0).$$

We call *3-block* a matrix of this form. This particularity of the matrix M will lead to very nice results. First we give an illustration.

Example 5. Considering the path ρ_R of Example 3 with the set $\mathsf{Constr}(\rho_R)$ being composed of the linear constraints

$$0 < t_1 < 1, \quad 1 < t_2 < 2, \quad t_1 + t_2 = 2$$

(see Example 4). The polyhedron $\overline{\mathsf{Pol}(\rho_R)}$ is defined by the following matrix system

$$\begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \\ 1 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 1 \\ -1 \\ 2 \\ 2 \\ -2 \end{pmatrix}.$$

Let us show that the matrix M is totally unimodular.

Definition 9. [NW88] *An integer matrix M is said* totally unimodular *if the determinant of all its square submatrices is equal to* 0, 1 *or* -1.

Lemma 2. *Any 3-block matrix is totally unimodular.*

Proof. We prove this lemma by induction on the size l of the square submatrices of M. The computation of their determinant is done with the cofactor method.

If $l = 1$ the result clearly holds. Suppose $l > 1$ and let $A \in \mathbb{Z}^{l\times l}$ be a submatrix of M. We have to prove that $\det(A)$ equals 0, 1 or -1. This proof is by induction on k the number of non null coefficients of the first column of A.

If $k = 0$, then $\det(A) = 0$. If $k = 1$, then we obtain the desired result by the induction hypothesis on l.

In order to treat the case $k > 1$, we need to introduce some notation and definition. As usual we denote by A_{ij} the coefficient of A located in row L_i and column C_j of A. We consider the rows L_i of A such that $A_{i1} \neq 0$,[10] and we define a total ordering on these rows as follows

$$L_i \subseteq L_{i'} \quad \text{iff} \quad \forall j \; A_{ij} \neq 0 \Rightarrow A_{i'j} \neq 0.$$

Consider two rows L_i, $L_{i'}$ such that $A_{i1} \neq 0$, $A_{i'1} \neq 0$ respectively, and $L_i \subseteq L_{i'}$. We build a new matrix B from A by replacing the row $L_{i'}$ by the row $L_{i'} - L_i$ if $A_{i1} = A_{i'1}$, and by the row $L_{i'} + L_i$ if $A_{i1} = -A_{i'1}$. The other rows are left unchanged. Since B is again 3-block, $\det(A) = \det(B)$, and B has $k-1$ non null coefficients in its first column, we can conclude that $\det(A)$ equals 0, 1 or -1 by the induction hypothesis on k. □

From the next theorem and Lemma 2, we have the following nice corollaries.

Theorem 2. [NW88] *Consider the polyhedron $\{t \in \mathbb{R}^m \mid M \cdot t \leq d\}$ with M a totally unimodular ($p \times m$) matrix and $d \in \mathbb{Z}^p$. Then the coordinates of its vertices are integers.*

[10] Recall that M is 3-block. Thus such a row L_i is formed by a block of 1's (resp. -1's) followed by a block of 0's.

Corollary 3. *The vertices of the polyhedron* $\overline{\mathsf{Pol}(\rho_R)}$ *have integer coordinates.*

Corollary 4. *The optimal cost* $\mathsf{OptCost}(\rho_R)$ *is an integer.*

In the next corollary, we indicate the relation between the optimal cost $\mathsf{OptCost}(r, r')$ of reaching the region r' from the region r and the optimal cost $\mathsf{OptCost}(\rho_R)$ associated with a path ρ_R of the region graph (see Definitions 7 and 8).

Corollary 5. *Let $\mathcal{A}$ be a timed automaton and r, r' be two regions of $R_\mathcal{A}$. Then*

$$\mathsf{OptCost}(r, r') = \inf\{\mathsf{OptCost}(\rho_R) \mid \rho_R = r \rightsquigarrow r' \text{ path in } R_\mathcal{A}\}.$$

Moreover, if $\mathsf{OptCost}(r, r') \neq \infty$, *then*

$$\mathsf{OptCost}(r, r') = \mathsf{OptCost}(\rho_R)$$

for some path $\rho_R = r \rightsquigarrow r'$ of $R_\mathcal{A}$, and $\mathsf{OptCost}(r, r')$ *is an integer.*

Proof. The first part of the corollary follows from the next equality.

$$\begin{aligned} &\inf\{\mathcal{C}(\rho) \mid \rho = q \rightsquigarrow q', q \in r, q' \in r'\} \\ =&\inf_{\rho_R} \inf\{\mathcal{C}(\rho) \mid \rho = q \rightsquigarrow q', [\rho] = \rho_R\}. \end{aligned}$$

The second part is an immediate consequence of Corollary 4. □

3.4 ε-Semantics

We have shown that Problem 2 is decidable : with the notation of Section 3.2, the optimal cost $\mathsf{OptCost}(\rho_R)$ can be obtained by computing the infimum value of the duration cost $\mathcal{C}_d(\rho(t_1, \ldots, t_m))$ under the set of constraints $\mathsf{Constr}(\rho_R)$. By the Simplex Method, it is obtained at one of the vertices of the polyhedron $\overline{\mathsf{Pol}(\rho_R)}$. Moreover, these vertices have integer coordinates by Corollary 3. All these results suggest that when computing $\mathsf{OptCost}(\rho_R)$, only time-transitions with a time τ "arbitrarily close to an integer" have to be considered (see also the end of Example 3). We thus introduce the ε*-semantics* in Definition 10 and we formalize the previous suggestion in Lemma 3.

The notion of ε-semantics of a timed automaton $\mathcal{A}$ is similar to the semantics given in Definition 2, except that elapse τ of time at a location is restricted to τ close to an integer.

Definition 10. *Let $\mathcal{A} = (L, X, E, \mathcal{I})$ be a timed automaton and $\varepsilon \in]0, \frac{1}{2}]$ be a real number. The ε*-transition system $T^\varepsilon_\mathcal{A} = (Q, \rightarrow^\varepsilon)$ *has the same set Q as in $T_\mathcal{A}$ and a* transition relation

$$\rightarrow^\varepsilon \;=\; \bigcup_{\tau \in \mathbb{R}^+_\varepsilon} \xrightarrow{\tau} \;\cup\; \bigcup_{e \in E} \xrightarrow{e}$$

such that $\mathbb{R}^+_\varepsilon = \{\tau \in \mathbb{R}^+ \mid \exists N \in \mathbb{N} \;\; |N - \tau| < \varepsilon\}$.

We distinguish two kinds of time-transition $\xrightarrow{\tau}$ with $\tau \in \mathbb{R}^+_\varepsilon$: either $0 \leq N - \tau < \varepsilon$, or $0 \leq \tau - N < \varepsilon$.[11] In the first case we use notation $\xrightarrow{N^-}$, and in the second case $\xrightarrow{N^+}$.

A finite path in the ε-transition system $T^\varepsilon_\mathcal{A}$ is called an ε*-run*; it is denoted by ρ^ε. Clearly any ε-run of $T^\varepsilon_\mathcal{A}$ can be seen as a run of $T_\mathcal{A}$.

Remark 13. When the context is clear enough, we use notation $q \rightarrow q'$ instead of $q \rightarrow^\varepsilon q'$ for transitions of $T^\varepsilon_\mathcal{A}$.

In the next lemma, we show that the optimal cost $\mathsf{OptCost}(\rho_R)$ can be approximated by the cost of some well-chosen ε-run.

[11] The two cases are mutually exclusive by the choice of $\varepsilon \in]0, \frac{1}{2}]$.

Lemma 3. *Let $\mathcal{A}$ be a weighted timed automaton, and ρ_R be a canonical initialized path in $R_\mathcal{A}$. Let $\varepsilon \in]0, \frac{1}{2}]$. Then there exists an initialized ε-run ρ^ε in $T^\varepsilon_\mathcal{A}$ such that*

$$[\rho^\varepsilon] = \rho_R \text{ and } |\mathsf{OptCost}(\rho_R) - \mathcal{C}(\rho^\varepsilon)| < \varepsilon.$$

Proof. We use the notation of Sect. 3.2. We suppose that ρ_R has the form

$$\rho_R = r'_0 \to r_1 \to r'_1 \to r_2 \cdots \to r_m \to r'_m$$

with the related parameterized run

$$\rho(t_1, t_2, \ldots, t_m) = q'_0 \xrightarrow{t_1} q_1 \xrightarrow{e_1} q'_1 \xrightarrow{t_2} q_2 \xrightarrow{e_2} \cdots \xrightarrow{t_m} q_m \xrightarrow{e_m} q'_m.$$

(see (4) and (6)). Consider the set of constraints $\mathsf{Constr}(\rho_R)$ and the polyhedron $\mathsf{Pol}(\rho_R)$ defined by them (see (9) and (11)).

By Remark 11, we know that computing the optimal cost $\mathsf{OptCost}(\rho_R)$ reduces in computing the optimal duration cost $\mathcal{C}_d$. By the Simplex Method and Corollary 3, this duration cost is obtained at one of the vertices $(\tau_1, ..., \tau_m) \in \mathbb{N}^m$ of $\overline{\mathsf{Pol}(\rho_R)}$ with integer coordinates.

Let us show how to define the required ε-run ρ^ε. Suppose $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{C})$ and let $K = \max_{l \in L} |\mathcal{C}(l)|$. Let ε' be such that $0 < \varepsilon' \le \varepsilon$ and $mK\varepsilon' < \varepsilon$. Since $\overline{\mathsf{Pol}(\rho_R)}$ is the closure of the polyhedron $\mathsf{Pol}(\rho_R)$, there exists a point $(\tau'_1, ..., \tau'_m) \in \mathsf{Pol}(\rho_R)$ such that $|\tau_k - \tau'_k| < \varepsilon'$ for all $k \in \{1, ..., m\}$. By Lemma 1, the run $\rho(\tau'_1, ..., \tau'_m)$ of $T_\mathcal{A}$ satisfies $[\rho(\tau'_1, ..., \tau'_m)] = \rho_R$. Moreover, since $\tau_k \in \mathbb{N}$, $\forall k$, and $\varepsilon' \le \varepsilon$, $\rho(\tau'_1, ..., \tau'_m)$ is an ε-run. Therefore we define $\rho^\varepsilon = \rho(\tau'_1, ..., \tau'_m)$. Looking at Definition 6 and Remark 11, we have

$$\Big|\mathsf{OptCost}(\rho_R) - \mathcal{C}\big(\rho^\varepsilon\big)\Big| = \Big|\sum_{k \in \{1,...,m\}} \mathcal{C}(l_k)\tau_k - \sum_{k \in \{1,...,m\}} \mathcal{C}(l_k)\tau'_k\Big| \le Km\varepsilon' < \varepsilon.$$

□

4 Solving the cost-optimal reachability problem

In this section, we solve the cost-optimal reachability problem for weighted timed automata (Problem 1) and we prove that it is PSPACE-COMPLETE as announced in Sect. 3. This proof needs several steps that we now briefly introduce. By Lemma 3, we have seen that to solve Problem 2 for a weighted timed automaton $\mathcal{A}$, it was sufficient to consider runs of the transition system $T_\mathcal{A}$ restricted to the ε-semantics (with ε arbitrarily close to 0). This observation motivates the introduction of the ε-region graph in Sect. 4.1, which is a refinement of the region graph $R_\mathcal{A}$. In Sect. 4.2, we establish what is the correspondence between runs of the ε-semantics and paths of the ε-region graph (Lemmas 4 and 5). In Section 4.3, we introduce the notion of discrete graph, a notion similar to the ε-region graph, which is independent of ε. We show how to augment the discrete graph with a weight function in relation to the cost function of $\mathcal{A}$. Then, we give the counterparts of the two previous lemmas with weight (Lemmas 8 and 9). All these steps lead to Theorem 3 where it is stated that solving Problem 1 reduces to compute some minimum weight in the discrete graph. The announced complexity of the cost-optimal reachability problem is proved in Sect. 4.4.

4.1 ε-Region graph

In this section, given a timed automaton $\mathcal{A}$, we define the concept of ε-region graph which can be seen as a refinement of $R_\mathcal{A}$. The refinement that we propose is simpler than the one given in [ALP01].

Let $\varepsilon \in]0, \frac{1}{2}]$. We define the ε-equivalence denoted $\approx^\varepsilon$ on clock valuations. This new equivalence relation refines the equivalence relation $\approx$ given in Definition 3. We recall that for every clock x_i, c_i is the largest constant such that x_i is compared with in any guard and any invariant of $\mathcal{A}$.

Definition 11. *Let $\varepsilon \in]0, \frac{1}{2}]$. Two clock valuations ν and ν' are ε-equivalent, $\nu \approx^\varepsilon \nu'$, iff they satisfy the following conditions*[12]

[12] With the choice of $\varepsilon \in]0, \frac{1}{2}]$, the last two conditions are mutually exclusive.

– *$\nu \approx \nu'$;*
– *$\bar{\nu}_i < \varepsilon$ iff $\bar{\nu}'_i < \varepsilon$ for all $i \in \{1, \ldots, n\}$ with $\nu_i \leq c_i$;*
– *$1 - \varepsilon < \bar{\nu}_i$ iff $1 - \varepsilon < \bar{\nu}'_i$ for all $i \in \{1, \ldots, n\}$ with $\nu_i \leq c_i$.*

Fig. 6 indicates the partition induced by the ε-equivalence for the timed automaton of Fig. 2.

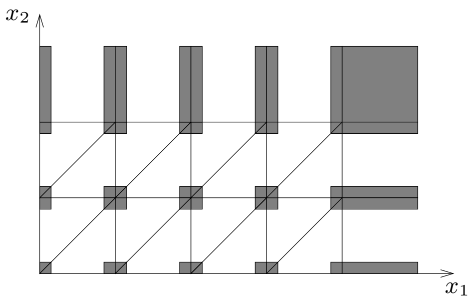

Fig. 6. The ε-equivalence $\approx^\varepsilon$

The relation $\approx^\varepsilon$ is extended to the states of $T_{\mathcal{A}}$ as done previously with $\approx$. An equivalence class is called an *ε-region*. The ε-region to which a state q belongs is denoted $[q]^\varepsilon$ and the set of all ε-regions is denoted by R^ε.

In order to define the ε-region graph of a timed automaton $\mathcal{A}$, we do not need all the ε-regions of R^ε (contrarily to the construction of $R_{\mathcal{A}}$). Due to Lemma 3, we only need to consider the ε-regions $[(l, \nu)]^\varepsilon$ whose clock values ν are close enough to n-tuples of integers (the dashed zones on Fig. 6).

Definition 12. *Given a timed automaton $\mathcal{A}$ and $\varepsilon \in]0, \frac{1}{2}]$, the set of* acceptable *ε-regions, denoted S^ε, is defined by*

$$S^\varepsilon = \big\{[(l, \nu)]^\varepsilon \mid \forall i \in \{1, \ldots, n\} : \nu_i \leq c_i \Rightarrow (\bar{\nu}_i < \varepsilon \text{ or } 1 - \varepsilon < \bar{\nu}_i)\big\}.$$

Remark 14. If $r^\varepsilon = [(l, \nu)]^\varepsilon$ is an ε-region of S^ε, then there exists a unique region $r \in R$, equal to $[(l, \nu)]$, such that $r^\varepsilon \subseteq r$. In the sequel, r^ε always denotes an ε-region included in the region r.[13]

Remark 15. Using the representation introduced in Remark 5, we can visualize an ε-region r^ε as on Fig. 7 (when r is a bounded region). We observe that the fractional parts $\bar{\nu}_i$ of the clock values are either less than ε or greater than $1 - \varepsilon$. We thus introduce the following notation[14]

$$\mathsf{Low}(r^\varepsilon) = \{x_i \mid \nu_i \leq c_i \text{ and } \bar{\nu}_i < \varepsilon\};$$
$$\mathsf{High}(r^\varepsilon) = \{x_i \mid \nu_i \leq c_i \text{ and } 1 - \varepsilon < \bar{\nu}_i\}.$$

This graphical representation of the ε-regions is very helpful in the proofs below.

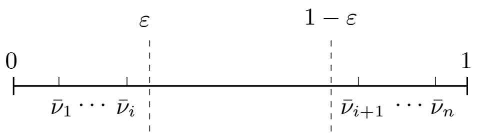

Fig. 7. Representation of the region $0 < \bar{\nu}_1 < \cdots < \bar{\nu}_i < \varepsilon \leq 1 - \varepsilon < \bar{\nu}_{i+1} < \cdots < \bar{\nu}_n$

[13] Similarly if $\delta \leq \varepsilon$, we will also use notation $r^\delta, r^\varepsilon, r$ with $r^\delta \subseteq r^\varepsilon \subseteq r$.
[14] Notice that the sets $\mathsf{Low}(r^\varepsilon)$ and $\mathsf{High}(r^\varepsilon)$ are disjoint since $\varepsilon \leq \frac{1}{2}$.

Remark 16. The acceptable ε-regions that we propose as a refinement of the classical regions of [AD94] are simpler than the refinement introduced in [ALP01]. Indeed in our case, the clock values of an acceptable ε-region r^ε are arbitrarily close to one of the corners of the region r, when in [ALP01] the clock values are arbitrarily close to one of the boundaries of r.

In the next example, we illustrate the interest of Definition 12 for computing the optimal cost $\mathsf{OtpCost}(r, r')$ for two regions r, r' of a timed automaton.

Example 6. We consider the weighted timed automaton $\mathcal{A}$ of Fig. 8. The cost of each location is indicated on the figure

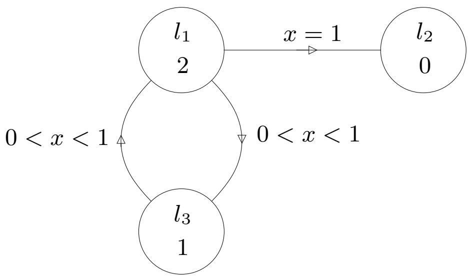

Fig. 8. A weighted timed automaton

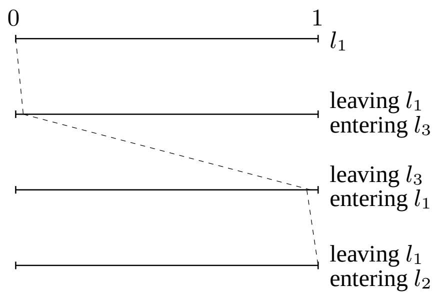

Fig. 9. The run $\rho_2(\varepsilon, 1 - 2 \cdot \varepsilon, \varepsilon)$

and the cost of each edge is null. The invariant $(x \leq 1)$ is assigned to each location, showing that $\mathcal{A}$ is bounded. We want to compute the optimal cost $\mathsf{OptCost}(r, r')$ for the two regions $r = [(l_1, 0)]$ and $r' = [(l_2, 1)]$ of $R_\mathcal{A}$.

Let $\rho_1 = (l_1, 0) \rightsquigarrow (l_2, 1)$ be a run of $T_\mathcal{A}$ not going through location l_3. Clearly it has a cost $\mathcal{C}(\rho_1) = 2$.

We now consider runs $\rho_2 = (l_1, 0) \rightsquigarrow (l_2, 1)$ going through l_3. This family of runs can be described by the parameterized run

$$\rho_2(t_1, t_2, t_3) = (l_1, 0) \xrightarrow{t_1} (l_1, t_1) \rightarrow (l_3, t_1) \xrightarrow{t_2} (l_3, t_1 + t_2)$$
$$\rightarrow (l_1, t_1 + t_2) \xrightarrow{t_3} (l_1, t_1 + t_2 + t_3) \rightarrow (l_2, t_1 + t_2 + t_3)$$

where t_1, t_2 and t_3 are constrained by

$$0 < t_1 < 1,\ 0 < t_1 + t_2 < 1 \text{ and } t_1 + t_2 + t_3 = 1. \tag{13}$$

The cost of the parameterized run $\rho_2(t_1, t_2, t_3)$ is given by $2 \cdot t_1 + t_2 + 2 \cdot t_3$. One can check that the infimum value of $2 \cdot t_1 + t_2 + 2 \cdot t_3$ under the constraints (13) is equal to 1, and that it is obtained at the point $(t_1, t_2, t_3) = (0, 1, 0)$.

Therefore, the optimal cost $\mathsf{OptCost}(r, r')$ is equal to 1.

We now study in more details the parameterized run $\rho_2(t_1, t_2, t_3)$ with (t_1, t_2, t_3) arbitrarily close to $(0, 1, 0)$. Let us fix $\varepsilon \in]0, \frac{1}{2}]$. Given $0 < \delta < \varepsilon$, the run $\rho_2(\delta, 1 - 2 \cdot \delta, \delta)$ respects the constraints given in (13). This run is depicted on Fig. 9. Notice on this figure how it was necessary to refine the region $(l_3, 0 < x < 1)$ into the two acceptable ε-regions $(l_3, 0 < x < \varepsilon)$ and $(l_3, 1 - \varepsilon < x < 1)$.

Given $\mathcal{A}$ a timed automaton and $\varepsilon \in]0, \frac{1}{2}]$, we now define the ε-region graph $R^\varepsilon_\mathcal{A}$. It is obtained in two steps: first we define the quotient graph $T_\mathcal{A}/_{\approx^\varepsilon}$, and then we restrict it to the set S^ε of acceptable ε-regions.

Definition 13. *Let $\mathcal{A}$ be a timed automaton and $\varepsilon \in]0, \frac{1}{2}]$. The* ε-region graph $R^\varepsilon_\mathcal{A} = (S^\varepsilon, \rightarrow)$ *is the finite subgraph of $T_\mathcal{A}/_{\approx^\varepsilon}$ induced by S^ε. Its vertex set is equal to S^ε. Its edge set is composed of the edges $r^\varepsilon \rightarrow r'^\varepsilon$, with $r^\varepsilon, r'^\varepsilon \in S^\varepsilon$, such that there exist two states $q \in r^\varepsilon$, $q' \in r'^\varepsilon$, and a transition $q \rightarrow q'$ in $T_\mathcal{A}$. The edge $r^\varepsilon \rightarrow r'^\varepsilon$ is called a* switch-edge *(resp.* time-edge*) if $q \rightarrow q'$ is a switch-transition (resp. time-transition).*

A path in $R^\varepsilon_\mathcal{A}$ is denoted ρ_{S^ε}. As for $R_\mathcal{A}$, the vertices of such a path are all bounded regions (see Remark 2). We say that a path $\rho_{S^\varepsilon} = r^\varepsilon \rightsquigarrow r'^\varepsilon$ in $R^\varepsilon_\mathcal{A}$ is *initialized* if r^ε is of the form $[(l, 0)]^\varepsilon$ such that all the clock values are null. Let us notice that we only consider finite paths of $R^\varepsilon_\mathcal{A}$ in this paper.

Remark 17. In the sequel, we only work with the ε-regions that are acceptable. Therefore, we omit the term "acceptable".

Remark 18. The size $|S^\varepsilon|$ is bounded by $(n+1)|R|$.[15] Indeed a region r of $R_\mathcal{A}$ gives rise to at most $n+1$ different ε-regions $r^\varepsilon \subseteq r$, since each such r^ε is specified by the way the interval $[0,1[$ is cut into the sets $\mathsf{Low}(r^\varepsilon)$ and $\mathsf{High}(r^\varepsilon)$ (see Fig. 7). Since $|R_\mathcal{A}|$ is in $\mathcal{O}(2^{|\mathcal{A}|})$, it follows that $|R^\varepsilon_\mathcal{A}|$ is also in $\mathcal{O}(2^{|\mathcal{A}|})$.

4.2 Links between $T^\varepsilon_\mathcal{A}$ and $R^\varepsilon_\mathcal{A}$

In this section, given a timed automaton $\mathcal{A}$ and $\varepsilon \in]0, \frac{1}{2}]$, we show how the runs of the ε-transition system $T^\varepsilon_\mathcal{A}$ are linked to the paths of the ε-region graph $R^\varepsilon_\mathcal{A}$, and conversely (Lemmas 4 and 5).

First, it is important to notice that there are no natural simulation of $\mathcal{R}^\varepsilon_\mathcal{A}$ by $T^\varepsilon_\mathcal{A}$ (Example 7) and no natural simulation of $T^\varepsilon_\mathcal{A}$ by $\mathcal{R}^\varepsilon_\mathcal{A}$ (Example 8).

Example 7. Let $\mathcal{A}$ be a timed automaton with one location l and two clocks x_1, x_2. Let us fix $\varepsilon \in]0, \frac{1}{2}]$ and $\delta = \frac{\varepsilon}{10}$. Let r^ε be the ε-region such that $0 < x_2 < x_1 < 1$, $\mathsf{Low}(r^\varepsilon) = \{x_2\}$ and $\mathsf{High}(r^\varepsilon) = \{x_1\}$, and let r'^ε be the ε-region such that $0 < x_2 < x_1 = 1$, $\mathsf{Low}(r'^\varepsilon) = \{x_1, x_2\}$ and $\mathsf{High}(r'^\varepsilon) = \varnothing$. We clearly have $r^\varepsilon \to r'^\varepsilon$. However given $(l,\nu) = (l, 1-\varepsilon+\delta, \varepsilon-\delta) \in r^\varepsilon$, it is impossible to find (l,ν') such that $(l,\nu) \to (l,\nu')$ and $(l,\nu') \in r'^\varepsilon$. This situation is illustrated on Fig. 10.

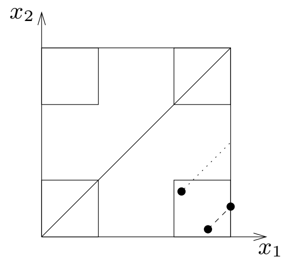

Fig. 10. No natural simulation of $\mathcal{R}^\varepsilon_\mathcal{A}$ by $T^\varepsilon_\mathcal{A}$

Example 8. Let $\mathcal{A}$ be a timed automaton with one location l and one clock x_1. Let us fix $\varepsilon \in]0, \frac{1}{3}]$ and $\delta = \frac{\varepsilon}{10}$. We consider the transition $(l,\nu) \xrightarrow{2\delta} (l,\nu')$ of $T^\varepsilon_\mathcal{A}$ such that $\nu = \varepsilon - \delta$ and $\nu' = \varepsilon + \delta$. Clearly (l,ν) belongs to some ε-region r^ε of $R^\varepsilon_\mathcal{A}$ since $\nu < \varepsilon$. However, it is impossible to find an ε-region r'^ε such that $(l,\nu') \in r'^\varepsilon$ because we have neither $\nu' < \varepsilon$ nor $1-\varepsilon < \nu'$ by the choice of ε and δ. (See Definition 12).

Although there is no natural simulation between $\mathcal{R}^\varepsilon_\mathcal{A}$ and $T^\varepsilon_\mathcal{A}$, we are able to relate runs of $T^\varepsilon_\mathcal{A}$ and paths of $R^\varepsilon_\mathcal{A}$ in a weaker way. This relation is described in the next two lemmas.

We recall that the timed automata of this paper are bounded (see Remark 2). Therefore, the regions and ε-regions considered in these lemmas are supposed to be bounded.

Lemma 4. *Let $\mathcal{A}$ be a timed automaton and $\varepsilon \in]0, \frac{1}{3}]$. Let $\rho_{S^\varepsilon} = r^\varepsilon_0 \to r^\varepsilon_1 \to \cdots \to r^\varepsilon_m$ be an initialized path in $R^\varepsilon_\mathcal{A}$. Then there exists an initialized ε-run $\rho^\varepsilon = (l_0,\nu^0) \to (l_1,\nu^1) \to \cdots \to (l_m,\nu^m)$ in $T^\varepsilon_\mathcal{A}$ such that $(l_k,\nu^k) \in r^\varepsilon_k$ for all $k \in \{0,\ldots,m\}$.*

In the statement of this lemma, the same ε number is used in both the path ρ_{S^ε} in $R^\varepsilon_\mathcal{A}$ and the ε-run ρ^ε in $T^\varepsilon_\mathcal{A}$.

The proof proceeds by induction on the length of ρ_{S^ε} and therefore it requires the use of $m+1$ intermediate $\varepsilon_k \leq \varepsilon$, $k \in \{0,\ldots,m\}$, of the form $\varepsilon_k = \frac{\varepsilon}{2^{m-k}}$. For instance, to avoid the situation of Example 7, one takes (l,ν) in $r^{\frac{\varepsilon}{2}}$ (instead of r^ε) to obtain a transition $(l,\nu) \to (l,\nu')$ with $(l,\nu') \in r'^\varepsilon$ (see Fig. 10). The proof is technical since several cases have to be considered, however it is not difficult. It can be skipped at a first reading.

[15] We recall that n is the number of clocks.

Proof (of Lemma 4). We are going to build the required ε-run ρ^ε as follows : for all $k \in \{0,\ldots,m\}$, we will insure that $(l_k,\nu^k) \in r_k^{\varepsilon_k}$ and the prefix

$$\rho^{\varepsilon_k} = (l_0,\nu^0)\to(l_1,\nu^1)\cdots\to(l_k,\nu^k) \tag{14}$$

is a run in $T_{\mathcal{A}}^{\varepsilon_k}$, with $\varepsilon_k = \frac{\varepsilon}{2^{m-k}}$. Since $\varepsilon_k \le \varepsilon$, we have $r_k^{\varepsilon_k} \subseteq r_k^{\varepsilon}$ and ρ^{ε_k} is an ε-run of $T_{\mathcal{A}}^{\varepsilon}$.[16] Thus the thesis holds with $k = m$.

We proceed by induction on k. Suppose $k = 0$. Since ρ_{S^ε} is initial, r_0^ε has the form $[(l_0,0)]$. The unique state of ρ^{ε_0} is thus $(l_0,0)$.

Let $k \ge 0$ and suppose by the induction hypothesis that we have built a path ρ^{ε_k} like in (14) with the desired conditions. Since ρ^{ε_k} is also an ε_{k+1}-run in $T_{\mathcal{A}}^{\varepsilon_{k+1}}$, we have to show that we can find a transition $(l_k,\nu^k)\to(l_{k+1},\nu^{k+1})$ in $T_{\mathcal{A}}^{\varepsilon_{k+1}}$ such that $(l_{k+1},\nu^{k+1}) \in r_{k+1}^{\varepsilon_{k+1}}$.

To make the sequel more readable, we change the notation as follows. We denote the state (l_k,ν^k) by (l,ν) and the state (l_{k+1},ν^{k+1}) by (l',ν'). Similarly notation r^ε and r'^ε is used instead of r_k^ε and r_{k+1}^ε respectively; notation r^{ε_k} and $r'^{\varepsilon_{k+1}}$ is used instead of $r_k^{\varepsilon_k}$ and $r_{k+1}^{\varepsilon_{k+1}}$ respectively. We denote by r (resp. r') the region of $R_{\mathcal{A}}$ which contains r^{ε_k} (resp. $r'^{\varepsilon_{k+1}}$).

We now consider the two possible cases, switch-edge and time-edge, for the edge $r^\varepsilon \to r'^\varepsilon$ in $R_{\mathcal{A}}^\varepsilon$. In each case we define the adequate state (l',ν').

Suppose that $r^\varepsilon \to r'^\varepsilon$ is a switch-edge. By the induction hypothesis, $(l,\nu) \in r^{\varepsilon_k}$. Since $r \to r'$ is a switch-edge in $R_{\mathcal{A}}$, there exists a switch-transition $(l,\nu) \xrightarrow{e} (l',\nu')$ in $T_{\mathcal{A}}$. Since $\varepsilon_k < \varepsilon_{k+1}$ and $(l,\nu) \in r^{\varepsilon_k}$, this transition is also a switch-transition in $T_{\mathcal{A}}^{\varepsilon_{k+1}}$ such that $(l',\nu') \in r^{\varepsilon_{k+1}}$.

We now treat the case where $r^\varepsilon \to r'^\varepsilon$ is a time-edge. First we suppose that r' is a closed region. It follows that there exists a unique $(l,\nu') \in r'$ (and a unique τ) such that $(l,\nu) \xrightarrow{\tau} (l,\nu')$ is a time-transition in $T_{\mathcal{A}}$. Hence there exists a clock x_f whose fractional part is equal to 0 in r' (*i.e.* $\bar\nu'_f = 0$). We are going to prove that $(l,\nu) \xrightarrow{\tau} (l,\nu')$ is a transition in $T_{\mathcal{A}}^{\varepsilon_{k+1}}$ such that $(l,\nu') \in r'^{\varepsilon_{k+1}}$. In other words we show that $|N-\tau| < \varepsilon_{k+1}$ for some $N \in \mathbb{N}$ (see Definition 10) and $\bar\nu'_i \in [0,\varepsilon_{k+1}[\cup]1-\varepsilon_{k+1},1[$ for each $i \in \{1,\ldots,n\}$ (see Definition 12). We have to distinguish four cases depending on the belonging of x_f and x_i to $\mathsf{Low}(r^{\varepsilon_k})$ or $\mathsf{High}(r^{\varepsilon_k})$.

1. If $x_f \in \mathsf{Low}(r^{\varepsilon_k})$, then $\tau = N - \bar\nu_f$ for some $N \in \mathbb{N}\setminus\{0\}$ (this case is illustrated on Fig. 11). Thus $N-\tau = \bar\nu_f < \varepsilon_k$, showing that $|N-\tau| < \varepsilon_{k+1}$. We distinguish two subcases.

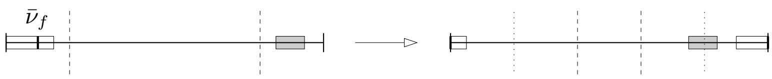

Fig. 11. The proof at a glance when $x_f \in \mathsf{Low}(r^{\varepsilon_k})$

   (a) $x_i \in \mathsf{Low}(r^{\varepsilon_k})$.
   If $\bar\nu_i \ge \bar\nu_f$, then $\bar\nu'_i = \bar\nu_i - \bar\nu_f$. Hence by induction hypothesis we have $0 \le \bar\nu'_i < \varepsilon_k < \varepsilon_{k+1}$. If $\bar\nu_i < \bar\nu_f$, then $\bar\nu'_i = 1 - (\bar\nu_f - \bar\nu_i)$. Hence by induction hypothesis we have $1-\varepsilon_{k+1} < 1-\varepsilon_k < \bar\nu'_i < 1$.
   (b) $x_i \in \mathsf{High}(r^{\varepsilon_k})$.
   We have $\bar\nu'_i = \bar\nu_i - \bar\nu_f$. By induction hypothesis, we conclude that $1-\varepsilon_{k+1} = 1-2\varepsilon_k < \bar\nu'_i < 1$.
2. If $x_f \in \mathsf{High}(r^{\varepsilon_k})$, then $\tau = N - \bar\nu_f$ for some $N \in \mathbb{N}\setminus\{0\}$ (this case is illustrated on Fig. 12). Thus $\tau - N + 1 < \varepsilon_k$, showing that $|\tau - N + 1| < \varepsilon_{k+1}$. We distinguish two subcases.
   (a) $x_i \in \mathsf{Low}(r^{\varepsilon_k})$.
   We have $\bar\nu'_i = \bar\nu_i + (1-\bar\nu_f)$. Hence $0 \le \bar\nu'_i < 2\varepsilon_k = \varepsilon_{k+1}$.
   (b) $x_i \in \mathsf{High}(r^{\varepsilon_k})$.
   If $\bar\nu_i < \bar\nu_f$, then $\bar\nu'_i = \bar\nu_i + (1-\bar\nu_f)$ and $1-\varepsilon_{k+1} < 1-\varepsilon_k < \bar\nu_i{}' < 1$. If $\bar\nu_i \ge \bar\nu_f$, then $\bar\nu'_i = \bar\nu_i - \bar\nu_f$ and $0 \le \bar\nu'_i < \varepsilon_k < \varepsilon_{k+1}$.

[16] Notice that we use the notation $r_k^{\varepsilon_k}$, r_k^ε as proposed in Remark 14.

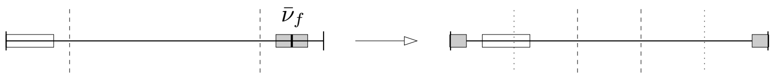

Fig. 12. The proof at a glance when $x_f \in \mathsf{High}(r^{\varepsilon_k})$

This concludes the case where $r^\varepsilon \to r'^\varepsilon$ is a time-edge such that r' is a closed region.[17]

We now treat the case where $r^\varepsilon \to r'^\varepsilon$ is a time-edge such that r' is an open region. We have to define τ and (l, ν') such that $(l, \nu) \xrightarrow{\tau} (l, \nu')$ is a time-transition of $T_{\mathcal{A}}^{\varepsilon_{k+1}}$ and $(l, \nu') \in r'^{\varepsilon_{k+1}}$. We begin to introduce additional notation (see Fig. 13). Among the clocks which belong to $\mathsf{Low}(r^{\varepsilon_k})$, we denote by x_a (resp. x_b) the one whose valuation has the smallest (resp. largest) fractional part. Similarly for the clocks of $\mathsf{High}(r^{\varepsilon_k})$, x_c (resp. x_d) is the one whose valuation has the smallest (resp. largest) fractional part.

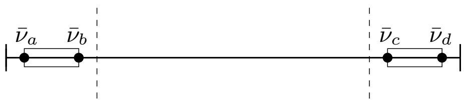

Fig. 13. Additional notation

Since the region r' is supposed to be open, either there exists a closed region r'' such that $r \to r'' \to r'$ (with possibly $r = r''$) such that $r' = \mathrm{succ}(r'')$, or such a closed region r'' does not exist, and then $r = r'$.

1. If r'' exists, by using the previous case, we can find $(l, \nu'') \in r''$ such that $(l, \nu) \xrightarrow{\tau'} (l, \nu'')$ is a transition in $T_{\mathcal{A}}^{\varepsilon_{k+1}}$, $|N - \tau| < \varepsilon_{k+1}$ for some $N \in \mathbb{N}$, and $(l, \nu'') \in r''^{\varepsilon_{k+1}}$. We then choose τ'' such that $\tau'' < \min(\varepsilon_{k+1} - \bar{\nu}''_b, 1 - \bar{\nu}''_d)$ and $|N - (\tau' + \tau'')| < \varepsilon_{k+1}$ (see Fig. 14). We define (l, ν') such that $(l, \nu'') \xrightarrow{\tau''} (l, \nu')$. With $\tau = \tau' + \tau''$, it follows that $(l, \nu) \xrightarrow{\tau} (l, \nu')$ is a transition in $T_{\mathcal{A}}^{\varepsilon_{k+1}}$ such that $(l, \nu') \in r'^{\varepsilon_{k+1}}$.

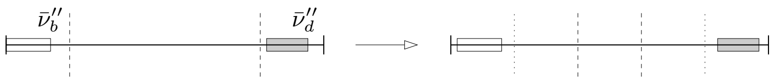

Fig. 14. The proof at a glance for transition $r'' \to r'$

2. If r'' does not exist, then $r = r'$. In the case $r^\varepsilon = r'^\varepsilon$, we proceed with an argument similar to the one of the previous case. Indeed it suffices to take $\tau < \min(\varepsilon_k - \bar{\nu}_b, 1 - \bar{\nu}_d)$. With $N = 0$, we have $|N - \tau| < \varepsilon_k < \varepsilon_{k+1}$. In the case $r^\varepsilon \neq r'^\varepsilon$, let us show that

$$\mathsf{Low}(r^\varepsilon) = \mathsf{High}(r'^\varepsilon), \text{ and } \mathsf{High}(r^\varepsilon) = \mathsf{Low}(r'^\varepsilon) = \varnothing. \tag{15}$$

   The hypothesis $\varepsilon \leq \frac{1}{3}$ will be necessary. Assume that (15) does not hold. Let us study in more detail the transition $r^\varepsilon \to r'^\varepsilon$ in the light of Definition 13. The situation $\mathsf{Low}(r^\varepsilon) = \varnothing$ and $\mathsf{High}(r^\varepsilon) \neq \varnothing$ is impossible. Therefore $\mathsf{Low}(r^\varepsilon)$ and $\mathsf{High}(r^\varepsilon)$ are both non empty. Consider a time-transition $(l, \tilde{\nu}) \xrightarrow{\tilde{\tau}} (l, \tilde{\nu}')$ of $T_{\mathcal{A}}$ such that $(l, \tilde{\nu}) \in r^\varepsilon$ and $(l, \tilde{\nu}') \in r'^\varepsilon$. Since $r^\varepsilon \neq r'^\varepsilon$, we must have *(i)* $\tilde{\nu}_b < \varepsilon$, $\tilde{\nu}_b + \tilde{\tau} > 1 - \varepsilon$, and *(ii)* $\tilde{\nu}_d > 1 - \varepsilon$, $\tilde{\nu}_d + \tilde{\tau} < 1$ (see Fig. 15). It follows that $1 - \varepsilon < \varepsilon + \tilde{\tau}$ in case *(i)*, and $1 - \varepsilon + \tilde{\tau} < 1$ in case *(ii)*. This is impossible because $\varepsilon \leq \frac{1}{3}$. Since (15) holds, we choose $\tau = 1 - \varepsilon_k$. This case is illustrated on Fig. 16. The transition $(l, \nu) \xrightarrow{\tau} (l, \nu')$ is

[17] The hypothesis that r'^ε is an ε-region is of no importance in the arguments given in this case.

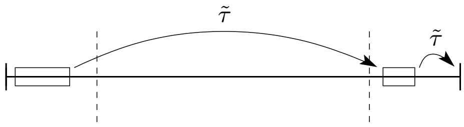

Fig. 15. An impossible situation

thus a time-transition of $T_{\mathcal{A}}^{\varepsilon_{k+1}}$. It remains to show that $(l, \nu') \in r'^{\varepsilon_{k+1}}$, that is, $1 - \varepsilon_{k+1} < \bar{\nu}'_c$ and $\bar{\nu}'_d < 1$. We have $\bar{\nu}'_c = \bar{\nu}_a + 1 - \varepsilon_k > 1 - \varepsilon_k > 1 - \varepsilon_{k+1}$, showing the first inequality. To obtain the second one, notice that $\bar{\nu}'_d = \bar{\nu}_b + 1 - \varepsilon_k < \varepsilon_k + 1 - \varepsilon_k = 1$.

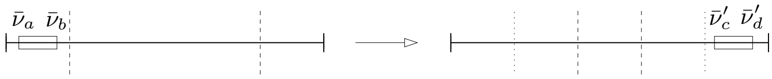

Fig. 16. The proof at a glance when $\mathsf{Low}(r^{\varepsilon}) = \mathsf{High}(r'^{\varepsilon})$, and $\mathsf{High}(r^{\varepsilon}) = \mathsf{Low}(r'^{\varepsilon}) = \varnothing$

□

Lemma 5. *Let $\mathcal{A}$ be a timed automaton. Let $\rho^{\delta} = (l_0, \nu^0) \to (l_1, \nu^1) \to \cdots \to (l_m, \nu^m)$ be an initialized δ-run in $T_{\mathcal{A}}^{\delta}$, with $\delta \in]0, \frac{1}{2(m+1)}]$. Then, with $\varepsilon = (m+1)\delta$, there exists a path $\rho_{S^{\varepsilon}} = r_0^{\varepsilon} \to r_1^{\varepsilon} \to \cdots \to r_m^{\varepsilon}$ in $R_{\mathcal{A}}^{\varepsilon}$ such that $(l_k, \nu^k) \in r_k^{\varepsilon}$ for all $k \in \{0, \ldots, m\}$.*

Contrary to Lemma 4 where the same ε number was used, the statement of this lemma requires the use of different numbers ε and δ. This is necessary to avoid the situation of Example 8. Again the proof of this lemma is technical, but not difficult. It can be skipped at a first reading.

Proof (of Lemma 5). Consider the regions $r_k = [(l_k, \nu^k)]$ of $R_{\mathcal{A}}$, for $k \in \{0, \ldots, m\}$. We are going to build the required path $\rho_{S^{\varepsilon}}$ as follows : for all $k \in \{0, \ldots, m\}$, we have $(l_k, \nu^k) \in r_k^{\varepsilon_k}$ and the prefix

$$\rho_{S^{\varepsilon_k}} = r_0^{\varepsilon_k} \to r_1^{\varepsilon_k} \to \cdots \to r_k^{\varepsilon_k}$$

is a path in $R_{\mathcal{A}}^{\varepsilon_k}$, with $\varepsilon_k = (k+1)\delta$.[18] Since $\varepsilon_k \leq \varepsilon$, we have $r_k^{\varepsilon_k} \subseteq r_k^{\varepsilon}$ and $\rho_{S^{\varepsilon_k}}$ is also a path in $R_{\mathcal{A}}^{\varepsilon}$. Thus the thesis holds with $k = m$.

We proceed by induction on k. If $k = 0$, then $(l_0, \nu^0) \in r_0^{\varepsilon_0}$ since $\nu^0 = 0$.

Let $k \geq 0$. Suppose by induction hypothesis that we have built the path $\rho_{S^{\varepsilon_k}}$ with the desired conditions. This path can be seen as a path in $R_{\mathcal{A}}^{\varepsilon_{k+1}}$ since $r_j^{\varepsilon_k} \subseteq r_j^{\varepsilon_{k+1}}$ for all $j \in \{0, \ldots, k\}$. Consider the edge $r_k \to r_{k+1}$ of $R_{\mathcal{A}}$. If we show that $(l_{k+1}, \nu^{k+1}) \in r_{k+1}^{\varepsilon_{k+1}}$, then $r_k^{\varepsilon_{k+1}} \to r_{k+1}^{\varepsilon_{k+1}}$ is an edge of $R_{\mathcal{A}}^{\varepsilon_{k+1}}$, and case $k+1$ is thus solved.

As in the proof of Lemma 4, we change the notation as follows. We denote the states $(l_k, \nu^k), (l_{k+1}, \nu^{k+1})$ by $(l, \nu), (l', \nu')$ respectively, and the regions r_k, r_{k+1} by r, r' respectively. In a way to prove that $(l', \nu') \in r'^{\varepsilon_{k+1}}$, we treat the different types of transition $(l, \nu) \to (l', \nu')$ (see Definition 10).

Suppose that $(l, \nu) \to (l', \nu')$ is a switch-transition. Since $(l, \nu) \in r^{\varepsilon_k}$ by induction hypothesis and $\varepsilon_k < \varepsilon_{k+1}$, then $(l', \nu') \in r'^{\varepsilon_k} \subseteq r'^{\varepsilon_{k+1}}$.

Suppose now that $(l, \nu) \xrightarrow{\tau} (l', \nu')$ is a time-transition such that $|N - \tau| < \delta$ for some $N \in \mathbb{N}$. We have to consider the two cases $(l, \nu) \xrightarrow{N^+} (l, \nu')$ and $(l, \nu) \xrightarrow{N^-} (l, \nu')$.

1. Suppose $\tau = N + \tau'$ with $0 \leq \tau' < \delta$. This case is illustrated on Fig. 17. We have to prove that $(l, \nu') \in r'^{\varepsilon_{k+1}}$, i.e. $\bar{\nu}'_i \in [0, \varepsilon_{k+1}[\cup]1 - \varepsilon_{k+1}, 1[$ for all $i \in \{1, \ldots, n\}$. A clock x_i belongs either to $\mathsf{Low}(r_k^{\varepsilon_k})$ or to $\mathsf{High}(r_k^{\varepsilon_k})$.

[18] As in the proof of the previous lemma, we use the notation discussed in Remark 14. On the other hand, notice that $\varepsilon \in]0, \frac{1}{2}]$ by the choice of δ.

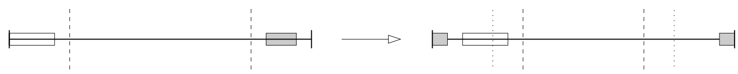

Fig. 17. The proof at a glance for transition $(l, \nu) \xrightarrow{N^+} (l, \nu')$

(a) $x_i \in \mathsf{Low}(r_k^{\varepsilon_k})$. Thus by induction hypothesis, $0 \le \bar{\nu}'_i = \bar{\nu}_i + \tau' < \varepsilon_k + \delta = \varepsilon_{k+1}$.
(b) $x_i \in \mathsf{High}(r_k^{\varepsilon_k})$. Then either $\bar{\nu}'_i = \bar{\nu}_i + \tau'$ or $\bar{\nu}'_i = \bar{\nu}_i + \tau' - 1$. In the first case, we have $1 - \varepsilon_{k+1} < 1 - \varepsilon_k < \bar{\nu}_i \le \bar{\nu}'_i < 1$. In the second case, we have $0 \le \bar{\nu}'_i < \delta < \varepsilon_{k+1}$.

2. Suppose that $\tau = N - \tau'$ with $0 < \tau' < \delta$. This case is illustrated on Fig. 18. Let us show that $\bar{\nu}'_i \in [0, \varepsilon_{k+1}[\cup]1 - \varepsilon_{k+1}, 1[$ for all $i \in \{1, \ldots, n\}$.

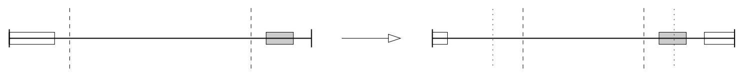

Fig. 18. The proof at a glance for transition $(l, \nu) \xrightarrow{N^-} (l, \nu')$

(a) $x_i \in \mathsf{Low}(r_k^{\varepsilon_k})$. Then either $\bar{\nu}'_i = \bar{\nu}_i - \tau'$, or $\bar{\nu}'_i = \bar{\nu}_i - \tau' + 1$. In the first case, we have $0 \le \bar{\nu}'_i \le \bar{\nu}_i < \varepsilon_k < \varepsilon_{k+1}$. In the second case, we have $1 - \varepsilon_{k+1} < 1 - \varepsilon_k < \bar{\nu}_i < \bar{\nu}'_i < 1$.
(b) $x_i \in \mathsf{High}(r_k^{\varepsilon_k})$. Therefore $\bar{\nu}'_i = \bar{\nu}_i - \tau'$ and $1 - \varepsilon_{k+1} = 1 - \varepsilon_k - \delta < \bar{\nu}'_i < 1$.

□

4.3 Weighted discrete graph

In the previous subsection, we gave the relation between the ε-semantics and the ε-region graph of a timed automaton $\mathcal{A}$. In this section, we introduce the notion of discrete graph, a notion similar to the ε-region graph, which is independent of ε (Definition 14). Then, we consider $\mathcal{A}$ as a weighted timed automaton with a cost function $\mathcal{C}$. We show how the discrete graph can be augmented with a weight function $\mathcal{W}$ in relation to $\mathcal{C}$ (Definition 15). We end the section with an important result that indicates how the optimal cost $\mathsf{OptCost}(r, r')$, with r, r' being two regions of $R_{\mathcal{A}}$, can be computed thanks to the weighted discrete graph (Theorem 3).

In [BBL04], Bouyer et al. propose the construction of a graph called the corner point abstraction, for studying the optimal way of staying into a designated set of safe locations. This construction shares several ideas with the construction proposed here for the weighted discrete graph.

Let $\mathcal{A}$ be a timed automaton. We begin with a lemma that states that all the ε-region graphs $R^{\varepsilon}_{\mathcal{A}}$ are isomorphic. The proof is in the same vein as for Lemma 4.

Lemma 6. *Let $\mathcal{A}$ be a timed automaton. Then all the ε-region graphs $R^{\varepsilon}_{\mathcal{A}}$, with $\varepsilon \in]0, \frac{1}{3}]$, are isomorphic graphs.*

Proof. Consider $R^{\delta}_{\mathcal{A}} = (S^{\delta}, \rightarrow)$ and $R^{\varepsilon}_{\mathcal{A}} = (S^{\varepsilon}, \rightarrow)$, with $\delta, \varepsilon \in]0, \frac{1}{3}]$ such that $\delta < \varepsilon$. We have to prove that $R^{\delta}_{\mathcal{A}}$ and $R^{\varepsilon}_{\mathcal{A}}$ are isomorphic graphs, that is, there exists a one-to-one correspondence between S^{δ} and S^{ε} that respects the edge relation $\rightarrow$ of each graph.

For any δ-region r^{δ} of $R^{\delta}_{\mathcal{A}}$, since $\delta < \varepsilon$, there exists exactly one ε-region r^{ε} of $R^{\varepsilon}_{\mathcal{A}}$ such that $r^{\delta} \subseteq r^{\varepsilon}$.[19] This establishes the one-to-one correspondence between S^{δ} and S^{ε}. Of course we have $\mathsf{Low}(r^{\varepsilon}) = \mathsf{Low}(r^{\delta})$ and $\mathsf{High}(r^{\varepsilon}) = \mathsf{Low}(r^{\delta})$.

If $r^{\delta} \rightarrow r'^{\delta}$ is an edge in $R^{\delta}_{\mathcal{A}}$, then clearly there is an edge $r^{\varepsilon} \rightarrow r'^{\varepsilon}$ in $R^{\varepsilon}_{\mathcal{A}}$. The converse is more difficult to prove. However the proof follows arguments similar to the ones given in the proof of Lemma 4. Let us explain them, with less details.[20]

[19] We again use the notation discussed in Remark 14
[20] We use the notation of the proof of Lemma 4. Fig. 11-16 will be helpful.

Let $r^\varepsilon \to r'^\varepsilon$ be an edge in $R^\varepsilon_\mathcal{A}$. It is a switch-edge or a time-edge. We have to show that there exists an edge $r^\delta \to r'^\delta$ in $R^\delta_\mathcal{A}$. If $r^\varepsilon \to r'^\varepsilon$ is a switch-edge, it is not difficult to verify that $r^\delta \to r'^\delta$ exists.

We now treat the case where $r^\varepsilon \to r'^\varepsilon$ is a time-edge. Let $(l,\nu) \xrightarrow{\tau} (l,\nu')$ be a time-transition in $T_\mathcal{A}$ such that $(l,\nu) \in r^\varepsilon$ and $(l,\nu') \in r'^\varepsilon$. We define new clock values μ from ν as follows

$$\mu_i = \begin{cases} \lfloor \nu_i \rfloor + \frac{\delta}{2\varepsilon}\bar{\nu}_i & \text{if } x_i \in \mathsf{Low}(r^\varepsilon) \\ \lfloor \nu_i \rfloor + 1 - (\frac{\delta}{2\varepsilon}(1-\bar{\nu}_i)) & \text{if } x_i \in \mathsf{High}(r^\varepsilon) \end{cases}$$

One verifies that for each i

$$\bar{\mu}_i \in [0, \frac{\delta}{2}[\cup]1-\frac{\delta}{2}, 1[.$$

In particular, $(l,\mu) \in r^\delta$. If we exhibit a time-transition $(l,\mu) \xrightarrow{\tau'} (l,\mu')$ in $T_\mathcal{A}$ with $(l,\mu') \in r^\delta$, then we obtain the required time-edge $r^\delta \to r'^\delta$ of $R^\delta_\mathcal{A}$.

First we suppose that r' is a closed region. Hence, there exists a clock x_f such that $\bar{\nu}'_f = 0$. It follows that $\tau = N - \nu_f$ with $N = \nu'_f \in \mathbb{N}$. We define $\tau' = N - \mu_f$ and $\mu' = \mu + \tau'$. Let us show that $(l,\mu') \in r'^\delta$, i.e. $\bar{\mu}'_i \in [0,\delta[\cup]1-\delta,1[$ for each i. We have to distinguish four cases.

1. $x_f \in \mathsf{Low}(r^\varepsilon)$.
   (a) $x_i \in \mathsf{Low}(r^\varepsilon)$.
   If $\bar{\mu}_i \geq \bar{\mu}_f$, then $\bar{\mu}'_i = \bar{\mu}_i - \bar{\mu}_f$. We have $0 \leq \bar{\mu}'_i < \frac{\delta}{2} < \delta$. If $\bar{\mu}_i < \bar{\mu}_f$, then $\bar{\mu}'_i = 1 - (\bar{\mu}_f - \bar{\mu}_i)$. We have $1-\delta < 1-\frac{\delta}{2} < \bar{\mu}'_i < 1$.
   (b) $x_i \in \mathsf{High}(r^\varepsilon)$.
   We have $\bar{\mu}'_i = \bar{\mu}_i - \bar{\mu}_f$. We conclude that $1-\delta = 1-2\frac{\delta}{2} < \bar{\mu}'_i < 1$.
2. $x_f \in \mathsf{High}(r^\varepsilon)$.
   (a) $x_i \in \mathsf{Low}(r^\varepsilon)$.
   We have $\bar{\mu}'_i = \bar{\mu}_i + (1-\bar{\mu}_f)$. Hence $0 \leq \bar{\mu}'_i < 2\frac{\delta}{2} = \delta$.
   (b) $x_i \in \mathsf{High}(r^\varepsilon)$.
   If $\bar{\mu}_i < \bar{\mu}_f$, then $\bar{\mu}'_i = \bar{\mu}_i + (1-\bar{\mu}_f)$ and $1-\delta < 1-\frac{\delta}{2} < \bar{\mu}'_i < 1$. If $\bar{\mu}_i \geq \bar{\mu}_f$, then $\bar{\mu}'_i = \bar{\mu}_i - \bar{\mu}_f$ and $0 \leq \bar{\mu}'_i < \frac{\delta}{2} < \delta$.

We have thus proved that $(l,\mu) \xrightarrow{\tau'} (l,\mu')$ is a transition in $T_\mathcal{A}$ with $(l,\mu') \in r'^\delta$.

We now treat the case where r' is an open region. Either there exists a closed region r'' such that $r \to r'' \to r'$ and $r' = \text{succ}(r'')$, or r'' does not exist and then $r = r'$.

1. If r'' exists, by using the previous case, we can find a transition $(l,\mu) \xrightarrow{\tau'_1} (l,\mu'')$ in $T_\mathcal{A}$ such that $(l,\nu'') \in r''^\delta$. We then choose τ'_2 such that $\tau'_2 < \min(\delta - \bar{\mu}''_b, 1-\bar{\mu}''_d)$, and we define $\mu' = \mu + \tau'_1 + \tau'_2$. It follows that $(l,\mu) \xrightarrow{\tau'_1+\tau'_2} (l,\mu')$ is a transition in $T_\mathcal{A}$ such that $(l,\mu') \in r'^\delta$.
2. If r'' does not exist, then $r = r'$. In the case $r^\varepsilon = r'^\varepsilon$, we proceed with an argument similar to the one of the previous case with $\tau' < \min(\delta - \bar{\mu}_b, 1-\bar{\nu}_d)$.
   In the case $r^\varepsilon \neq r'^\varepsilon$, we show as in the proof of Lemma 4 that

   $$\mathsf{Low}(r^\varepsilon) = \mathsf{High}(r'^\varepsilon), \text{ and } \mathsf{High}(r^\varepsilon) = \mathsf{Low}(r'^\varepsilon) = \varnothing. \tag{16}$$

   We then choose $\tau' = 1-\delta$. Let us show that, with $\mu' = \mu + \tau'$, we have $(l,\mu') \in r'^\delta$, that is, $1-\delta < \bar{\mu}'_c$ and $\bar{\mu}'_d < 1$. We have $\bar{\mu}'_c = \bar{\mu}_a + 1 - \delta > 1-\delta$, and $\bar{\mu}'_d = \bar{\mu}_b + 1 - \delta < 1$.

The proof is completed. □

Due to the previous lemma, the only difference between the ε-region graphs, with $\varepsilon \in]0, \frac{1}{3}]$, is the size of their ε-regions depending on ε. We thus introduce the following graph, independently of any ε, which is isomorphic to all $R^\varepsilon_\mathcal{A}$. It can be seen as the limit graph of $R^\varepsilon_\mathcal{A}$ when ε converges to 0.

Definition 14. *Let $\mathcal{A}$ be a timed automaton. We denote by $\dot{R}_{\mathcal{A}} = (\dot{S}, \rightarrow)$ a graph isomorphic to each $R^{\varepsilon}_{\mathcal{A}} = (S^{\varepsilon}, \rightarrow)$, with $\varepsilon \in]0, \frac{1}{3}]$, and we call it the* discrete graph *of $\mathcal{A}$. We also use the same terminology of switch-edge and time-edge.*

Remark 19. In the sequel, as done in Remark 14, we use the same letter r to express that the vertex $\dot{r}$ of $\dot{S}$ is *isomorphic* to the vertex r^{ε} of S^{ε}. Moreover, we say that the edge $\dot{r} \rightarrow \dot{r}'$ is *isomorphic* to $r^{\varepsilon} \rightarrow r'^{\varepsilon}$, and that the path $\dot{r} \rightsquigarrow \dot{r}'$ is *isomorphic* to $r^{\varepsilon} \rightsquigarrow r'^{\varepsilon}$.

We now want to augment the discrete graph with a weight function. First, in the next lemma, we show that given a time-edge $r^{\varepsilon} \rightarrow r'^{\varepsilon}$ in the ε-region graph $R^{\varepsilon}_{\mathcal{A}}$, we can associate a unique integer N which represents, up to 2ε, the time elapsed between r^{ε} and r'^{ε}. We recall that both ε-regions r^{ε} and r'^{ε} are bounded (See Remark 2).

Let us notice that it is impossible to associate a unique integer with an edge $r \rightarrow r'$ of the region graph $R_{\mathcal{A}}$ in such a way.

Lemma 7. *Let $\mathcal{A}$ be a timed automaton. Let $r^{\varepsilon} \rightarrow r'^{\varepsilon}$ be a time-edge in the ε-region graph $R^{\varepsilon}_{\mathcal{A}}$, with $\varepsilon \in]0, \frac{1}{6}]$. Then there exists a unique $N \in \mathbb{N}$ such that for all time-transitions $(l, \nu) \xrightarrow{\tau} (l, \nu')$ in $T_{\mathcal{A}}$ with $(l, \nu) \in r^{\varepsilon}$, $(l, \nu') \in r'^{\varepsilon}$:*

$$|\tau - N| < 2\varepsilon.$$

Moreover, N is independent of ε.

Proof. Let $(l, \nu) \xrightarrow{\tau} (l, \nu')$ be a time-transition such that $(l, \nu) \in r^{\varepsilon}$ and $(l, \nu') \in r'^{\varepsilon}$. We first prove that there exists $N \in \mathbb{N}$ such that $|\tau - N| < 2\varepsilon$. We then prove that this integer N is the same for all such time-transitions.

1. Existence. Assume the contrary, that is, $|\tau - N| \geq 2\varepsilon$ for all $N \in \mathbb{N}$. In particular for $M = \lfloor \tau \rfloor$, we have $\tau = M + \tau'$ and $2\varepsilon \leq \tau' \leq 1 - 2\varepsilon$. Let x_i be a clock. We consider two cases according to $x_i \in \mathsf{Low}(r^{\varepsilon})$ or $x_i \in \mathsf{High}(r^{\varepsilon})$. Let us study bounds for $\nu'_i = \nu_i + \tau$.
   (a) $x_i \in \mathsf{Low}(r^{\varepsilon})$. Thus we have
   $$M + 2\varepsilon \leq \nu_i + M + \tau' = \nu'_i < \varepsilon + M + (1 - 2\varepsilon) = (M + 1) - \varepsilon.$$
   It follows that $2\varepsilon \leq \bar{\nu}_i < 1 - 2\varepsilon$. This contradicts $(l, \nu') \in r'^{\varepsilon}$.
   (b) $x_i \in \mathsf{High}(r^{\varepsilon})$. It follows that
   $$(M + 1) + \varepsilon = (1 - \varepsilon) + M + 2\varepsilon < \nu_i + M + \tau' = \nu'_i < 1 + M + (1 - 2\varepsilon) = (M + 2) - 2\varepsilon.$$
   It follows that $\varepsilon < \bar{\nu}_i \leq 1 - 2\varepsilon$ again in contradiction with $(l, \nu') \in r'^{\varepsilon}$.
2. Uniqueness. We consider two time-transitions $(l, \nu) \xrightarrow{\tau} (l, \nu')$ and $(l, \tilde{\nu}) \xrightarrow{\tilde{\tau}} (l, \tilde{\nu}')$ such that $(l, \nu), (l, \tilde{\nu}) \in r^{\varepsilon}$ and $(l, \nu'), (l, \tilde{\nu}') \in r'^{\varepsilon}$. We know that there exist $N, \tilde{N} \in \mathbb{N}$ such that $|\tau - N| < 2\varepsilon$ and $|\tilde{\tau} - \tilde{N}| < 2\varepsilon$. Let us show that $N = \tilde{N}$.
   $$|\tilde{N} - N| = |(\tau - N) - (\tilde{\tau} - \tilde{N}) + (\tilde{\tau} - \tau)| < 4\varepsilon + |\tilde{\tau} - \tau|.$$
   For all $i \in \{1, \ldots, n\}$, we have $\nu'_i = \nu_i + \tau$ and $\tilde{\nu}'_i = \tilde{\nu}_i + \tilde{\tau}$. Moreover we recall that $(l, \nu), (l, \tilde{\nu}) \in r^{\varepsilon}$ and $(l, \nu'), (l, \tilde{\nu}') \in r'^{\varepsilon}$. Therefore
   $$|\tilde{\tau} - \tau| = |(\tilde{\nu}'_i - \nu'_i) - (\tilde{\nu}_i - \nu_i)| < 2\varepsilon.$$
   It follows that
   $$|\tilde{N} - N| < 6\varepsilon.$$
   By hypothesis $\varepsilon \leq \frac{1}{6}$. Hence $N = \tilde{N}$.

It remains to prove that N is independent of ε. Let $\varepsilon, \varepsilon' \in]0, \frac{1}{6}]$ and $N, N' \in \mathbb{N}$ be such that $|\tau - N| < 2\varepsilon$ and $|\tau - N'| < 2\varepsilon'$. Then

$$|N' - N| = |(\tau - N) + (\tau - N')| < 2\varepsilon + 2\varepsilon' < 1.$$

Therefore, $N = N'$.

□

Remembering the definition of the discrete graph $\dot{R}_{\mathcal{A}}$ (see Definition 14), the number N proposed in Lemma 7 for the time-edge $r^\varepsilon \to r'^\varepsilon$ of $R^\varepsilon_{\mathcal{A}}$ can also be associated with the time-edge $\dot{r} \to \dot{r}'$ of $\dot{R}_{\mathcal{A}}$ isomorphic to $r^\varepsilon \to r'^\varepsilon$.

We now consider $\mathcal{A}$ as a weighted timed automaton $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{C})$, and we explain how to assign a weight to each edge of the discrete graph $\dot{R}_{\mathcal{A}}$ of $\mathcal{A}$, in relation with the cost function $\mathcal{C}$. Let $\varepsilon \in]0, \frac{1}{6}]$ and let $\dot{r} \to \dot{r}'$ be an edge of $\dot{R}_{\mathcal{A}}$. It is isomorphic to an edge $r^\varepsilon \to r'^\varepsilon$ of the ε-region graph $R^\varepsilon_{\mathcal{A}}$. Consider a transition

$$(l, \nu) \to (l', \nu') \tag{17}$$

in $T_{\mathcal{A}}$ such that $(l, \nu) \in r^\varepsilon$ and $(l', \nu') \in r'^\varepsilon$. It is a time-transition $(l, \nu) \xrightarrow{\tau} (l', \nu')$ or a switch-transition $(l, \nu) \xrightarrow{e} (l', \nu')$.

1. Transition $(l, \nu) \xrightarrow{\tau} (l', \nu')$. In this case, $\dot{r} \to \dot{r}'$ is a time-edge. We associate with it a weight $\mathcal{W}(\dot{r}, \dot{r}')$ equal to
$$\mathcal{W}(\dot{r}, \dot{r}') = N \cdot \mathcal{C}(l) \tag{18}$$
where N is the unique integer of Lemma 7.
2. Transition $(l, \nu) \xrightarrow{e} (l', \nu')$. Thus $\dot{r} \to \dot{r}'$ is a switch-edge. We associate with it a weight $\mathcal{W}(\dot{r}, \dot{r}')$ equal to
$$\mathcal{W}(\dot{r}, \dot{r}') = \mathcal{C}(e). \tag{19}$$

Definition 15. *Let $\mathcal{A}$ be a weighted timed automaton. The* weighted discrete graph $\dot{R}^w_{\mathcal{A}} = (\dot{S}, \to, \mathcal{W})$ *of $\mathcal{A}$ is the discrete graph $\dot{R}_{\mathcal{A}}$ of $\mathcal{A}$ augmented with the weight function $\mathcal{W}$ as defined in (18) and (19).*

Remark 20. We are conscious that this definition is incorrect in some very particular cases. Indeed (see Remark 1), both weights defined in (18), (19) can be assigned to the same edge $\dot{r} \to \dot{r}'$ when the transition $(l, \nu) \to (l', \nu')$ defined in (17) is both a time-transition and a switch-transition. If such a case happens, the edge $\dot{r} \to \dot{r}'$ must be duplicated in a way that each of the two weights is assigned to each of the two copies.

Remark 21. We notice that weights labeling the edges of $\dot{R}^w_{\mathcal{A}}$ are polynomials in the constants appearing in $\mathcal{A}$ (see (18) (19). Therefore, since $|R^\varepsilon_{\mathcal{A}}|$ is in $\mathcal{O}(2^{|\mathcal{A}|})$ by Remark 18, we also have $|\dot{R}^w_{\mathcal{A}}|$ in $\mathcal{O}(2^{|\mathcal{A}|})$.

Definition 16. *Let $\mathcal{A}$ be a weighted timed automaton. Let $\dot{\rho} = \dot{r}_0 \to \dot{r}_1 \to \dot{r}_2 \cdots \to \dot{r}_m$ be a path in $\dot{R}^w_{\mathcal{A}}$. Then the weight $\mathcal{W}(\dot{\rho})$ of $\dot{\rho}$ is equal to*

$$\mathcal{W}(\dot{\rho}) = \sum_{k=0}^{m-1} \mathcal{W}(\dot{r}_k, \dot{r}_{k+1}).$$

It is an integer number.

In the next two lemmas, we relate the weight of paths in $\dot{R}^w_{\mathcal{A}}$ to the cost of runs in $T^\varepsilon_{\mathcal{A}}$. These lemmas are the counterparts of Lemmas 4 and 5 with weight.

Lemma 8. *Let $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{C})$ be a weighted timed automaton and let $K = \sum_{l \in L} |\mathcal{C}(l)|$. Let $\dot{\rho} = \dot{r} \rightsquigarrow \dot{r}'$ be an initialized path of length m in $\dot{R}^w_{\mathcal{A}}$. Let $\varepsilon \in]0, \frac{1}{6}]$. Then there exist two ε-regions $r^\varepsilon, r'^\varepsilon$ of $R^\varepsilon_{\mathcal{A}}$ respectively isomorphic to $\dot{r}, \dot{r}'$, and there exists an ε-run $\rho^\varepsilon = q \rightsquigarrow q'$ of length m in $T^\varepsilon_{\mathcal{A}}$ such that*

$$|\mathcal{W}(\dot{\rho}) - \mathcal{C}(\rho^\varepsilon)| \le 2\varepsilon K m$$

and $q \in r^\varepsilon$, $q' \in r'^\varepsilon$.

Proof. Suppose $\dot{\rho}$ has the form $\dot{r}_0 \to \dot{r}_1 \to \cdots \to \dot{r}_m$. It is isomorphic to the ε-run $\rho_{S^\varepsilon} = r^\varepsilon_0 \to r^\varepsilon_1 \to \cdots \to r^\varepsilon_m$ in $R^\varepsilon_{\mathcal{A}}$. Since $\dot{\rho}$ is initialized, $r^\varepsilon_0 = [(l_0, 0)]^\varepsilon$ for some location l_0. By Lemma 4, there exists an ε-run $\rho^\varepsilon = (l_0, 0) \to (l_1, \nu^1) \to \cdots \to (l_m, \nu^m)$ in $T^\varepsilon_{\mathcal{A}}$ such that $(l_k, \nu^k) \in r^\varepsilon_k$ for all k. Looking at Definitions 6 and 16, by Lemma 7, we verify that $|\mathcal{W}(\dot{\rho}) - \mathcal{C}(\rho^\varepsilon)| \le 2\varepsilon K m$. ☐

Lemma 9. *Let $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{C})$ be a weighted timed automaton and let $K = \sum_{l \in L} |\mathcal{C}(l)|$. Let $\rho^\delta = q \rightsquigarrow q'$ be an initialized δ-run of length m in $T^\delta_\mathcal{A}$, with $\delta \in]0, \frac{1}{6(m+1)}]$. Then there exist two ε-regions $r^\varepsilon, r'^\varepsilon$ of $R^\varepsilon_\mathcal{A}$ such that $q \in r^\varepsilon$, $q' \in r'^\varepsilon$, and there exists a path $\dot\rho = \dot r \rightsquigarrow \dot r'$ of length m in $\dot R^w_\mathcal{A}$ such that $\dot r, \dot r'$ are respectively isomorphic to $r^\varepsilon, r'^\varepsilon$ and*

$$|\mathcal{W}(\dot\rho) - \mathcal{C}(\rho^\delta)| \leq 2\varepsilon K m$$

with $\varepsilon = (m+1)\delta$.

Proof. Suppose that ρ^δ is of the form $(l_0, 0) \to (l_1, \nu^1) \to \cdots \to (l_m, \nu^m)$. By Lemma 5, there exists a path $\rho_{S^\varepsilon} = r^\varepsilon_0 \to r^\varepsilon_1 \to \cdots \to r^\varepsilon_m$ in $R^\varepsilon_\mathcal{A}$ such that $(l_k, \nu^k) \in r^\varepsilon_k$ for all $k \in \{0, \ldots, m\}$. We consider the isomorphic path $\dot\rho = \dot r_0 \to \dot r_1 \to \cdots \to \dot r_m$ of $\dot R^w_\mathcal{A}$. As in the proof of Lemma 8 we conclude that $|\mathcal{W}(\dot\rho) - \mathcal{C}(\rho^\delta)| < 2\varepsilon K m$. ◻

Let $\mathcal{A}$ be a timed automaton. Let r, r' be two regions of $R_\mathcal{A}$ where r satisfies the first assumption of Remark 8, i.e., r is composed of a unique state of the form $(l, 0)$. We are going to state an important result about $\mathsf{OptCost}(r, r')$. Before, we need to fix some notation. Thus, given $\varepsilon \in]0, \frac{1}{2}]$, there is exactly one ε-region r^ε included in r (also composed of the unique state $(l, 0)$). We denote by $\dot r$ the vertex of $\dot R^w_\mathcal{A}$ isomorphic to r^ε. On the hand, the region r' gives rise to at most $n+1$ different ε-regions $r'^\varepsilon \subseteq r'$ (see Remark 18). We denote by $\mathcal{S}(r')$ this set of ε-regions, and by $\dot{\mathcal{S}}(r')$ the set of vertices of $\dot R^w_\mathcal{A}$ that are isomorphic to them.

Theorem 3. *Let $\mathcal{A}$ be a weighted timed automaton and r, r' two regions of $R_\mathcal{A}$. Then*

$$\mathsf{OptCost}(r, r') = \inf\{\mathcal{W}(\dot\rho) \mid \exists \dot r' \in \dot{\mathcal{S}}(r'),\ \dot\rho = \dot r \rightsquigarrow \dot r' \text{ path in } \dot R^w_\mathcal{A}\}. \quad (20)$$

Proof. We denote $\inf\{\mathcal{W}(\dot\rho) \mid \exists \dot r' \in \dot{\mathcal{S}}(r'),\ \dot\rho = \dot r \rightsquigarrow \dot r'\}$ by $\mathsf{InfWeight}$. Suppose $\mathsf{OptCost}(r, r') = +\infty$, i.e. there is no run $\rho = q \rightsquigarrow q'$ of $T_\mathcal{A}$ such that $q \in r$, $q' \in r'$, then there is no path $\dot\rho = \dot r \rightsquigarrow \dot r'$ for any $\dot r' \in \dot{\mathcal{S}}(r')$. Otherwise, by Lemma 8, there exists an ε-run $\rho^\varepsilon = q \rightsquigarrow q'$ with $q \in r^\varepsilon$ and $q' \in r'^\varepsilon$. This ε-run can be seen as a run $\rho = q \rightsquigarrow q'$ of $T_\mathcal{A}$ with $q \in r$ and $q' \in r'$, a contradiction. So $\mathsf{InfWeight} = +\infty$ and (20) holds in this case.

Assume $\mathsf{OptCost}(r, r') \in \mathbb{R} \cup \{-\infty\}$ and $\mathsf{OptCost}(r, r') < \mathsf{InfWeight}$. By Corollary 5, it follows that there is a path $\rho_R = r \rightsquigarrow r'$ in $R_\mathcal{A}$ with length m such that $\mathsf{OptCost}(\rho_R) < \mathsf{InfWeight}$. By Lemmas 3 and 9 respectively used with ε and δ chosen small enough, we can find a path $\dot\rho = \dot r \rightsquigarrow \dot r'$ in $\dot R^w_\mathcal{A}$ such that $\dot r' \in \dot{\mathcal{S}}(r')$ and $\mathcal{W}(\dot\rho) < \mathsf{InfWeight}$. This is impossible.

Assume now that $\mathsf{OptCost}(r, r') \in \mathbb{R}$ and $\mathsf{OptCost}(r, r') > \mathsf{InfWeight}$. By definition of the $\inf$ operator, we have $\mathsf{OptCost}(r, r') > \mathcal{W}(\dot\rho)$ for some $\dot\rho = \dot r \rightsquigarrow \dot r'$ with $\dot r' \in \dot{\mathcal{S}}(r')$. We get a contradiction using Lemma 8 with ε chosen small enough.

This proves (20). ◻

4.4 Complexity

In this section, we prove the main result of this paper, that is the cost-optimal reachability problem is PSPACE-COMPLETE (Theorem 1).

Proof (of Theorem 1). We begin with some preliminary considerations. The discrete graph $\dot R^w_\mathcal{A}$ has size in $\mathcal{O}(2^{|\mathcal{A}|})$, and the weights labelling its edges are polynomials in the constants appearing in $\mathcal{A}$ (see Remark 21). In the sequel of the proof, we consider paths $\dot\rho$ in $\dot R^w_\mathcal{A}$ with a length bounded by the number of vertices of $\dot R^w_\mathcal{A}$, thus with a length at most exponential in $|\mathcal{A}|$. These paths are called *elementary*. Therefore, the encoding of the cost of an elementary path $\dot\rho$ can be done in PSPACE.

Let us now prove that the cost-optimal reachability problem is in PSPACE. By Theorem 3, computing the optimal cost $\mathsf{OptCost}(r, r')$ given two regions r, r' of $R_\mathcal{A}$, reduces in computing $\inf\{\mathcal{W}(\dot\rho) \mid \exists \dot r' \in \dot{\mathcal{S}}(r'),\ \dot\rho = \dot r \rightsquigarrow \dot r' \text{ path in } \dot R^w_\mathcal{A}\}$. There are three possibilities :

- there is no path $\dot\rho = \dot r \rightsquigarrow \dot r'$ with $\dot r' \in \dot{\mathcal{S}}(r')$ in $\dot R^w_\mathcal{A}$, and thus $\mathsf{OptCost}(r, r') = +\infty$;
- there is such a path $\dot\rho$ containing a cycle with a negative weight, and thus $\mathsf{OptCost}(r, r') = -\infty$;
- there is such a path $\dot\rho$, and none of these paths contains a cycle with a negative weight. Therefore $\mathsf{OptCost}(r, r')$ is an integer equal to the minimum value of $\{\mathcal{W}(\dot\rho) \mid \exists \dot r' \in \dot{\mathcal{S}}(r'),\ \dot\rho = \dot r \rightsquigarrow \dot r'\}$.

Let us notice that in the three previous situations, the considered paths and cycles can be supposed to be elementary. In the third situation, a path $\dot\rho$ with a minimum value $\mathcal{W}(\dot\rho)$ can also supposed to be elementary. The algorithm works as follows.

1. Guess an elementary path $\dot\rho = \dot r \rightsquigarrow \dot r'$ for some $\dot r' \in \dot{\mathcal{S}}(r')$. Note that the length of $\dot\rho$ is exponential in $|\mathcal{A}|$, and that each vertex of $\dot R^w_{\mathcal{A}}$ can be stored in polynomial space. Hence one can decide in NPSPACE, thus in PSPACE, whether $\mathsf{OptCost}(r, r')$ is equal to $+\infty$ or not.
2. We assume $\mathsf{OptCost}(r, r') \neq +\infty$.
   Guess a vertex $\dot r_0$ in $\dot R^w_{\mathcal{A}}$, and check whether there exist an elementary path from $\dot r$ to $\dot r_0$ and another one from $\dot r_0$ to some $\dot r' \in \mathcal{S}(r')$ (as explained in 1., this can be done in PSpace). Then guess an elementary cycle from $\dot r_0$ to $\dot r_0$ and compute on-the-fly its weight (as explained at the beginning of the proof, the computation of this weight can be done in PSPACE). Therefore it can be decided in PSPACE whether $\mathsf{OptCost}(r, r')$ is equal to $-\infty$ or not.
3. We assume $\mathsf{OptCost}(r, r') \in \mathbb{Z}$.
   Guess an elementary path $\dot\rho = \dot r \rightsquigarrow \dot r'$ with $\dot r' \in \dot{\mathcal{S}}(r')$, and compute on-the-fly its weight $\mathcal{W}(\dot\rho)$. As explained in 2., this can be done in PSPACE. Store the weight $\mathcal{W}(\dot\rho)$ in variable *aux*. If there is no elementary path $\dot\rho_1 = \dot r \rightsquigarrow \dot r'_1$ with $\dot r'_1 \in \dot{\mathcal{S}}(r')$ with a weight strictly less than *aux*, then it means that $\mathsf{OptCost}(r, r')$ is equal to *aux*. Therefore guess such a path $\dot\rho_1$, compute its weight $\mathcal{W}(\dot\rho_1)$ on-the-fly, and compare $\mathcal{W}(\dot\rho_1)$ with *aux*. It follows that the complexity of this procedure is in N-(CO-NPSPACE), thus in PSPACE.

The proposed algorithm is globally in PSPACE showing that the cost-optimal reachability problem is in PSPACE. It remains to prove that it is PSPACE-hard. We do that by reduction of the reachability problem for timed automata known to be PSPACE-complete [AD94]. Let $\mathcal{A}$ be a timed automaton. We augment it with a cost function $\mathcal{C}$ that assigns a null cost to each location and edge of $\mathcal{A}$. Then, trivially, a region r' is reachable from a region r if and only if the optimal cost $\mathsf{OptCost}(r, r')$ is different from $+\infty$. □

We conclude Sect. 4 with the following important remark.

Remark 22. In Remark 12, we have mentioned that Problem 2 remains decidable if the duration cost is a concave function (resp. convex function) and the considered optimum cost is an infimum (resp. supremum).

Given a weighted timed automaton $\mathcal{A}$, we recall that the definitions of ε-semantics $T^\varepsilon_{\mathcal{A}}$, ε-region graph $R^\varepsilon_{\mathcal{A}}$ and discrete graph $\dot R_{\mathcal{A}}$ have been introduced independently of the cost function $\mathcal{C}$ used in $\mathcal{A}$. Their definition was only based on the crucial Corollary 3 indicating that when computing an optimum cost, only time-transitions with a time τ arbitrarily close to an integer have to be considered.

In Definition 15, we have shown how to augment the discrete graph $\dot R_{\mathcal{A}}$ with a weight function $\mathcal{W}$ in relation with $\mathcal{C}$. We have given the related Lemmas 8 and 9.

Let us consider some possible generalizations of cost and weight functions. In (18), given a time-transition $(l, \nu) \xrightarrow{\tau} (l', \nu')$ in $T_{\mathcal{A}}$ and the related time-edge $\dot r \to \dot r'$ in $\dot R_{\mathcal{A}}$, the duration cost of the time-transition is equal to

$$\tau \cdot \mathcal{C}(l), \tag{21}$$

and the weight of the time-edge is equal to

$$N \cdot \mathcal{C}(l). \tag{22}$$

The number N is the unique integer of Lemma 7 satisfying $|\tau - N| < 2\varepsilon$. Suppose that (21) and (22) are respectively replaced by $f(\tau) \cdot \mathcal{C}(l)$ and $f(N) \cdot \mathcal{C}(l)$ where f is a continuous function. It follows that we still have an analog of Lemma 7 with $|f(\tau) - f(N)| < \delta$ and δ small enough, as well as the analog of Lemmas 8 and 9. Therefore, Theorem 3 remains true with a concave duration cost function and the continuous function f mentioned above.[21] If additionally these functions are computable, we get a generalization of Theorem 1.

[21] For instance with $f = \ln$ and $\mathcal{C}_d(\rho(t_1, \ldots, t_m)) = \sum_{k \in \{1,\ldots,m\}} \mathcal{C}(l_k) \cdot \ln(t_k)$ (see (10)).

5 Assumptions

Till this section, the whole paper has been written under two assumptions concerning Problem 1 (see Remark 8) : First, the region r given in Problem 1 is composed of a unique state of the form $(l, 0)$. Second, the infimum cost is only considered. On the other hand, we have supposed in Remark 2 that the timed automata of this paper are diagonal-free and bounded. We show in this section that all these assumptions can be discarded.

5.1 Supremum cost

Let us go through the paper and indicate the modifications to be done when the supremum cost is considered instead of the infimum cost.

In Definition 7, the optimal cost $\mathsf{OptCost}(r, r')$ is the supremum of the costs of the runs $\rho = q \rightsquigarrow q'$ of $T_{\mathcal{A}}$ such that $q \in r$ and $q' \in r'$. It is equal to $-\infty$ when there is no such run ρ. Otherwise it belongs to $\mathbb{R} \cup \{+\infty\}$. Similarly, in Definition 8, the optimal cost $\mathsf{OptCost}(\rho_R)$ is the supremum of the costs $\mathcal{C}(\rho)$ among the runs ρ of $T_{\mathcal{A}}$ such that $[\rho] = \rho_R$.

The proof of Corollary 1 stating that Problem 2 is decidable is the same. Indeed the Simplex Method acts similarly when a supremum or an infimum value has to be computed. Here the supremum value of $\mathcal{C}_d(\rho(t_1, \ldots, t_m))$ is also obtained at one of the vertices of the polyhedron $\overline{\mathsf{Pol}(\rho_R)}$. Therefore Corollaries 4 and 5 also hold for the supremum costs $\mathsf{OptCost}(\rho_R)$ and $\mathsf{OptCost}(r, r')$.[22]

In the case of a supremum cost, Theorem 3 states that

$$\mathsf{OptCost}(r, r') = \sup\{\mathcal{W}(\dot{\rho}) \mid \exists \dot{r}' \in \dot{\mathcal{S}}(r'),\ \dot{\rho} = \dot{r} \rightsquigarrow \dot{r}' \text{ path in } \dot{R}^w_{\mathcal{A}}\}.$$

The proof has to be adapted since the $\sup$ operator is considered. This can be done easily.

The proof of Theorem 1 essentially remains the same. It must be slightly adapted to deal with the $\sup$ operator instead of the $\inf$ operator.

5.2 Any region r

In Definition 7, the optimal cost $\mathsf{OptCost}(r, r')$ is defined for any regions r, r' of $R_{\mathcal{A}}$. Along the paper, we have assumed that r is composed of a unique state of the form $(l, 0)$. We now indicate the modifications to be done when r is *any* region. We here come back to the infimum cost.

We first consider Sect. 3.2 dedicated to the solution of Problem 2. The approach is similar : Given $\rho_R = r \rightsquigarrow r'$ a path in $R_{\mathcal{A}}$, we construct a set of constraints $\mathsf{Constr}(\rho_R)$ that define a polyhedron $\mathsf{Pol}(\rho_R)$. The optimal cost $\mathsf{OptCost}(r, r')$ is then computed thanks to one of the vertices of $\overline{\mathsf{Pol}(\rho_R)}$.

Let us go into details. We use the same notation as in Section 3.2. Let us write ρ_R as in (4)

$$\rho_R = r'_0 \rightarrow r_1 \rightarrow r'_1 \rightarrow r_2 \cdots \rightarrow r_m \rightarrow r'_m.$$

The runs ρ of $T_{\mathcal{A}}$ such that $[\rho] = \rho_R$ can be parameterized as done in (6), with the difference that the first region r'_0 is not equal to $[(l_1, 0)]$. Instead of (6), we write

$$\rho(t_1, t_2, \ldots, t_{n+m}) = q'_0 \overset{t_{n+1}}{\rightarrow} q_1 \overset{e_1}{\rightarrow} q'_1 \overset{t_{n+2}}{\rightarrow} q_2 \overset{e_2}{\rightarrow} \cdots \overset{t_{n+m}}{\rightarrow} q_m \overset{e_m}{\rightarrow} q'_m$$

such that

- the state q'_0 depends on the parameters $t_1, t_2, \ldots, t_n$,
- each state q_k (resp. q'_k) depends on the parameters $t_1, t_2, \ldots, t_{n+k}$, for $k \in \{1, \ldots, m\}$.

[22] Of course, the $\inf$ operator has to be replaced by the $\sup$ operator in Corollary 5.

Let us study the form of $q'_0 = (l_1, x'^0_1, x'^0_2, \ldots, x'^0_n) \in r'_0$. Without loss of generality we can suppose that the ordering of the clocks is as follows

$$0 \leq x'^0_1 \leq x'^0_2 \leq \cdots \leq x'^0_{n-1} \leq x'^0_n.$$

We define the n parameters $t_1, \ldots, t_n$ such that

$$t_{n-j} = \begin{cases} x'^0_1 & \text{if } j = 0 \\ x'^0_{j+1} - x'^0_j & \text{otherwise} \end{cases} \tag{23}$$

for $j \in \{0, \ldots, n-1\}$. These parameters are represented on Fig. 19. With this definition, we have $x'^0_i = x'^0_i(t_1, \ldots, t_n)$,

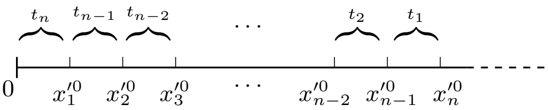

Fig. 19. The parameters $t_1, \ldots, t_n$.

for $i \in \{1, \ldots, n\}$, equal to the sum

$$x'^0_i(t_1, \ldots, t_n) = t_{n-i+1} + \cdots + t_{n-1} + t_n \tag{24}$$

which expresses a dependence on the parameters $t_1, \ldots, t_n$ like in (7).

Concerning the other states $q_k = (l_k, x^k)$ (resp. $q'_k = (l_{k+1}, x'^k)$), with $k \in \{1, \ldots, m\}$, we also have a dependence on the parameters like in (7). The clocks $x^k_i(t_1, \ldots, t_{n+k})$ and $x'^k_i(t_1, \ldots, t_{n+k})$ are either null or of the form

$$t_{h+1} + t_{h+2} + \cdots + t_{n+k-1} + t_{n+k} \tag{25}$$

with $n \leq h \leq n + k$.

Therefore, as done in (9), we have to consider the set of constraints

$$\mathsf{Constr}(\rho_R) = r'_0(t_1, \ldots, t_n) \quad \cup \bigcup_{k \in \{1, \ldots, m\}} r_k(t_1, \ldots, t_{n+k}) \tag{26}$$

With the following subsets of $(\mathbb{R}^+)^{n+m}$

$$A(\rho_R) = \{(\tau_1, \ldots, \tau_{n+m}) \in (\mathbb{R}^+)^{n+m} \mid [\rho(\tau_1, \ldots, \tau_{n+m})] = \rho_R\},$$
$$B(\rho_R) = \{(\tau_1, \ldots, \tau_{n+m}) \in (\mathbb{R}^+)^{n+m} \mid (\tau_1, \ldots, \tau_{n+m}) \models \mathsf{Constr}(\rho_R)\}.$$

we have the analog of Lemma 1, i.e.

$$A(\rho_R) = B(\rho_R).$$

The proof of this lemma is similar, except that the base case of the induction has to be adapted to the region r'_0. This is easily done by using the additional constraints $r'_0(t_1, \ldots, t_n)$ appearing in (26).

Therefore, as done in Sect. 3.2, the optimal cost $\mathsf{OptCost}(\rho_R)$ can be obtained by computing the infimum value of the duration cost $\mathcal{C}_d(\rho(t_1, \ldots, t_{n+m}))$ under the set of constraints $\mathsf{Constr}(\rho_R)$. This infimum value is obtained at one of the vertices of the polyhedron $\overline{\mathsf{Pol}(\rho_R)}$ which is the closure of the polyhedron $\mathsf{Pol}(\rho_R)$ equal to

$$\mathsf{Pol}(\rho_R) = \{(\tau_1, \ldots, \tau_{n+m}) \in (\mathbb{R}^+)^{n+m} \mid (\tau_1, \ldots, \tau_{n+m}) \models \mathsf{Constr}(\rho_R)\}.$$

This can be computed by the Simplex Method. It follows that Problem 2 is decidable (Corollary 1) and that it is decidable whether $\mathsf{OptCost}(\rho_R)$ is realizable (Corollary 2).

Let us now go through Sect. 3.3. All the results of this section are similar because we have equations (24) and (25) like in (7) that express each clock as a sum of consecutive t_k.

In particular, since the vertices of the polyhedron $\overline{\mathsf{Pol}(\rho_R)}$ have integer coordinates, a run $\rho = \rho(\tau_1, \ldots, \tau_{n+m})$ with a cost $\mathcal{C}(\rho)$ arbitrarily close to $\mathsf{OptCost}(\rho_R)$ has its first state $q'_0 \in r'_0$ with its clock values arbitrarily close to an integer (see (23)).

In Sect. 3.4, due to the previous discussion, the statement of Lemma 3 is modified as follows.

Lemma 3. *Let $\mathcal{A}$ be a weighted timed automaton, and $\rho_R = r \rightsquigarrow r'$ be a canonical path in $R_{\mathcal{A}}$. Let $\varepsilon \in]0, \frac{1}{2}]$. Then there exists an ε-run $\rho^\varepsilon = q \rightsquigarrow q'$ in $T^\varepsilon_{\mathcal{A}}$ such that $[\rho^\varepsilon] = \rho_R$,*

$$|\mathsf{OptCost}(\rho_R) - \mathcal{C}(\rho^\varepsilon)| < \varepsilon$$

and $q \in r^\varepsilon$.

The only modification appears at the end of the lemma, with $q \in r^\varepsilon$. The proof remains the same.

We now go to Sect. 4. We have to pay attention to Lemmas 4, 5, 8 and 9, and to Theorems 3 and 1. We indicate the modified statements.

Lemma 4. *Let $\mathcal{A}$ be a timed automaton and $\varepsilon \in]0, \frac{1}{3}]$. Let $\rho_{S^\varepsilon} = r^\varepsilon_0 \to r^\varepsilon_1 \to \cdots \to r^\varepsilon_m$ be a path in $R^\varepsilon_{\mathcal{A}}$. Then there exists an ε-run $\rho^\varepsilon = (l_0, \nu^0) \to (l_1, \nu^1) \to \cdots \to (l_m, \nu^m)$ in $T^\varepsilon_{\mathcal{A}}$ such that $(l_k, \nu^k) \in r^\varepsilon_k$ for all $k \in \{0, \ldots, m\}$.*

The proof of this lemma is the same except for case $k = 0$. Instead of defining the first state $(l_0, \nu^0) = (l_0, 0)$, we choose it such that $(l_0, \nu^0) \in r^{\varepsilon_0}_0$ with $\varepsilon_0 = \frac{\varepsilon}{2^m}$.

Lemma 5. *Let $\mathcal{A}$ be a timed automaton. Let $\rho^\delta = (l_0, \nu^0) \to (l_1, \nu^1) \to \cdots \to (l_m, \nu^m)$ be a δ-run in $T^\delta_{\mathcal{A}}$, such that $\delta \in]0, \frac{1}{2(m+1)}]$ and $(l_0, \nu^0) \in r^\delta_0$ for some δ-region r^δ_0 of $R^\delta_{\mathcal{A}}$. Then, with $\varepsilon = (m+1)\delta$, there exists a path $\rho_{S^\varepsilon} = r^\varepsilon_0 \to r^\varepsilon_1 \to \cdots \to r^\varepsilon_m$ in $R^\varepsilon_{\mathcal{A}}$ such that $(l_k, \nu^k) \in r^\varepsilon_k$ for all $k \in \{0, \ldots, m\}$.*

The proof of this lemma is the same except for case $k = 0$. By hypothesis, we have $(l_0, \nu^0) \in r^\delta_0 = r^{\varepsilon_0}_0$.

Lemma 8. *Let $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{C})$ be a weighted timed automaton and let $K = \sum_{l \in L} |\mathcal{C}(l)|$. Let $\dot\rho = \dot r \rightsquigarrow \dot r'$ be a path of length m in $\dot R^w_{\mathcal{A}}$. Let $\varepsilon \in]0, \frac{1}{6}]$. Then there exist two ε-regions $r^\varepsilon, r'^\varepsilon$ of $R^\varepsilon_{\mathcal{A}}$ respectively isomorphic to $\dot r, \dot r'$, and there exists an ε-run $\rho^\varepsilon = q \rightsquigarrow q'$ of length m in $T^\varepsilon_{\mathcal{A}}$ such that*

$$|\mathcal{W}(\dot\rho) - \mathcal{C}(\rho^\varepsilon)| \leq 2\varepsilon K m$$

and $q \in r^\varepsilon$, $q' \in r'^\varepsilon$.

The proof is unchanged.

Lemma 9. *Let $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{C})$ be a weighted timed automaton and let $K = \sum_{l \in L} |\mathcal{C}(l)|$. Let $\rho^\delta = q \rightsquigarrow q'$ be a δ-run of length m in $T^\delta_{\mathcal{A}}$, such that $\delta \in]0, \frac{1}{6(m+1)}]$ and $(l_0, \nu^0) \in r^\delta_0$ for some δ-region r^δ_0 of $R^\delta_{\mathcal{A}}$. Then there exist two ε-regions $r^\varepsilon, r'^\varepsilon$ of $R^\varepsilon_{\mathcal{A}}$ such that $q \in r^\varepsilon$, $q' \in r'^\varepsilon$, and there exists a path $\dot\rho = \dot r \rightsquigarrow \dot r'$ of length m in $\dot R^w_{\mathcal{A}}$ such that $\dot r, \dot r'$ are respectively isomorphic to $r^\varepsilon, r'^\varepsilon$ and*

$$|\mathcal{W}(\dot\rho) - \mathcal{C}(\rho^\delta)| \leq 2\varepsilon K m$$

with $\varepsilon = (m+1)\delta$.

The proof is unchanged.

Concerning Theorem 3, the modifications come from the fact that r is any region. Instead of having a unique vertex $\dot r$ associated to r, we now have to consider all the vertices $\dot r \in \mathcal{S}(r)$. The statement of the theorem is thus as follows, with a similar proof.

Theorem 3. *Let $\mathcal{A}$ be a weighted timed automaton and r, r' two regions of $R_{\mathcal{A}}$. Then*

$$\mathsf{OptCost}(r, r') = \inf\{\mathcal{W}(\dot\rho) \mid \exists \dot r \in \dot{\mathcal{S}}(r), \exists \dot r' \in \dot{\mathcal{S}}(r'),\ \dot\rho = \dot r \rightsquigarrow \dot r' \textit{ path in } \dot R^w_{\mathcal{A}}\}.$$

Finally, the proof of Theorem 1 is similar, except that the algorithm has to deal with paths $\dot\rho = \dot r \rightsquigarrow \dot r'$ such that $\dot r \in \dot{\mathcal{S}}(r)$ and $\dot r' \in \dot{\mathcal{S}}(r')$.

5.3 Any timed automaton

In this paper, we have restricted our study to bounded and diagonal-free timed automata. These restrictions already appear in [BBL04,LR05]. Indeed, it is well known that diagonal constraints can be removed from timed automata [BDGP98] (while preserving strong bisimilarity), and we here shortly explain how to transform a diagonal-free timed automaton into a bounded one. This construction is a folklore result. We recall it here since we could not find it in any paper of the literature.

Let $\mathcal{A} = (L, X, E, \mathcal{I}(,\mathcal{C}))$ be a (weighted) diagonal-free timed automaton. Let M be an integer strictly greater than all constants appearing in guards of $\mathcal{A}$. Then we construct the following automaton $\mathcal{A}' = (L', X, E', \mathcal{I}'(,\mathcal{C}'))$:

– the set L' of locations is $L \times 2^X$
– the set E' of edges is
  - $((l, Z), g_Z, Y, (l', Z'))$ if (l, g, Y, l') is an edge of $\mathcal{A}$, and g_Z is the guard obtained by replacing every $x \sim c$ with $x \in Z$ by either true or false, depending on $\sim$: if $\sim$ is $\geq$ or $>$, then it is replaced by true, otherwise it is replaced by false. The set Z' is equal to $Z \setminus Y$
  - $((l, Z), x = M, \{x\}, (l, Z \cup \{x\}))$ for every location (l, Z)
– the invariant $\mathcal{I}'$ is such that $\mathcal{I}'(l, Z) = \mathcal{I}(l) \wedge \bigwedge_{x \in X} x \leq M$
– The cost function $\mathcal{C}'$ is naturally defined by $\mathcal{C}'((l, Z), g_Z, Y, (l', Z')) = \mathcal{C}(l, g, Y, l')$, $\mathcal{C}'((l, Z), x = M, \{x\}, (l, Z \cup \{x\})) = 0$, and $\mathcal{C}'(l, Z) = \mathcal{C}(l)$.

Intuitively, a location (l, Z) represents the location l where all clocks in Z are inactive (*i.e.* they should be strictly above the greatest constant of $\mathcal{A}$, the truth value of every guard of $\mathcal{A}$ is thus known).

The automaton $\mathcal{A}'$ is clearly bounded (by M). It is easy to check that every run ρ of $T_{\mathcal{A}}$ has a corresponding run ρ' in $T_{\mathcal{A}'}$, and *vice-versa*. Moreover these two runs have exactly the same costs. Thus, computing the optimal cost in $\mathcal{A}$ can be reduced to computing the optimal cost in $\mathcal{A}'$.

However, the two constructions needed to restrict to bounded diagonal-free timed automata induce an exponential blowup in the number of locations of the timed automaton. More precisely, the number of locations of the resulting automaton is $|L| \cdot 2^{|\mathsf{Diag}|} \cdot 2^{|X|}$ where $|\mathsf{Diag}|$ is the number of diagonal guards in the original automaton, whereas the number of edges becomes $|E| \cdot 2^{|\mathsf{Diag}|} \cdot 2^{|X|} + (|L| \cdot 2^{|\mathsf{Diag}|} \cdot 2^{|X|}) \cdot |X|$. Nevertheless, the size of the region graph of the resulting automaton remains exponential, because exponential factors are multiplied (see Remark 6). All our complexity computations thus remain correct and computing the optimal cost also remains PSPACE-COMPLETE.

6 Application to optimal reachability in timed games

In this section, we propose an application of Theorem 1 in the context of optimal reachability timed games. Contrarily to the other sections, the presentation is quite informal, and the insight is given through an example. Optimal reachability timed games have been first introduced in [LMM02] and further studied in [ABM04,BCFL04,BBR05]. We refer to [BBR05] for precise definitions.

A *weighted timed game* $\mathcal{A}_G$ is a weighted timed automaton with a distinguished set of winning locations, and where the set of edges is split into controllable edges (played by the *controller*) and uncontrollable edges (played by the *environment*). We assume a classical definition of *strategy*, and the aim of a game is, for the controller, from the state $(l_0, 0)$, to reach a winning location and to minimize the cost of the plays, whatever does the environment. To illustrate these notions, we better give an example.

Example 9. [BCFL04] We consider the weighted timed game $\mathcal{A}_G$ of Fig. 20. Dashed (resp. plain) arrows are for uncontrollable (resp. controllable) edges. The only winning location is l_4. When the cost is non null, it is indicated on the edge/location.

Let us consider plays of the game starting from $(l_0, 0)$. If the environment chooses the edge from location l_1 to location l_2, then the accumulated cost along the game is $5t + 10(2 - t) + 1$ where t is the elapse of time at location l_0. If it chooses the edge from l_1 to l_3 is, the accumulated cost is then $5t + (2 - t) + 7$. The optimal cost the controller can ensure is thus

$$\inf_{t \leq 2} \max(5t + 10(2 - t) + 1, 5t + (2 - t) + 7) = 14 + \frac{1}{3},$$

and the optimal elapse of time is then $t = \frac{4}{3}$. The optimal strategy for the controller is thus to wait in location l_0 until $x = \frac{4}{3}$, and then enter location l_1. Then, the environment chooses to go either to l_2 or to l_3, and finally as soon as $x = 2$, the controller goes to l_4.

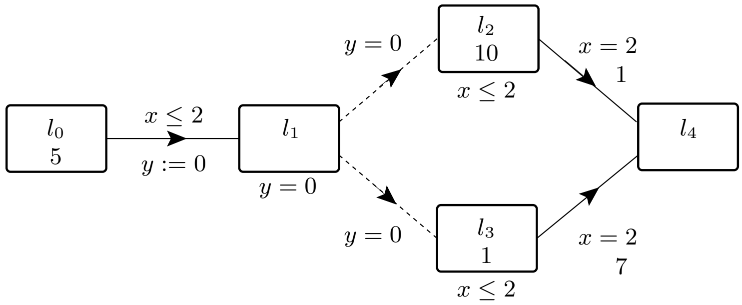

Fig. 20. A weighted timed automaton inspired from [BCFL04]

This example indicates that the region partitioning of [AD94] is not sufficient for solving optimal weighted timed games. Restricted decidability results have however been obtained in [ABM04,BCFL04]. But the general problem has been recently proved undecidable [BBR05]. Thus optimal strategies cannot in general be computed.

However as an application of Theorem 1, given a weighted timed game $\mathcal{A}_G$ and a strategy λ, we can compute the infimum (resp. supremum) cost obtained when considering executions of $\mathcal{A}_G$ *played according to* λ. This allows to compare two given strategies on a weighted timed game. A natural criterion to prefer a strategy to another one could be to choose the strategy with lower supremum cost. Let us illustrate how it works on the game $\mathcal{A}_G$ of Example 9.

When looking at Fig. 20, one can easily be convinced that a strategy on $\mathcal{A}_G$ only consists in choosing the elapse of time t at location l_0. The possible values for t are in the interval $[0, 2]$. Hence there are three natural strategies to consider: λ_i which imposes to stay i time units in location l_0 where $i = 0, 1, 2$. Considering the executions of $\mathcal{A}_G$ played according to λ_i is equivalent to consider the executions of the weighted timed automaton $\mathcal{A}_i$ depicted on Fig. 21. Let us notice that the weighted timed automaton $\mathcal{A}_i$ has not to be considered as a timed game anymore.

Following Theorem 1 one can compute the infimum cost $\mathsf{InfCost}$ (resp. supremum cost $\mathsf{SupCost}$) among the runs ρ reaching location l_4 from $(l_0, 0)$. The different cases are illustrated on Fig. 22. The results are as follows.

– On $\mathcal{A}_0$, $\mathsf{InfCost} = 9$ and $\mathsf{SupCost} = 21$,
– On $\mathcal{A}_1$, $\mathsf{InfCost} = 13$ and $\mathsf{SupCost} = 16$,
– On $\mathcal{A}_2$, $\mathsf{InfCost} = 11$ and $\mathsf{SupCost} = 17$

Thus if the criterion to prefer a strategy to another one is the lowest supremum cost, strategy λ_1 is here the prefered one.

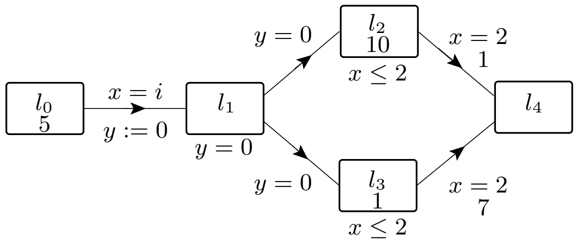

Fig. 21. The weighted timed automaton $\mathcal{A}_i$

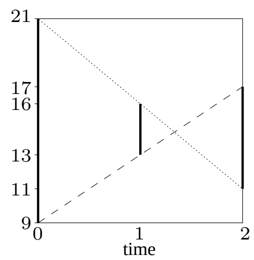

Fig. 22. $\mathsf{InfCost}$ and $\mathsf{SupCost}$ for the strategies λ_i, $i = 0, 1, 2$

Let us now briefly explain how we can use Theorem 1 in general in order to compare strategies. Given a weighted timed game $\mathcal{A}_G$ and a strategy λ, the first step is to compute the weighted timed automaton which results from the weighted timed game constrained by the strategy. Let us call $\mathcal{A}_\lambda$ this automaton. The first question we have to ask is the following. *"Is there an infinite run of $\mathcal{A}_\lambda$ that always avoids the winning locations ?"*. If the answer is *yes*, the strategy λ has to be rejected, since it does not ensure reaching a winning location. Otherwise, if the answer is *no*, we directly apply Theorem 1 to the weighted timed automaton $\mathcal{A}_\lambda$. This leads to an upper bound $\mathsf{SupCost}$ and a lower bound $\mathsf{LowCost}$ on the cost obtained by the executions of $\mathcal{A}_G$ played according to λ. Therefore different strategies λ for a weighted timed game $\mathcal{A}_G$ can be compared by referring to these values $\mathsf{SupCost}$ and $\mathsf{InfCost}$.

7 Conclusion

In this paper, we have settled the exact complexity of the cost-optimal reachability problem: it is PSPACE-COMPLETE. This result closes a gap left open by previous works where only an EXPTIME algorithm was proposed to solve the problem [ALP01].

To establish our result, we have first studied the structure of the problem and shown that a simplier version of the problem, the cost-optimal path reachability problem, is naturally related to a linear programming problem such that the associated polyhedron has vertices with integer coordinates. As a direct consequence, optimal runs using time-transitions with a time τ arbitrarily closed to an integer always exist. Using this property, a finite discrete graph called the weighted discrete graph, which refines the classical region graph, can be constructed. A formal relation between optimal paths in the discrete weighted graph and optimal runs in the weighted timed automaton is established. The construction that we propose is simple and can be explored nondeterministically to obtain an optimal PSPACE algorithm.

Furthermore, we have shown that our construction extends to more general settings: negative costs, cost-optimal reachability with respect to the supremum, concave or convexe cost functions. Finally, computing optimal costs have interesting applications in the design of controllers.

References

[ABM04] R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *ICALP'04: Automata, Languages, and Programming*, volume 3142 of *Lecture Notes in Computer Science*, pages 122–133. Springer-Verlag, 2004.

[ACD93] R. Alur, C. Courcoubetis, and D. L. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, 1993.

[ACH93] R. Alur, C. Courcoubetis, and T. A. Henzinger. Computing accumulated delays in real-time systems. In *CAV'93: Computer Aided Verification*, volume 697 of *Lecture Notes in Computer Science*, pages 181–193. Springer-Verlag, 1993.

[AD94] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[ALP01] R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In *HSCC'01: Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 49–62. Springer-Verlag, 2001.

[AM99] E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *HSCC'99: Hybrid Systems: Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 1999.

[BBL04] P. Bouyer, E. Brinksma, and K.G. Larsen. Staying alive as cheaply as possible. In *HSSC'04: Hybrid Systems: Computation and Control*, volume 2993 of *Lecture Notes in Computer Science*, pages 203–218. Springer-Verlag, 2004.

[BBR05] T. Brihaye, V. Bruyère, and J.-F. Raskin. On Optimal Timed Strategies. In *FORMATS'05: Formal Modelling and Analysis of Timed Systems*, volume 3829 of *Lecture Notes in Computer Science*, pages 49–64. Springer-Verlag, 2005.

[BCFL04] P. Bouyer, F. Cassez, E. Fleury, and K.G. Larsen. Optimal strategies in priced timed game automata. In *FST&TCS'04: Foundations of Software Technology and Theoretical Computer Science*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160. Springer-Verlag, 2004.

[BDGP98] B. Bérard, V. Diekert, P. Gastin, and A. Petit. Characterization of the expressive power of silent transitions in timed automata. *Fundamenta Informaticae*, 36(2-3):145–182, 1998.

[BFH+01] G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. W. Vaandrager. Minimum-cost reachability for priced timed automata. In *HSCC'01: Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 147–161. Springer-Verlag, 2001.

[Hen96] T.A. Henzinger. The theory of hybrid automata. In *LICS'96: Logic in Computer Science*, pages 278–292. IEEE Computer Society Press, 1996.

[HHWT95] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. A user guide to HYTECH. In *TACAS'95: Tools and Algorithms for the Construction and Analysis of Systems*, volume 1019 of *Lecture Notes in Computer Science*, pages 41–71. Springer-Verlag, 1995.

[HKPV95] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *Proceedings of the 27th Annual Symposium on Theory of Computing*, pages 373–382. ACM Press, 1995.

[KPSY99] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Decidable integration graphs. *Information and Computation*, 150(2):209–243, 1999.

[LMM02] S. La Torre, S. Mukhopadhyay, and A. Murano. Optimal-reachability and control for acyclic weighted timed automata. In *IFIP TCS'02: Foundations of Information Technology in the Era of Networking and Mobile Computing*, volume 223 of *IFIP Conference Proceedings*, pages 485–497. Kluwer, 2002.

[LPY97] K.G. Larsen, P. Pettersson, and W. Yi. Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1-2):134–152, 1997.

[LR05] K. G. Larsen and J. I. Rasmussen. Optimal conditional reachability for multi-priced timed automata. In *FoSSaCS'05: Foundations of Software Science and Computational Structures*, volume 3441 of *Lecture Notes in Computer Science*, pages 234–249. Springer-Verlag, 2005.

[NW88] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1988.

[Ras05] J.-F. Raskin. An introduction to hybrid automata. In *Handbook of Networked and Embedded Control Systems*, pages 491–518. Birkhäuser, 2005.

[Roc70] R. T. Rockafellar. *Convex Analysis*. Princeton Univ. Press, New Jersey, 1970.