Optimal Infinite Scheduling for Multi-Priced Timed Automata

Patricia Bouyer[1⋆], Ed Brinksma[2], and Kim G. Larsen[3]

[1] LSV – CNRS & ENS de Cachan – UMR 8643 – France
Email: bouyer@lsv.ens-cachan.fr
[2] Department of Computer Science – University of Twente – The Netherlands
Email: brinksma@cs.utwente.nl
[3] BRICS – Aalborg University – Denmark
Email: kgl@cs.auc.dk

Abstract. This paper is concerned with the derivation of infinite schedules for timed automata that are in some sense optimal. To cover a wide class of optimality criteria we start out by introducing an extension of the (priced) timed automata model that includes both costs and rewards as separate modelling features. A precise definition is then given of what constitutes optimal infinite behaviours for this class of models. We subsequently show that the derivation of optimal non-terminating schedules for such double-priced timed automata is computable. This is done by a reduction of the problem to the determination of optimal mean-cycles in finite graphs with weighted edges. This reduction is obtained by introducing the so-called corner-point abstraction, a powerful abstraction technique of which we show that it preserves optimal schedules.

1 Introduction

In the past years the application of model-checking techniques to scheduling problems has become an established line of research. Scheduling problems can often be reformulated in terms of reachability, viz. as the (im)possibility to reach a state that improves on a given optimality criterion. Although there exists a wide body of literature and established results on (optimal) scheduling in the

⋆ This work has been mostly done while visiting CISS at Aalborg University in Denmark and has been supported by CISS and by ACI Cortos, a program of the french ministry of research.

fields of real-time systems and operations research, the model-checking approach is interesting on two accounts. First of all, it serves as a benchmarking activity in which the effectivity and efficiency of model-checking can be compared to the best known results obtained by other techniques. Second, most classical scheduling solutions have good properties only in the context of additional assumptions that may or, quite often, may not apply in actual practical circumstances. Here model-checking techniques have the advantage of offering a generic approach for finding solutions in a model, in much the same way that, say, numerical integration techniques may succeed where symbolic methods fail.

Of course, model-checking comes with its own restrictions and stumbling blocks, the most notorious being the state-space explosion. A lot of research, therefore, is devoted to the containment of this problem by sophisticated techniques, such as data structures for compact state space representation, smart state space search strategies, etc. An interesting idea for the model-checking of reachability properties that has received more attention recently is to somehow "guide" the exploration of the (symbolic) state space such that "promising" sets of states are visited first. In a number of applications [19, 20, 26, 14] model-checkers have been used to solve a number of non-trivial scheduling problems. Such approaches are different from classical, full state space exploration model-checking algorithms. They are used together with, for example, branch-and-bound techniques [7] to prune parts of the search tree that are guaranteed not to contain optimal solutions. This development has motivated research into the extension of model checking algorithms with optimality criteria. They provide a basis for the guided exploration of state spaces, and improve the potential of model-checking techniques for the resolution of scheduling problems. Work on extensions for application of the real-time model-checker Uppaal [24, 11] to optimal scheduling problems is reported in the articles [10, 9, 23]; related work is reported in [8, 6]. A closely related activity is reported in [1, 2], where specific search algorithms on timed automata models are defined to solve classes of scheduling problems, such as job-shop and task graph scheduling.

The formulation of scheduling synthesis as a reachability problem is not accurate in cases of reactive behaviours, where actually an infinite (optimal) schedule must be determined in case of reactive behaviours. In this case, not the (optimal) reachability of a good *final* state, but the reachability of good (optimal) infinite behaviours is relevant. Borrowing terminology from performance analysis, we can say that we are interested in the *stationary* behaviours of the system. In the discrete case, stationary behaviours are cyclic behaviours. Assuming cyclic behaviour the cost of reaching a cycle will be insignificant compared to the infinite cost related to non-terminating cyclic behaviours (assuming a single cycle execution has some positive cost). Approximating infinite behaviours by finite ones can yield good and even optimal solutions if it is possible to search sufficiently "deep", but costly pre-ambles may also obscure limit optimal behaviours [25].

In this paper we study optimal infinite behaviour in the context of *priced* timed automata[4]. In a discrete setting the detection of optimal behaviours goes back to Karp's algorithm [21], which determines the minimal mean cost of the cycles in a finite graph with weighted edges. Our contribution in this paper is that we show the computability of the corresponding symbolic question for priced timed automata using a reduction to a discrete problem *à la* Karp based on the so-called *corner-point abstraction*.

A second contribution is that we will not only establish computability of the problem in the original setting of priced timed automata, the model used in [10, 9, 6], but also in an extension that features two price parameters, viz. *costs* and *rewards*. This is motivated by the fact that the optimality of infinite behaviours is usually expressed as a limit ratio between accumulated costs and rewards. In practical terms they may involve measures such as units of money, production, consumption, time, energy, etc., as in throughput (units/time), production cost (units/money), efficiency (units/energy), etc. In principle all of such measures could count both as cost and reward depending on the particular problem. In this paper the difference between cost and reward is merely a technical one: for infinite behaviour we insist that accumulated rewards diverge (tend to positive infinity), whereas the accumulation of cost has no such constraint. Optimality is then interpreted as maximizing or minimizing the cost/reward ratio.

The structure of the rest of this paper is as follows. In section 2 we define double-priced transition systems, and on that basis introduce the model of double-priced timed automata. Section 3 states the main technical result of the paper together with the assumptions that must be made. Section 4 introduces the central notion of corner-point abstraction related to the region automaton construction for timed automata. Section 5 contains the proof of a necessary result,

[4] Called *linearly priced timed automata* in [10, 9, 23] and *weighted timed automata* in [6].

which states that quotients of affine functions over regions (and more generally zones) attain their extreme values in corner points. In section 6 we show the corner-point abstraction to be sound, and in section 7 to be complete w.r.t. optimal behaviours. In section 8, we deal with the complexity of the computation of optimal infinite schedules. In section 9, finally, we draw our conclusions and give indications for future work.

This paper is a long version of [12].

2 Models and Problems

2.1 Double-Priced Transition Systems

A *Double-Priced Transition System* (DPTS for short) is a tuple $(S, s_0, T, \mathsf{cost}, \mathsf{reward})$ where S is a set of states, $s_0 \in S$ is the initial state, $T \subseteq S \times S$ is the set of transitions, and $\mathsf{cost}, \mathsf{reward} : T \to \mathbb{R}$ are price functions. If (s, s') is a transition then $\mathsf{cost}(s, s')$ and $\mathsf{reward}(s, s')$ are two *prices* (the cost and the reward) associated with the transition (s, s'). We shall use the notation $s \to s'$ whenever $(s, s') \in T$, and $s \xrightarrow{c,r} s'$ whenever $(s, s') \in T$ with $\mathsf{cost}(s, s') = c$ and $\mathsf{reward}(s, s') = r$.

Let $\gamma = s_0 \to s_1 \cdots \to s_n$ be a finite execution of a DPTS $(S, s_0, T, \mathsf{cost}, \mathsf{reward})$. The price functions extend to γ in a natural way:

$$\mathsf{Cost}(\gamma) = \sum_{k=1}^{n} \mathsf{cost}(s_{k-1}, s_k) \quad \text{and} \quad \mathsf{Reward}(\gamma) = \sum_{k=1}^{n} \mathsf{reward}(s_{k-1}, s_k) \ .$$

Moreover, for a finite execution γ the *ratio* $\mathsf{Ratio}(\gamma)$ is defined as

$$\mathsf{Ratio}(\gamma) = \frac{\mathsf{Cost}(\gamma)}{\mathsf{Reward}(\gamma)}$$

if this quotient does exist (*i.e.* if $\mathsf{Reward}(\gamma) \neq 0$). Now consider an infinite execution $\mathit{\Gamma}$. Denote by $\mathit{\Gamma}_n$ the finite prefix of length n of $\mathit{\Gamma}$. The *ratio* of $\mathit{\Gamma}$ is defined as

$$\mathsf{Ratio}(\mathit{\Gamma}) = \lim_{n \to +\infty} \mathsf{Ratio}(\mathit{\Gamma}_n)$$

provided this limit exists. Otherwise, we consider the *infimum ratio* and the *supremum ratio* (denoted respectively as $\underline{\mathsf{Ratio}}$ and $\overline{\mathsf{Ratio}}$) defined by

$$\underline{\mathsf{Ratio}}(\mathit{\Gamma}) = \liminf_{n \to +\infty} (\mathsf{Ratio}(\mathit{\Gamma}_n)) \quad \text{and} \quad \overline{\mathsf{Ratio}}(\mathit{\Gamma}) = \limsup_{n \to +\infty} (\mathsf{Ratio}(\mathit{\Gamma}_n)) \ .$$

Given a DPTS $\mathcal{A}$, we define the *optimal ratio* $\mu^*_{\mathcal{A}}$ as

$$\mu^*_{\mathcal{A}} = \inf\{ \ \underline{\mathsf{Ratio}}(\Gamma) \ \mid \ \Gamma \text{ is an infinite execution of } \mathcal{A} \ \}$$

An infinite execution (also called *schedule*) $\Gamma^*_{\mathcal{A}}$ of $\mathcal{A}$ is *ratio-optimal* if $\mathsf{Ratio}(\Gamma^*_{\mathcal{A}}) = \mu^*_{\mathcal{A}}$. Note that a ratio-optimal run may not exist. In this case, we will say that $\left(\Gamma^{*,\varepsilon}_{\mathcal{A}}\right)_{\varepsilon>0}$ is a ratio-optimal family of runs whenever for every $\varepsilon > 0$, $|\underline{\mathsf{Ratio}}(\Gamma^{*,\varepsilon}_{\mathcal{A}}) - \mu^*_{\mathcal{A}}| < \varepsilon$.

The *optimal ratio problem* consists then in computing $\mu^*_{\mathcal{A}}$ and, if it does exist, $\Gamma^*_{\mathcal{A}}$, or a family $\left(\Gamma^{*,\varepsilon}_{\mathcal{A}}\right)_{\varepsilon>0}$.

Example 1. Consider a DPTS with states $\{A, B, C\}$ and transitions $A \xrightarrow{1,1} B$, $B \xrightarrow{1,0} B$, $B \xrightarrow{2,1} C$, $C \xrightarrow{1,0} B$, $C \xrightarrow{2,1} C$ and $C \xrightarrow{1,1} A$, and with A initial state. To see that the ratio is not always defined consider the execution $B \to C \to B^2 \to C^2 \to B^4 \to C^4 \to \cdots \to B^{2^n} \to C^{2^n} \cdots$. Computing ratios of finite prefixes, we get respectively

$$\begin{aligned} &\mathsf{Ratio}(B \to C \to B^2 \to C^2 \to \cdots \to B^{2^n}) = 3 \\ \text{whereas} \quad &\mathsf{Ratio}(B \to C \to B^2 \to C^2 \to \cdots \to B^{2^n} \to C^{2^n}) = 5 \end{aligned}$$

On the other hand, the execution consisting in an infinite repetition of the cycle $A \to B \to C \to A$ has a well-defined ratio, $\frac{4}{3}$, which is in fact the optimum ratio of the given DPTS. □

2.2 Double-Priced Timed Automata

For *finite-state* DPTSs the optimal ratio μ^* is obviously computable. Karp's Theorem [21] provides an algorithm with time complexity $\mathcal{O}(V.E)$ (V being the number of states and E the number of edges) in the case that the reward of each transition is 1. Extensions of Karp's algorithm have been proposed for computing μ^* in the general case, see for example [17, 18]. In the remainder of this paper we shall settle the computability of μ^* for *infinite-state* DPTS derived from so-called double-priced timed automata being timed automata extended with price(-*rates*) for determining cost and reward of discrete and delay transitions.

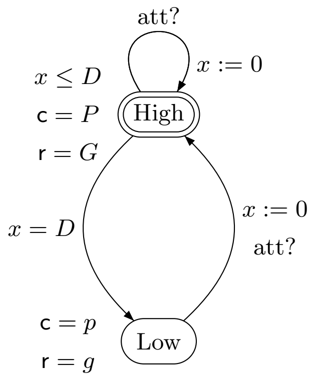
Given a set of clocks X, the set of clock constraints $\mathcal{C}(X)$ is defined inductively by the following rules:

$$g ::= x \bowtie c \ \mid \ g \wedge g$$

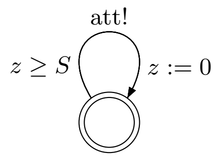
where $x \in X$, $c \in \mathbb{N}$ and $\bowtie \in \{<, \leq, =, \geq, >\}$.

Definition 1. *A* Double-Priced Timed Automaton *(DPTA for short) over a set of clocks X is a tuple $(L, \ell_0, E, I, \mathsf{c}, \mathsf{r})$, where L is a finite set of locations, ℓ_0 is the initial location, $E \subseteq L \times \mathcal{C}(X) \times 2^X \times L$ is the set of edges*[5]*, $I : L \longrightarrow \mathcal{C}(X)$ assigns invariants to locations and $\mathsf{c}, \mathsf{r} : (L \cup E) \longrightarrow \mathbb{Z}$ assign price-rates to locations and prices to edges.*

Example 2. Consider a production system consisting of a number of machines $M_1, \ldots M_n$ all attended to by a single operator O. Each machine M_i has two production modes: a *high* (H) and a *low* (L) mode, characterized by the amount of goods produced per time-unit (G respectively g) and the amount of power consumed per time-unit (P respectively p). From the producer's point of view the *high* production mode is preferable as it has a better (*i.e.* smaller) P/G-ratio than the *low* production mode. Unfortunately, each machine can only operate in the *high* production mode for a certain amount of time (D) without being attended to by the operator. The operator, in turn, needs a minimum time-separation (S) between attending machines. The figure on the right provides DPTA's for a typical machine and an operator.[6]In Fig. 1 we consider a production system obtained as the product of a machine M_1 with parameters $D = 3, P = 3, G = 4, p = 5, g = 2$, a machine M_2 with parameters $D = 6, P = 3, G = 2, p = 5, g = 2$ and a single operator with seperation time $S = 4$. In the product construction a cost (reward) rate of a composite location is obtained as sum of the cost (reward) rates of the corresponding component locations. □



Single machine
$M(D, G, P, g, p)$



Operator $O(S)$

The semantics of a DPTA is given as a DPTS. Intuitively, there are two types of transitions: delay transitions with cost and reward obtained by applying the

[5] In case $(\ell, g, Y, \ell') \in E$, we write $\ell \xrightarrow{g,Y} \ell'$.
[6] The cost and reward rates are both zero in the single location of the Operator.

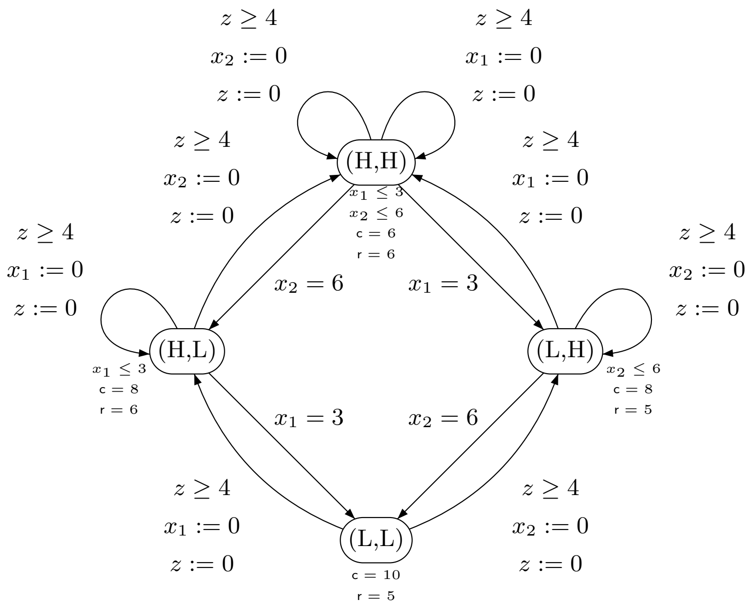


Fig. 1. Production System with Two Machines $M(D = 3, P = 3, G = 4, p = 5, g = 3)$ and $M(D = 6, P = 3, G = 2, p = 5, g = 2)$ and an Operator $O(4)$.

rates c and r of the source location, and discrete transitions with cost and reward given by the values of c and r of the corresponding edge. Before formally stating the semantics, we introduce a few definitions. A clock valuation $u \in \mathbb{R}_{\geq 0}^X$ is a function which assigns values to clocks. If $d \in \mathbb{R}_{>0}$ is a delay, then $u + d$ denotes the clock valuation such that for each clock x, $(u + d)(x) = u(x) + d$. If Y is a set of clocks then $[Y \leftarrow 0]u$ is the clock valuation u' with $u'(x) = 0$ if $x \in Y$ and $u'(x) = u(x)$ otherwise. Finally we write $u \models g$ if and only if the clock valuation u satisfies the guard g (defined in the natural way).

Definition 2. *The semantics of a DPTA* $\mathcal{A} = (L, \ell_0, T, I, \mathsf{c}, \mathsf{r})$ *over set of clocks* X *is the DPTS* $(S, s_0, \longrightarrow, \mathsf{cost}, \mathsf{reward})$ *over* X*, where* $S = L \times \mathbb{R}_{\geq 0}^X$*,* $s_0 = (\ell_0, \mathbf{0})$ *(where* $\mathbf{0}$ *is the valuation assigning* 0 *to each clock of* X*), and* $\longrightarrow$ *is defined as follows:*

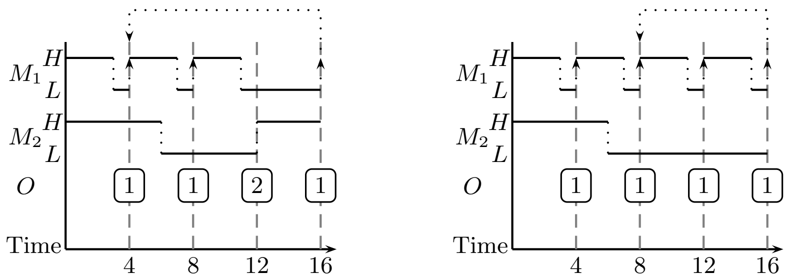
- $(\ell, u) \xrightarrow{c,r} (\ell, u + d)$ *if* $u + t \models I(\ell)$ *for every* $0 \leq t \leq d$*,* $c = \mathsf{c}(\ell) \cdot d$ *and* $r = \mathsf{r}(\ell) \cdot d$
- $(\ell, u) \xrightarrow{c,r} (\ell', u')$ *if there exists a transition* $\ell \xrightarrow{g,Y} \ell'$ *in* T *such that* $u \models g$*,* $u' = [Y \leftarrow 0]u$*,* $u' \models I(\ell')$*,* $c = \mathsf{c}\left(\ell \xrightarrow{g,Y} \ell'\right)$*, and* $r = \mathsf{r}\left(\ell \xrightarrow{g,Y} \ell'\right)$*.*

Example 3. Reconsider the Production System from Fig. 1. The following is an infinite execution providing a scheduling policy for the operator with the cost-reward ratio $96/66 \approx 1,455$:

$$\begin{aligned}
&((H,H), x_1 = x_2 = z = 0) \xrightarrow{\mathbf{18,18}} ((L,H), x_1 = x_2 = z = 3) \xrightarrow{\mathbf{8,5}} \\
&((L,H), x_1 = x_2 = z = 4) \longrightarrow ((H,H), x_1 = z = 0, x_2 = 4) \quad (*) \quad \xrightarrow{\mathbf{12,12}} \\
&((H,L), x_1 = z = 2, x_2 = 6) \xrightarrow{\mathbf{8,6}} ((L,L), x_1 = z = 3, x_2 = 7) \xrightarrow{\mathbf{10,5}} \\
&((L,L), x_1 = z = 4, x_2 = 8) \longrightarrow ((H,L), x_1 = z = 0, x_2 = 8) \xrightarrow{\mathbf{24,18}} \\
&((L,L), x_1 = z = 3, x_2 = 11) \xrightarrow{\mathbf{10,5}} ((L,L), x_1 = z = 4, x_2 = 12) \longrightarrow \\
&((L,H), x_1 = 4, x_2 = z = 0) \xrightarrow{\mathbf{32,20}} ((L,H), x_1 = 8, x_2 = z = 4) \\
&\longrightarrow ((H,H), x_1 = z = 0, x_2 = 4) \quad (*)
\end{aligned}$$

Fig. 2(a) illustrates this schedule as a Gantt chart. An other execution providing a scheduling policy with the cost-reward ratio $68/46 \approx 1,478$ is given in Fig. 2(b).

□



(a) Schedule with ratio 1,455

(b) Schedule with ratio 1,478

Fig. 2. Schedules for the Production System with ratios 1,455 and 1,478.

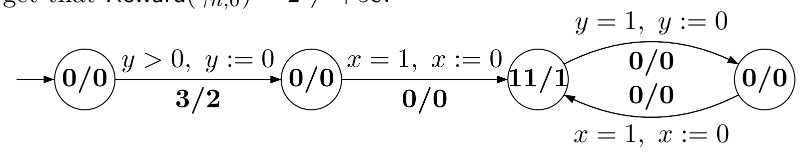
Remark. Let us point out several interesting subclasses of DPTAs. The reward will be said *impulse-based* whenever all reward-rates in locations are zero. This class corresponds roughly to the *mean* ratio as in classical finite-state systems [18]. An other interesting class is the one where the reward corresponds to the elapsing of time, that is when all location reward-rates are 1 and all transition rewards are 0. This last class corresponds to the usual intuitive notion of stationary behaviours where the measure is the cost by unit of time.

3 Result

Restrictions. In the remainder of this paper, we do several restrictions on the models we consider. We first restrict ourselves to reward functions that are non-negative. Though negative costs have *a priori* no real significance for modelling real systems, we will not restrict to non-negative costs, to get a more general result.

We also restrict ourselves to double-priced timed automata where the reward is strongly reward-diverging in the following sense: a DPTA $\mathcal{A}$ is *strongly reward-diverging* if, closing all the constraints of $\mathcal{A}$ (that is replacing in $\mathcal{A}$ each constraint $x < c$ by $x \leq c$ and each constraint $x > c$ by $x \geq c$), every infinite path Γ of the new *closed* automaton should satisfy that $\mathsf{Reward}(\Gamma) = +\infty$. Intuitively, the new closed automaton accepts all limit paths of the first automaton. The following example shows that this assumption is necessary if we want to have a link between the ratio of the limit of a sequence of paths and the limit of a sequence of ratios of paths, as we will do in the following. This restriction is close to the usual *non-zeno* hypothesis where an infinite number of actions can not be done in a finite amount of time. In our case, an infinite number of actions can not be done without some reward.

Example 4. The following DPTA (where $\mathbf{k/h}$ is a shortcut for $\mathsf{c} = k$ and $\mathsf{r} = h$) does not meet the previous restriction. Indeed consider the path $\gamma_{n,d}$ that takes the first transition at date d and then takes n times the loop. We have that $\mathsf{Reward}(\gamma_{n,d}) = 2 + d.n$. Thus, the ratio of any real infinite path is $+\infty$ (because for those states d is positive). Now, if we consider the infinite path where d is 0 (this path is a path of the automaton where all constraints have been closed), we get that $\mathsf{Reward}(\gamma_{n,0}) = 2 \neq +\infty$.



□

As we will see later, this restriction will have an important implication (*cf* Proposition 2).

Assumption for the following. We assume that timed automata are *bounded*, that is there exists a constant M such that for every reachable extended state (ℓ, v), for every clock x, $v(x) \leq M$. This is not a restriction as every DPTA can be transformed into an "equivalent" bounded timed automaton (strongly bisimilar and with the same costs and rewards).

We can now state the main result of this paper.

Theorem 1. *The optimal ratio problem is computable for strongly reward-diverging DPTAs with non-negative rewards.*

A more precise statement of the above theorem is obtained by notions of *soundness* and *completeness*. Given two DPTSs $\mathcal{S}$ and $\mathcal{S}'$ we say that $\mathcal{S}'$ is *sound* w.r.t $\mathcal{S}$ whenever $\mu^*_{\mathcal{S}'} \leq \mu^*_{\mathcal{S}}$ and we say that $\mathcal{S}'$ is *complete* w.r.t. $\mathcal{S}$ whenever $\mu^*_{\mathcal{S}} \leq \mu^*_{\mathcal{S}'}$. Theorem 1 is now a corollary of the following Proposition:

Proposition 1. *Let $\mathcal{A}$ be a bounded and strongly reward-diverging DPTA with non-negative rewards. Then there exists a finite-state DPTS $\mathcal{S}$ which is sound and complete w.r.t. the DPTS defined by $\mathcal{A}$.*

The finite-state DPTS we will prove sound and complete w.r.t. to a bounded DPTA $\mathcal{A}$ is the so-called *corner-point abstraction* of $\mathcal{A}$ that we define in the next section.

4 Regions and Corner-Point Abstraction

The aim of this section is to propose a discretization of timed automata behaviours based on an extension of the region automaton construction [4, 5]. We fix a DPTA $\mathcal{A} = (L, \ell_0, T, I, \mathsf{c}, \mathsf{r})$ and we assume that it is bounded by M. Moreover, we denote by k its number of clocks.

4.1 Regions and Corner-Points

In this paper, we will use the standard notion of regions, as initially defined by Alur and Dill [4]. As we consider only bounded timed automata, we only need bounded regions. A *region* (bounded by M) over a (finite) set of clocks X is a tuple $r = (h, [X_0, ..., X_p])$ where $h : X \longrightarrow \mathbb{N} \cap [0, M]$ assigns to each clock an integer value between 0 and M, p is some integer, and $(X_i)_{i=0,...,p}$ forms a partition of X such that for all $i > 0$, $X_i \neq \emptyset$ and $h(x) = M$ implies $x \in X_0$.

Given a valuation v, we say that v is in the region r whenever:

- for any clock $x \in X$, the integer part of $v(x)$ is $h(x)$,
- for any clock $x \in X$, $x \in X_0 \iff v(x) = h(x)$,
- for all clocks (x, y), $\{v(x)\} \leq \{v(y)\} \iff x \in X_i$ and $y \in X_j$ with $i \leq j$.

where $\{\cdot\}$ represents the fractional part.

A(n M-)*corner-point* is an element $\alpha = (a_j)_{1 \le j \le k}$ of $\mathbb{N}^k$ such that for every $1 \le j \le k$, $0 \le a_j \le M$. Let R be a region. A corner-point α is associated with R whenever it is in the closure of R (for the usual topology of $\mathbb{R}^k$). Let $r = (h, [X_0, ..., X_p])$ be a region. It has $p+1$ corner-points, $(\alpha_i)_{0 \le i \le p}$, α_i being defined by:

$$\alpha_i(x) = \begin{cases} h(x) & \text{if } x \in X_j \text{ with } j \le i \\ h(x)+1 & \text{if } x \in X_j \text{ with } j > i \end{cases}$$

4.2 Corner-Point Abstraction

We will construct a finite state DPTS $\mathcal{A}_{\mathsf{cp}} = (S, s_0, T', \mathsf{cost}, \mathsf{reward})$ called the *corner-point abstraction* of $\mathcal{A}$ where states (set S) are of the form (ℓ, R, α) with ℓ being a location, R a region and α a corner-point of R. Transitions (set T') of $\mathcal{A}_{\mathsf{cp}}$ are defined in the following manner:

Discrete transitions. If $e = \ell \xrightarrow{g,Y} \ell'$ is a transition of $\mathcal{A}$, there will be transitions $e' = (\ell, R, \alpha) \longrightarrow (\ell', R', \alpha')$ in $\mathcal{A}_{\mathsf{cp}}$ with $R \subseteq g$, $R' = [Y \leftarrow 0]R$, α corner-point associated with R, α' corner-point associated with R' and $\alpha' = [Y \leftarrow 0]\alpha$. We then set $\mathsf{cost}(e') = \mathsf{c}(e)$ and $\mathsf{reward}(e') = \mathsf{r}(e)$.

Idling transitions. There are two types of idling transitions.

- There are transitions $e' = (\ell, R, \alpha) \longrightarrow (\ell, R, \alpha')$ where α and α' are distinct corner-points of R and α' is the time successor of α (in which case, $\alpha' = \alpha + 1$). We then set $\mathsf{cost}(e') = \mathsf{c}(\ell)$ and $\mathsf{reward}(e') = \mathsf{r}(\ell)$ (intuitively the delay between the corner-points α and α' is one time unit).
- There are transitions $e' = (\ell, R, \alpha) \longrightarrow (\ell, R', \alpha)$ where R' is the time successor region of R and α is a corner-point associated with both R and R'. We then set $\mathsf{cost}(e') = 0$ and $\mathsf{reward}(e') = 0$ (intuitively, there is no delay between the corner-point of the two distinct regions).

The following proposition is an important consequence of the strongly reward-divergence hypothesis.

Proposition 2. *Let $\mathcal{A}$ be a bounded, strongly reward-diverging DPTA with non-negative rewards. Then there exist two constants $\lambda > 0$ and $\mu \ge 0$ such that for any infinite path Π of $\mathcal{A}_{\mathsf{cp}}$*

$$\mathsf{Reward}(\Pi_n) \ge \lambda.n - \mu$$

where Π_n denotes the prefix of length n of Π.

Note that the above λ and μ only depend on the automaton $\mathcal{A}$, not on the paths.

Proof. Let us consider an infinite path Π of $\mathcal{A}_{\mathsf{cp}}$. Let K denote the number of states of $\mathcal{A}_{\mathsf{cp}}$. Every factor of length $K+1$ of Π has a reward of at least 1, otherwise we could find in $\mathcal{A}_{\mathsf{cp}}$ a reachable cycle with a 0-reward, which would be in contradiction with the strongly reward-divergence hypothesis (this cycle corresponds to a real path in the automaton where all constraints have been closed). For each n, we note Π_n the prefix of length n of Π.
Thus, we get that $\mathsf{Reward}(\Pi_n) \geq \mathsf{int}(\frac{n}{K}) \geq \frac{1}{K}n - 1$ where $\mathsf{int}(\cdot)$ represents the integral part. $\square$

In everything which follows we will assume that DPTAs are strongly reward-diverging. However, the only property which we really need is the one expressed by Proposition 2. Hypotheses under which the results we present in this paper hold are thus decidable, as we only need to check that there is no cycle in the corner-point abstraction whose reward is 0. Note that the exact hypothesis can also be decided using the corner-point abstraction of the closed automaton.

Let $\gamma : (\ell_0, u_0) \longrightarrow (\ell_1, u_1) \longrightarrow \ldots$ be a (finite or infinite) path in $\mathcal{A}$. The set of all paths

$$\pi : (\ell_0, R_0, \alpha_{0,0}) \longrightarrow (\ell_0, R_0, \alpha_{0,1}) \ldots (\ell_0, R_0, \alpha_{0,p_0}) \longrightarrow (\ell_1, R_1, \alpha_{1,0}) \ldots$$

in $\mathcal{A}_{\mathsf{cp}}$ such that for every i, $u_i \in R_i$ and for every j, $\alpha_{i,j}$ is a corner-point associated with R_i is denoted $\mathsf{proj}_{\mathsf{cp}}(\gamma)$. Note that if $\gamma : (\ell_0, u_0) \longrightarrow (\ell_1, u_1) \longrightarrow \ldots$ and $\gamma' : (\ell_0, v_0) \longrightarrow (\ell_1, v_1) \longrightarrow \ldots$ are two "region-equivalent" real-paths (*i.e.* for every i, u_i and v_i are region-equivalent), then $\mathsf{proj}_{\mathsf{cp}}(\gamma) = \mathsf{proj}_{\mathsf{cp}}(\gamma')$.

In the remainder of the paper, we will prove that the optimal ratio of the corner-point abstraction is the same as the optimal ratio of the original DPTA. As the corner-point abstraction can be effectively constructed and as computing optimal ratios in finite-state DPTSs (the corner-point abstraction is a finite-state DPTS) is effective (see for example [21, 17, 18]), we get that $\mu^*_{\mathcal{A}}$ is effectively computable for DPTAs $\mathcal{A}$ satisfying the strongly reward-divergence hypothesis.

Example 5. If we come back to the automaton of Example 4, as we have already seen, it does not meet the strongly reward-divergence restriction. It is easy to compute that for any real infinite path Γ_d (where d denotes the date the first transition is taken), $\mathsf{Ratio}(\Gamma_d) = 11$. However if we consider the path Π of the corner-point abstraction where d would be 0, we get that $\mathsf{Ratio}(\Pi) = \frac{3}{2}$. We see that we could change the costs and rewards on the transitions, and we would get that there is no relation between the ratio of paths in the original automaton and ratio of paths in the corner-point abstraction. This shows that strongly reward-divergence is somehow necessary. □

5 Quotient of Affine Functions

This section contains technical results that will be useful in the following. Let A be a closed set of $\mathbb{R}^n$ (with $n \geq 1$). The *border* of A is denoted by $\mathsf{Border}_n(A)$ and is defined as $A \setminus \mathring{A}$ where $\mathring{A}$ denotes the interior of A. Let A be a closed set and x a point in $\mathbb{R}^n$. The following statements are equivalent and characterize the border of A:

- $x \in \mathsf{Border}_n(A)$
- $x \in A$ and for every $\varepsilon > 0$, there exists $y \notin A$ such that $\|x - y\|_\infty < \varepsilon$. [7]

Lemma 1. *Let f be a function defined on a compact set $A \subset \mathbb{R}^n$ (where $n \geq 1$) such that*

$$f(x_1, ..., x_n) = \frac{\sum_{i=1}^n c_i x_i + c}{\sum_{i=1}^n r_i x_i + r}$$

Then the minimum of f on A is obtained on the border of A.

Proof. The proof will be done by induction on n, the dimension of the space.

Initial step. It corresponds to the case where $n = 1$. Assume that $f(x) = \frac{ax+b}{cx+d}$ with $c \neq 0$. The function f is defined on the set $\mathbb{R} \setminus \left\{\frac{-d}{c}\right\}$ and we can compute the derivative f' of f over this set:

$$f'(x) = \frac{a(cx + d) - (ax + b)c}{(cx + d)^2} = \frac{ad - bc}{(cx + d)^2}$$

The sign of f' is constant on both $]-\infty, \frac{-d}{c}[$ and $]\frac{-d}{c}, +\infty[$. The minimum of f over A is thus obtained in one of the points $\min(A^-)$, $\max(A^-)$, $\min(A^+)$, and $\max(A^+)$, where $A^- = A \cap]-\infty, \frac{-d}{c}[$, $A^+ = A \cap]\frac{-d}{c}, +\infty[$, $\min(B)$ is the smallest

[7] $\|.\|_\infty$ represents the usual infinite norm defined as $\|(x_i)_{i=1...n}\|_\infty = \max\{|x_i| \mid i = 1...n\}$.

point of B and $\max(B)$ is the largest point in B. In all cases, this minimum is obtained on the border of A.

Note that if $ad - bc = 0$, then it means that f is contant, and thus the minimum can in particular be obtained on the border of A.

Assume that $f(x) = ax + b$. Similarly, we prove that the minimum of f is obtained on the border of A.

Inductive step. Assume $n > 1$. As A is compact, the minimum of f on A is obtained for some value α for x_1. Now, assume that

$$g_\alpha(x_2, ..., x_n) = f(\alpha, x_2, ..., x_n)$$

[α can be viewed as a parameter]

Fix an α such that $A \cap (x_1 = \alpha) \neq \emptyset$. The function g_α is defined on a subset of $\mathbb{R}^{n-1}$. We denote by B_α the projection of $A \cap (x_1 = \alpha)$ onto the last $n-1$ coordinates, B_α is a compact set of $\mathbb{R}^{n-1}$. Note also that g_α has the same form as the function f. We know from induction hypothesis that $\min_{B_\alpha} g_\alpha$ is obtained on $\mathsf{Border}_{n-1}(B_\alpha)$. To get the induction step, it is then sufficient to prove that if $(x_2, ..., x_n)$ is in $\mathsf{Border}_{n-1}(B_\alpha)$, then $(\alpha, x_2, ..., x_n)$ is in $\mathsf{Border}_n(A)$.

Pick $(x_2, ...x_n)$ in $\mathsf{Border}_{n-1}(B_\alpha)$ and $\varepsilon > 0$. Then $(x_2, ..., x_n) \in B_\alpha$ and there exists $(y_2, ..., y_n) \notin B_\alpha$ such that $\|(x_2, ..., x_n) - (y_2, ..., y_n)\|_\infty < \varepsilon$. B_α is the projection of $A \cap x_1 = \alpha$ onto the $n-1$ last coordinates, thus $(\alpha, x_2, ..., x_n) \in A \cap x_1 = \alpha$ and $(\alpha, y_2, ..., y_n) \notin A \cap (x_1 = \alpha)$. However, $\|(\alpha, x_2, ..., x_n) - (\alpha, y_2, ..., y_n)\|_\infty = \|(x_2, ..., x_n) - (y_2, ..., y_n)\|_\infty < \varepsilon$.

We conclude that $(\alpha, x_2, ..., x_n)$ is in $\mathsf{Border}_n(A)$. $\square$

In the remainder of the section, we will use the standard notion of zone. A *zone* over the set of clocks X is a **convex** set of valuations defined by constraints of the forms $x \bowtie c$ and $x - y \bowtie c$ where x and y are in X, $\bowtie \in \{\leq, <, =, >, \geq\}$ and c is an integer. For example, the constraints $\{x \leq 3,\ y \geq 4, x - y < -5\}$ represent the set of valuations v such that $v(x) \leq 3$, $v(y) \geq 4$ and $v(x) - v(y) < -5$.

Lemma 2. *Let f be a function defined on a bounded zone $Z \subset \mathbb{R}^n$ (where $n \geq 1$) by*

$$f(x_1, ..., x_n) = \frac{\sum_{i=1}^n c_i x_i + c}{\sum_{i=1}^n r_i x_i + r}$$

We assume in addition that $\overline{Z}$ (the closure of Z for the usual topology) is included in the definition set of f. Then the infimum of f on Z is obtained on a point of $\overline{Z}$ with integer coordinates.

Proof. As Z is a bounded zone, $\overline{Z}$ is a compact set of $\mathbb{R}^n$. We have thus that $\inf_Z f = \min_{\overline{Z}} f$. The rest of the proof is done by induction on the dimension n of the space.

Initial step. The initial step is for $n = 1$. Zones in one dimension are lines (a, b) where a, b are integers and (,) are either [or]. As it has been done in the proof of Lemma 1, the minimum of f on $[a, b]$ is obtained in a or b, which are integers.

Induction step. We assume $n > 1$. Applying Lemma 1, we get that $\min_{\overline{Z}} f$ is obtained on $\mathsf{Border}_n(\overline{Z})$. Recall that $\overline{Z}$ can be obtained from Z by replacing the constraints $x - y < c$ by $x - y \leq c$ and that $\mathsf{Border}_n(\overline{Z})$ corresponds to the union of all the facets of $\overline{Z}$.[8]

The infimum of f on Z is thus on a facet whose equation is $x - y = c$ (resp. $x = c$). We eliminate variable x in f by replacing x with $y + c$ (resp. c) and we get a function g which has $n - 1$ variables. Without loss of generality we can assume that $x = x_1$. We then use the property that $\min_{\overline{Z}} f = \min_{Z'} g$ where $Z' = \mathsf{proj}_{(x_2,\dots,x_n)}(\overline{Z} \cap x_1 = y + c)$ (resp. $Z' = \mathsf{proj}_{(x_2,\dots,x_n)}(\overline{Z} \cap x_1 = c)$).

We know by induction hypothesis that the minimum of g is obtained with each x_i $(i > 1)$ having integer values. Thus, the minimum of f is obtained with $x_1 = y+c$ (resp. $x_1 = c$) which is an integer and all other clocks also have integer values. □

The two lemmas above together imply that the infimum of such a function f on a bounded zone Z is obtained in one of the corner-points of the zone. The result could be easily generalized to general bounded convex polyhedra, and not only bounded zones. The result would then be that the function f is minimized in one of the corner-points of the polyhedron, a corner-point representing intuitively an extremal point.

[8] A *facet* of a closed zone $\overline{Z}$ is an intersection $Z \cap (x = c)$ (or $Z \cap (x - y = c)$) where $x\{\leq, \geq\}c$ (or $x - y\{\leq, \geq\}c$) is a constraint, as tight as possible, defining Z, see [23] for more details.

6 Soundness of the Corner-Point Abstraction

The aim of this section is to prove that the corner-point abstraction is sound, that is for all the infinite paths in the timed automaton, we can find an infinite path in the corner-point abstract automaton with a smaller ratio. The proof will be done in two steps: first, we will consider **finite** paths, and then we will extend the result to **infinite** paths.

Theorem 2. *Let $\mathcal{A}$ be a bounded, strongly reward-diverging DPTA with non-negative rewards. Then, $\mu^*_{\mathcal{A}_{\mathsf{cp}}} \leq \mu^*_{\mathcal{A}}$.*

6.1 Considering finite paths

Proposition 3. *Let $\mathcal{A}$ be a bounded, stronly reward-diverging DPTA and let γ be a* ***finite*** *execution in $\mathcal{A}$. Then there exists an execution $\pi \in \mathsf{proj}_{\mathsf{cp}}(\gamma)$ such that*

$$\mathsf{Ratio}(\pi) \leq \mathsf{Ratio}(\gamma) \ .$$

The special case where the reward is impulse-based may be obtained as a direct consequence of previous works on cost-optimality in timed automata (*cf* for example [10, 9, 23]). The general case however requires a new proof. It will require the technical results developed in section 5.

Proof. Let $\gamma = (\ell_0, u_0) \longrightarrow (\ell_0, u_0 + d_0) \longrightarrow (\ell_1, u_1) \longrightarrow (\ell_1, u_1 + d_1) \cdots \longrightarrow (\ell_n, u_n)$ be a finite execution in $\mathcal{A}$ (with alternating delay and discrete transitions). We set for any $1 \leq i \leq n$, $t_i = \sum_{0 \leq j < i} d_j$. We moreover assume that this execution is read on the sequence of transitions $\ell_0 \xrightarrow{g_1, Y_1} \ell_1 \cdots \xrightarrow{g_n, Y_n} \ell_n$ in $\mathcal{A}$. The ratio of γ is:

$$f(t_1, ..., t_n) = \frac{\sum_{i=1}^{n} c_i(t_i - t_{i-1}) + c}{\sum_{i=1}^{n} r_i(t_i - t_{i-1}) + r}$$

where c_i, r_i are the cost and reward of the transition $\ell_{i-1} \xrightarrow{g_i, Y_i} \ell_i$ and c, r are the sum of all the discrete costs and rewards along γ.

We want to minimize this function with the constraints that for all i, $v'_i \in R_i$ where:

- $v'_i(x) = t_i - t_j$ where $j = \max\{k \leq i \mid x \in Y_k\}$
- R_i is the region to which belongs v_i

The set of constraints $\{v'_i \in R_i \mid i = 1...n\}$ defines a zone Z on the variables $(t_i)_{i=1...n}$. We can apply Lemma 2 and we get that the infimum of f on Z is obtained in (at least) a point with integer coordinates, say $(\alpha_i)_{i=1...n}$. Note that this point is in the closure of Z, and thus that it satisfies in particular the set of constraints $\{v'_i \in \overline{R_i} \mid i = 1...n\}$.

We define the valuations $(\sigma_i)_{i=1...n}$ by $\sigma_i(x) = \alpha_i - \alpha_j$ where $j = \max\{k \leq i \mid x \in Y_k\}$. Each valuation σ_i is in $\overline{R_i}$ and has integer coordinates. It is thus a corner-point of R_i. Moreover, the sequence of valuations $(\sigma_i)_i$ would be an accepted sequence if we replace the constraints R_i by $\overline{R_i}$. In addition, the time elapsed in each state ℓ_i would then be $\alpha_{i+1} - \alpha_i$.

We are now ready to build a path in the corner-point abstraction which will meet our requirements. For any i, there exist regions $(R_{i,j})_{j=0...n_i}$ for a given index n_i such that:

- $R_{i,0} = R_i$
- $R_{i,j+1}$ is the time successor region of $R_{i,j}$
- $R_{i+1} = [Y_i \leftarrow 0]R_{i,n_i}$

We construct a graph. Transitions are:

Initial idlings.
- $(\ell_i, R_{i,0}, \sigma_i) \longrightarrow (\ell_i, R_{i,1}, \sigma_i)$ if σ_i is also a corner-point of $R_{i,1}$
- $(\ell_i, R_{i,0}, \sigma_i) \longrightarrow (\ell_i, R_{i,0}, \sigma_i + 1)$ if $\sigma_i + 1$ is a corner-point of $R_{i,0}$

and then inductively:

Idlings. We assume $(\ell_i, R_{i,j}, \beta)$ has already been constructed.
- $(\ell_i, R_{i,j}, \beta) \longrightarrow (\ell_i, R_{i,j+1}, \beta)$ if $0 < j < n_i$ and β is a corner-point of $R_{i,j+1}$
- $(\ell_i, R_{i,j}, \beta) \longrightarrow (\ell_i, R_{i,j}, \beta + 1)$ if $0 < j \leq n_i$, $\beta + 1$ is a corner-point of $R_{i,j}$ and $[Y_{i+1} \leftarrow 0]\beta \neq \sigma_{i+1}$.

Discrete actions. We assume $(\ell_i, R_{i,n_i}, \beta)$ has already been constructed.
- $(\ell_i, R_{i,n_i}, \beta) \longrightarrow (\ell_{i+1}, R_{i+1,0}, [Y_{i+1} \leftarrow 0]\beta)$ if $i < n$ and $[Y_{i+1} \leftarrow 0]\beta = \sigma_{i+1}$.

Claim. The graph we just constructed is a path of $\mathcal{A}_{\mathsf{cp}}$. We denote it by π.

Proof. For any computed $(\ell, R, \alpha) \neq (\ell_n, R_{n,n_n}, \sigma_n)$, there is exactly one possible successor in the graph above. ♦

Claim. The path π is in $\mathsf{proj_{cp}}(\gamma)$.

Proof. The only thing we need to prove is that for each i, we will always go through a state $(\ell_i, R_{i,n_i}, \beta)$ with $[Y_{i+1} \leftarrow 0]\beta = \sigma_{i+1}$.

First note that there exists t such that $\sigma_{i+1} = [Y_{i+1} \leftarrow 0](\sigma_i + t)$. Moreover, $\sigma_{i+1} \in \overline{R_{i+1}} = [Y_i \leftarrow 0]R_{i,n_i}$, $\sigma_i \in \overline{R_i}$ and R_{i,n_i} is some time successor of R_i (which implies that $\overline{R_{i,n_i}}$ is also some time successor of $\overline{R_i}$). Thus $\sigma_i + t \in \overline{R_{i,n_i}}$. All the transitions between states $(\ell_i, R_{i,j}, \beta)$ are such that β is a time successor of σ_i which is in $\overline{R_{i,j}}$. Thus, after having gone through all the regions $R_{i,j}$, we will arrive in $\sigma_i + t$, viewed as a corner-point of R_{i,n_i}. ♦

We can now compute the ratio of π:

$$\mathsf{Ratio}(\pi) = \frac{\sum_{i=1}^{n} c_i(\alpha_i - \alpha_{i-1}) + c}{\sum_{i=1}^{n} r_i(\alpha_i - \alpha_{i-1}) + r} = f(\alpha_1, ..., \alpha_n)$$

(using the costs and rewards as defined in the construction of $\mathcal{A}_{\mathsf{cp}}$).

As $(\alpha_i)_{i=1...n}$ minimizes f over Z, we get that $\mathsf{Ratio}(\pi) \leq \mathsf{Ratio}(\gamma)$ and we are done. □

6.2 Extension to Infinite Paths

We will now prove that the previous property, restricted to finite executions, can be extended to infinite executions.

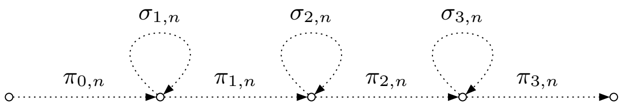
Proposition 4. *Let $\mathcal{A}$ be a bounded, strictly reward-diverging DPTA with non-negative rewards, and let Γ be a infinite path in $\mathcal{A}$. Then, there exists an infinite path Π in $\mathcal{A}_{\mathsf{cp}}$ such that*

$$\mathsf{Ratio}(\Pi) \leq \underline{\mathsf{Ratio}}(\Gamma) \quad . \tag{$\star$}$$

Notice that, on the contrary to Proposition 3 the path Π may not be in $\mathsf{proj_{cp}}(\Gamma)$. In addition, for any finite prefix γ of Γ, it may happen that no finite prefix of Π satisfies the property described in Proposition 3, which means that we will not solve the problem just by extending paths given by Proposition 3.

Proof. Let $\Gamma : (\ell_0, u_0) \longrightarrow (\ell_1, u_1) \ldots$ be an infinite path in $\mathcal{A}$. In the following, we will denote by Γ_n the prefix of length n of Γ.

Let α be the value of the minimal ratio for a reachable cycle in $\mathcal{A}_{\mathsf{cp}}$. Let n be an integer. From Proposition 3, there exists a path Π_n in $\mathsf{proj}_{\mathsf{cp}}(\Gamma_n)$ such that $\mathsf{Ratio}(\Pi_n) \leq \mathsf{Ratio}(\Gamma_n)$. Using Proposition 2, we get that $\mathsf{Reward}(\Pi_n) \in \Omega(n)$[9], which implies in particular that $\lim_{n\to+\infty} \mathsf{Reward}(\Pi_n) = +\infty$.



We decompose Π_n into cycles, *i.e.* we write $\Pi_n = \pi_{0,n}.\sigma_{1,n}.\pi_{1,n}\ldots\sigma_{p_n,n}.\pi_{p_n,n}$ where $\pi_{i,n}$ are simple paths and $\sigma_{i,n}$ are cycles (see the figure above). We assume in addition that this decomposition is maximal in the sense that the path $\pi_{0,n}.\pi_{1,n}\ldots\pi_{p_n,n}$ is acyclic. The maximality property of our decomposition implies that the total length of $\pi_{0,n}.\pi_{1,n}\ldots\pi_{p_n,n}$ is less than the number of nodes in $\mathcal{A}_{\mathsf{cp}}$.

We set $C(n) = \sum_{i=0}^{p_n} \mathsf{Cost}(\pi_{i,n})$ and $R(n) = \sum_{i=0}^{p_n} \mathsf{Reward}(\pi_{i,n})$ and we compute now the difference between $\mathsf{Ratio}(\Pi_n)$ and α:

$$\begin{aligned}\mathsf{Ratio}(\Pi_n) - \alpha &= \frac{\sum_{i=1}^{p_n} \mathsf{Cost}(\sigma_{i,n}) + C(n)}{\sum_{i=1}^{p_n} \mathsf{Reward}(\sigma_{i,n}) + R(n)} - \alpha \\ &= \frac{\frac{\sum_{i=1}^{p_n} \mathsf{Cost}(\sigma_{i,n})}{\sum_{i=1}^{p_n} \mathsf{Reward}(\sigma_{i,n})} + \frac{C(n)}{\sum_{i=1}^{p_n} \mathsf{Reward}(\sigma_{i,n})}}{1 + \frac{R(n)}{\sum_{i=1}^{p_n} \mathsf{Reward}(\sigma_{i,n})}} - \alpha\end{aligned}$$

We set $\beta(n) = \frac{\sum_{i=1}^{p_n} \mathsf{Cost}(\sigma_{i,n})}{\sum_{i=1}^{p_n} \mathsf{Reward}(\sigma_{i,n})}$ and we have that $\beta(n) \geq \alpha$ because α is the ratio of the minimal reachable cycle.[10] We get that

$$\mathsf{Ratio}(\Gamma_n) - \alpha \quad\geq\quad \mathsf{Ratio}(\Pi_n) - \alpha \quad=\quad \frac{\beta(n) - \alpha + \frac{C(n) - \alpha R(n)}{\sum_{i=1}^{p_n} \mathsf{Reward}(\sigma_{i,n})}}{1 + \frac{R(n)}{\sum_{i=1}^{p_n} \mathsf{Reward}(\sigma_{i,n})}} \qquad (\star\star)$$

Observe now that $R(n)$ and $C(n)$ are bounded and that $\lim_{n\to+\infty} \sum_{i=0}^{p_n} \mathsf{Reward}(\sigma_{i,n}) = +\infty$. We can now take the infimum limit of Equation $(\star\star)$, and we get:

$$\underline{\lim}_{n\to+\infty}(\mathsf{Ratio}(\Gamma_n)) - \alpha \geq \underline{\lim}_{n\to+\infty}\beta(n) - \alpha \geq 0$$

Hence, the infimum ratio of Γ is greater than the ratio of the optimal reachable cycle in $\mathcal{A}_{\mathsf{cp}}$. Hence we can take Π the path of $\mathcal{A}_{\mathsf{cp}}$ which consists first in reaching

[9] We recall that a function f over integers is in $\Omega(n)$ if there exists some positive κ and some integer n_0 such that for all $n \geq n_0$, $f(n) \geq \kappa.n$.

[10] Recall the property that if $b > 0$ and $d > 0$, then $\min\left(\frac{a}{b}, \frac{c}{d}\right) \leq \frac{a+c}{b+d} \leq \max\left(\frac{a}{b}, \frac{c}{d}\right)$.

the optimal reachable cycle and then cycling along this cycle. This path meets the requirement of the proposition. □

7 Completeness of the Corner-Point Abstraction

The aim of this section is to state the completeness of the corner-point abstraction. More precisely, we will prove that for every infinite path of the corner-point abstraction, there are real paths in the original automaton whose ratio is as close as we want to the ratio of the given path in the corner-point abstraction.

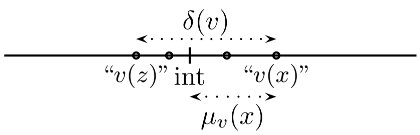
Theorem 3. *Let $\mathcal{A}$ be a bounded, strongly reward-diverging DPTA with non-negative rewards. Then,* $\mu^*_{\mathcal{A}} \leq \mu^*_{\mathcal{A}_{\mathsf{cp}}}$.

The proof of this theorem will be done in two steps: we will first prove that we can approximate paths in $\mathcal{A}_{\mathsf{cp}}$ by paths in $\mathcal{A}$ which are as close as we want to the original path (proposition 5). It will then be sufficient to prove that for each infinite path in $\mathcal{A}_{\mathsf{cp}}$, under the strongly reward-divergence assumption, we can find a real path in $\mathcal{A}$ whose ratio is as close as we want to the ratio of the given path in $\mathcal{A}_{\mathsf{cp}}$ (proposition 6).

Proposition 5. *Let $\mathcal{A}$ be a bounded DPTA. Let $\pi : (\ell_0, R_0, \alpha_0) \longrightarrow \cdots (\ell_n, R_n, \alpha_n) \cdots$ be a (possibly infinite) path in $\mathcal{A}_{\mathsf{cp}}$. Let $0 < \varepsilon < \frac{1}{2}$. There exists a real path $\gamma_\varepsilon : (\ell_0, u_0) \longrightarrow \cdots (\ell_n, u_n) \cdots$ in $\mathcal{A}$ such that $u_i \in R_i$ and $\|u_i - \alpha_i\|_\infty < \varepsilon$ for every i.*

Proof. Let v be a valuation. For any clock x, we define $\mu_v(x) = \min\{|v(x) - p| \mid p \text{ integer}\}$ and for any pair of clocks (x, y), $\nu_v(x, y) = \min\{|v(x) - v(y) - p| \mid p \text{ integer}\}$. We define the *diameter* of v as

$$\delta(v) = \max\left(\{\mu_v(x) \mid x \text{ clock}\} \cup \{\nu_v(x, y) \mid x, y \text{ clocks}\}\right)$$

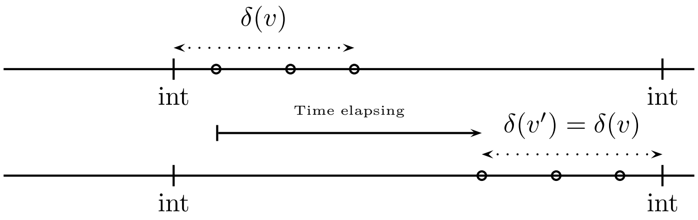


Proposition 5 will be a direct consequence of the following technical lemma.

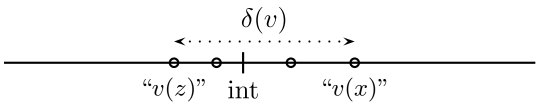
Lemma 3. *Consider a transition $(\ell, R, \alpha) \longrightarrow (\ell', R', \alpha')$ in $\mathcal{A}_{\mathsf{cp}}$, take a valuation $v \in R$ such that $\delta(v) < \varepsilon$ and $|v(x) - \alpha(x)| = \mu_v(x)$. There exists a valuation $v' \in R'$ such that $(\ell, v) \longrightarrow (\ell', v')$ in $\mathcal{A}$, $\delta(v') < \varepsilon$ and $|v'(x) - \alpha'(x)| = \mu_{v'}(x)$.*

Proof. We prove this lemma by distinguishing all the possible cases for a transition:

- Assume $(\ell, R, \alpha) \longrightarrow (\ell', R', \alpha')$ is a discrete transition which comes from some $\ell \xrightarrow{g,Y} \ell'$ of $\mathcal{A}$. We thus have that $R' = [Y \leftarrow 0]R$ and $\alpha' = [Y \leftarrow 0]\alpha$. Let us define v' as $[Y \leftarrow 0]v$. Obviously $v' \in R'$. If $x \in Y$, $v'(x) = \alpha'(x) = 0$, thus $0 = \mu_{v'}(x) \leq \mu_v(x)$. If $x \notin Y$, $v'(x) = v(x)$ and thus $\mu_{v'}(x) = \mu_v(x)$. If $x, y \notin Y$, $\nu_{v'}(x, y) = \nu_v(x, y)$. If $x, y \in Y$, $\nu_{v'}(x, y) = 0$. If $x \in Y$ but $y \notin Y$, $\nu_{v'}(x, y) = \mu_v(y)$. Thus, we get that $\delta(v') \leq \delta(v) < \varepsilon$. The second property can be obtained easily.
- Assume $(\ell, R, \alpha) \longrightarrow (\ell, R, \alpha + 1)$ is an idling transition. In that case, we have that for every x, $\{v(x)\} = \mu_v(x) > 0$ (and also that $\nu_v(x, y) = |\{v(x)\} - \{v(y)\}|$). We set $v' = v + (1 - \max_x(\mu_v(x)) - \min_y(\mu_v(y)))$. Let us illustrate this construction:

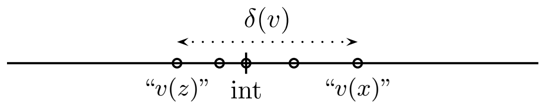
  

  It is then easy to prove that $v' \in R'$, $\delta(v') = \delta(v) < \varepsilon$ and for each clock x, $\mu_{v'}(x) = |v'(x) - \alpha'(x)|$ (this is because ε has been taken as strictly less than $\frac{1}{2}$).
- Assume $(\ell, R, \alpha) \longrightarrow (\ell, R', \alpha)$ is an idling transition where R' is the immediate time successor of R. There are several cases:
  - the integer is "inside" the valuations and no value $v(x)$ is an integer

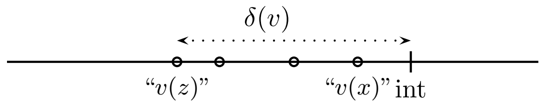
    We then set y as being the one such that $1 - \{v(y)\}$ is positive and minimal. We then define v' as $v'(x) = v(x) + 1 - \{v(y)\}$. Of course, $\delta(v') = \delta(v) < \varepsilon$, $v' \in R'$ and $\mu_{v'}(x) = \mu_v(x) + 1 - \{v(y)\} = v(x) - \alpha(x) + 1 - \{v(y)\} = v'(x) - \alpha'(x)$ if $\{v(x)\} < \frac{1}{2}$ and $\mu_{v'}(x) = \mu_v(x) - (1 - \{v(y)\}) = \alpha(x) - v(x) - 1 + \{v(y)\} = \alpha'(x) - v'(x)$ otherwise.

- the integer is “inside” the valuations and the set Z of clocks x such that $v(x)$ is an integer is non empty.



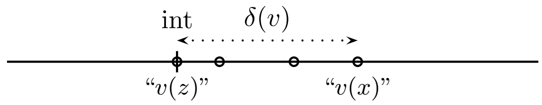
We then set y as being the one such that $1 - \{v(y)\}$ is positive and minimal. We then define v' as $v'(x) = v(x) + \frac{1-\{v(y)\}}{2}$. As previously, it is easy to get that $\delta(v') = \delta(v) < \varepsilon$, $v' \in R'$, and $|\alpha'(x) - v'(x)| = \mu_{v'}(x)$ for every clock x.

- the integer is on the right of the valuations



We set y the clock such that $1 - \{v(y)\}$ is minimal and we define v' such that for every clock x, $v'(x) = v(x) + 1 - \{v(y)\}$. We get that $\delta(v') = \delta(v) - (1 - \{v(y)\}) < \delta(v) < \varepsilon$. The rest is as previously.

- the integer is at the left corner of the valuations



We choose $d > 0$ such that $\delta(v) + d < \varepsilon$ and we set v' such that for each clock x, $v'(x) = v(x) + d$. In that case, we have $\delta(v') = \delta(v) + d < \varepsilon$. And everything else is as before.

In all cases, we are done and the proof is finished. ♦

Using this lemma, we construct inductively a path γ_ε as described above, at each step of the construction we have that $\|v_i - \alpha_i\|_\infty \leq \delta(v_i) < \varepsilon$. This concludes the proof. □

We now use this result on paths to prove the following proposition on ratios.

Proposition 6. *Let $\mathcal{A}$ be a bounded, strongly reward-diverging DPTA with non-negative rewards. Let Π be an infinite path in $\mathcal{A}_{\mathsf{cp}}$ such that $\mathsf{Ratio}(\Pi)$ is defined. Then the following holds: for any $\varepsilon > 0$, there exists a real path Γ^ε such that $|\mathsf{Ratio}(\Pi) - \underline{\mathsf{Ratio}}(\Gamma^\varepsilon)| < \varepsilon$.*

Note that in case we have only non-strict constraints along the path accepting Γ in $\mathcal{A}$, Π corresponds to a real path in $\mathcal{A}$, it thus corresponds to $\Gamma^*_{\mathcal{A}}$. Otherwise, the paths constructed in the following will give us an optimal family $\left(\Gamma^{*,\varepsilon}_{\mathcal{A}}\right)_{\varepsilon>0}$ of schedules.

Proof. Take $\varepsilon > 0$. Applying the previous lemma, we can find a real path Γ^ε such that at each step, the distance between the corner-point and the real valuation is strictly less than ε. We have that $\mathsf{Cost}(\Gamma^\varepsilon_n) = \mathsf{Cost}(\Pi_n) + \sum_{i=1}^n \sigma_i$ where each σ_i can be bounded by $2C\varepsilon$ where C is the maximal cost labeling the automaton $\mathcal{A}_{\mathsf{cp}}$. Similarly, we can write $\mathsf{Reward}(\Gamma^\varepsilon_n) = \mathsf{Reward}(\Pi_n) + \sum_{i=1}^n \tau_i$ where each τ_i can be bounded by $2C\varepsilon$. From Proposition 2, we have that there exists λ and μ such that $\mathsf{Reward}(\Pi_n) \geq \lambda n - \mu$. Compute

$$
\begin{aligned}
|\mathsf{Ratio}(\Gamma^\varepsilon_n) - \mathsf{Ratio}(\Pi_n)| &= \left|\frac{\mathsf{Cost}(\Pi_n) + \sum_{i=1}^n \sigma_i}{\mathsf{Reward}(\Pi_n) + \sum_{i=1}^n \tau_i} - \frac{\mathsf{Cost}(\Pi_n)}{\mathsf{Reward}(\Pi_n)}\right| \\
&= \left|\frac{\mathsf{Reward}(\Pi_n)\sum_{i=1}^n \sigma_i - \mathsf{Cost}(\Pi_n)\sum_{i=1}^n \tau_i}{(\mathsf{Reward}(\Pi_n))(\mathsf{Reward}(\Pi_n) + \sum_{i=1}^n \tau_i)}\right| \\
&\leq \frac{2(2C\varepsilon n)(nC)}{(\lambda n - \mu)(\lambda n - \mu - 2C\varepsilon n)} \\
&\leq \frac{4C^2\varepsilon}{\frac{\lambda}{2}(\frac{\lambda}{2} - 2C\varepsilon)} \text{ for some sufficiently great } n
\end{aligned}
$$

Thus, we get that

$$
|\underline{\mathsf{Ratio}}(\Gamma^\varepsilon) - \mathsf{Ratio}(\Pi)| \leq \frac{4C^2\varepsilon}{\frac{\lambda}{2}(\frac{\lambda}{2} - 2C\varepsilon)}
$$

We are done and we have even constructed a family of optimal schedules $(\Gamma^\varepsilon)_{\varepsilon>0}$. ◻

Note also that in $\mathcal{A}_{\mathsf{cp}}$ (which is a finite automaton), optimal schedules are cycles for which the ratio is defined [21, 17, 18]. The previous proposition thus proves the completeness of the corner-point abstraction and concludes this section.

8 Complexity Issues

In this section, we study the complexity of the optimal ratio problem and we prove that it is PSPACE-complete

Proposition 7. *The optimal ratio problem for (strongly reward-diverging) DPTAs (with non-negative rewards) is* PSPACE*-hard.*

Proof. The proof is done using a reduction on the reachability problem of timed automata, which is known to be PSPACE-complete [5]. Let $\mathcal{A} = (L, \ell_0, E, I, F)$ be a timed automata with F a set of final states. We transform $\mathcal{A}$ into a DPTA $\mathcal{B} = (L, \ell_0, E \cup E', I, \mathsf{c}, \mathsf{r})$ where:

- $E' = \{\ell \longrightarrow \ell \mid \ell \in F\}$
- $\mathsf{c}(\ell \xrightarrow{g,Y} \ell') = 2$ and $\mathsf{r}(\ell \xrightarrow{g,Y} \ell') = 1$ for every transition $\ell \xrightarrow{g,Y} \ell'$ in E whereas we set $\mathsf{c}(\ell \longrightarrow \ell) = \mathsf{r}(\ell \longrightarrow \ell) = 1$ for every $\ell \longrightarrow \ell$ in E'.

We have that a final state of $\mathcal{A}$ is reachable if and only if the optimal ratio in $\mathcal{B}$ is 1. □

For the upper bound complexity, we know that the optimal ratio problem for finite systems has a complexity $\mathcal{O}(n^2 m)$ where n is the number of vertices of the graph and m is the number of edges of the graph (see [18]). Now, the size of the corner-point abstraction is exponential in the number of clocks of the system, thus we get that the optimal ratio problem is in EXPTIME. However, we can do better and we get:

Proposition 8. *The optimal ratio problem for strongly reward-diverging DPTAs with non-negative rewards is in* PSPACE*.*

Proof. We fix a DPTA $\mathcal{A}$ and we consider its corner-point abstraction. We have proved that optimal schedules in $\mathcal{A}$ correspond to optimal cycles in the corner-point abstraction. Each state of $\mathcal{A}_{\mathsf{cp}}$ can be stored in polynomial space (see [5]). We first guess a cycle in $\mathcal{A}_{\mathsf{cp}}$, its length is at most exponential. Thus, computing its cost and reward can be done in polynomial space (the cost and reward are linear in the number of states). This gives an upper bound on the optimal ratio. The question is now: is there an other cycle which has a better ratio? We generate nondeterministically an other cycle and compute its ratio as previously. If this cycle is better, then the previous cycle was not the optimal one. The complexity of this last step is in CO-NPSPACE, which is known to be equal to NPSPACE, *i.e.* PSPACE (see [27], p.153). The global complexity of this procedure is thus in NPSPACE, *i.e.* in PSPACE. □

9 Future Work and Conclusion

In this paper, we have shown that the optimal infinite scheduling problem is computable for double-priced timed automata (and PSPACE-complete). We have reduced the problem to the computation of optimal infinite schedules in (weighted) finite-state graphs. This problem is equivalent to finding optimal cycles in finite-state graphs, which can be done using algorithms like Karp's algorithm [21] and some of its extensions and improvements [17, 18].

However, there is still a number of issues which are open for future work. The proof of computability, based on regions and corner-points, does not provide a realistic implementation strategy. We would like to obtain an efficient implementation based on zones and on-the-fly exploration of the symbolic state-space. A restriction to a setting where one of the prices (cost or reward) is uniform (same rate in all locations) may be particularly useful. Implementations for this specific case could be much more efficient than those for the general problem. An idea would then be to approximate optimal infinite schedules by working with (repeated) cost horizons or by applying partitioning and refinement techniques, as done in the tool Rapture [15, 16].

An extension of our present work would be to address the problem in the presence of adversaries, even if it seems very difficult, more difficult than that of cost-optimal winning strategies for (single-)priced timed automata with adversaries [22, 3, 13]. In the finite-state setting, however, the problem has been solved [28].

References

1. Y. Abdeddaim and O. Maler. Job-shop scheduling using timed automata. In *Proc. 13th International Conference on Computer Aided Verification (CAV'01)*, volume 2102 of *Lecture Notes in Computer Science*, pages 478–492. Springer, 2001.
2. Y. Abdeddaïm and O. Maler. Preemptive job-shop scheduling using stopwatch automata. In *Proc. 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'02)*, volume 2280 of *Lecture Notes in Computer Science*, pages 113–126. Springer, 2002.
3. R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability in weighted timed games. In *Proc. 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, volume 3142 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2004.

4. R. Alur and D. Dill. Automata for modeling real-time systems. In *Proc. 17th International Colloquium on Automata, Languages and Programming (ICALP'90)*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer, 1990.
5. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
6. R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In *Proc. 4th International Workshop on Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *Lecture Notes in Computer Science*, pages 49–62. Springer, 2001.
7. D. Applegate and W. Cook. A computational study of the job-shop scheduling problem. *OSRA Journal on Computing*, 3:149–156, 1991.
8. E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *Proc. 2nd International Workshop on Hybrid Systems: Computation and Control (HSCC'99)*, volume 1569 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 1999.
9. G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Efficient guiding towards cost-optimality in uppaal. In *Proc. 7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'01)*, volume 2031 of *Lecture Notes in Computer Science*, pages 174–188. Springer, 2001.
10. G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. In *Proc. 4th International Workshop on Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *Lecture Notes in Computer Science*, pages 147–161. Springer, 2001.
11. J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, W. Yi, and C. Weise. New generation of uppaal. In *Proc. International Workshop on Software Tools for Technology Transfer (STTT'98)*, BRICS Notes Series, pages 43–52, 1998.
12. P. Bouyer, E. Brinksma, and K. G. Larsen. Staying alive as cheaply as possible. In *Proc. 7th International Workshop on Hybrid Systems: Computation and Control (HSCC'04)*, volume 2993 of *Lecture Notes in Computer Science*, pages 203–218. Springer, 2004.
13. P. Bouyer, F. Cassez, E. Fleury, and K. G. Larsen. Optimal strategies in priced timed game automata. In *Proc. 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'04)*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160. Springer, 2004.
14. E. Brinksma, A. Mader, and A. Fehnker. Verification and optimization of a PLC control schedule. *Journal of Software Tools for Technology Transfer*, 4(1):21–33, 2002.

15. P. R. D'Argenio, B. Jeannet, H. E. Jensen, and K. G. Larsen. Reachability analysis of probabilistic systems by successive refinements. In *Proc. 1st Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification (PAPM-PROBMIV'01)*, volume 2165 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2001.

16. P. R. D'Argenio, B. Jeannet, H. E. Jensen, and K. G. Larsen. Reduction and refinement strategies for probabilistic analysis. In *Proc. 2nd Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification (PAPM-PROBMIV'02)*, volume 2399 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2002.

17. A. Dasdan and R. K. Gupta. Faster maximum and minimum mean cycle algorithms for system performance analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(10):889–899, 1998.

18. A. Dasdan, S. Irani, and R. K. Gupta. Efficient algorithms for optimum cycle mean and optimum cost to time ratio problems. In *Proc. 36th ACM/IEEE Design Automation Conference (DAC'99)*, pages 47–42. ACM, 1999.

19. A. Fehnker. Scheduling a steel plant with timed automata. In *Proc. 6th International Conference on Real-Time Computing Systems and Applications (RTCSA'99)*, pages 280–286. IEEE Computer Society Press, 1999.

20. T. Hune, K. G. Larsen, and P. Pettersson. Guided synthesis of control programs using uppaal. In *Proc. IEEE ICDS International Workshop on Distributes Systems Verification and Validation*, pages E15–E22. IEEE Computer Society Press, 2000.

21. R. M. Karp. A characterization of the minimum mean-cycle in a digraph. *Discrete Mathematics*, 23(3):309–311, 1978.

22. S. La Torre, S. Mukhopadhyay, and A. Murano. Optimal-reachability and control for acyclic weighted timed automata. In *Proc. 2nd IFIP International Conference on Theoretical Computer Science (TCS 2002)*, volume 223 of *IFIP Conference Proceedings*, pages 485–497. Kluwer, 2002.

23. K. G. Larsen, G. Behrmann, E. Brinksma, A. Fehnker, T. Hune, P. Pettersson, and J. Romijn. As cheap as possible: Efficient cost-optimal reachability for priced timed automata. In *Proc. 13th International Conference on Computer Aided Verification (CAV'01)*, volume 2102 of *Lecture Notes in Computer Science*, pages 493–505. Springer, 2001.

24. K. G. Larsen, P. Pettersson, and W. Yi. Uppaal in a nutshell. *Journal of Software Tools for Technology Transfer*, 1(1–2):134–152, 1997.

25. A. Mader. Deriving schedules for a smart card personalisation system. Technical report TR-CTIT-04-05, Centre for Telematics and Information Technology, University of Twente, The Netherlands, 2004.

26. P. Niebert and S. Yovine. Computing efficient operations schemes for chemical plants in multi-batch mode. *European Journal of Control*, 7(4):440–453, 2001.
27. C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
28. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1–2):343–359, 1996.