Optimal Conditional Reachability for Multi-priced Timed Automata

Kim Guldstrand Larsen and Jacob Illum Rasmussen

Department of Computer Science, Aalborg University, Denmark
{kgl, illum}@cs.aau.dk

Abstract. In this paper, we prove decidability of the optimal conditional reachability problem for multi-priced timed automata, an extension of timed automata with multiple cost variables evolving according to given rates for each location. More precisely, we consider the problem of determining the minimal cost of reaching a given target state, with respect to some primary cost variable, while respecting upper bound constraints on the remaining (secondary) cost variables. Decidability is proven by constructing a zone-based algorithm that always terminates while synthesizing the optimal cost with a single secondary cost variable. The approach is then lifted to any number of secondary cost variables.

1 Introduction

Recently, research has been focused on extending the framework of timed automata (TA), [2], towards linear hybrid automata (LHA), [3], by allowing continuous variables with non-uniform rates and maintaining a decidable reachability problem.

One such class of models is that of priced (or weighted) timed automata (PTA), [9, 4], which are timed automata augmented with a single cost variable. For this class of timed automata, the minimum-cost reachability problem, i.e. finding the minimum cost of reaching some goal location, is decidable. The restriction with respect to linear hybrid automata is that the cost variable cannot be tested in guards and invariant, it cannot be reset[1], and it grows monotonically.

Ignoring the variable c_2, Figure 1 depicts a PTA for which the rate of c_1 is, respectively, 1 and 2 in locations l_1 and l_2. The type of reachability question we can ask for this model is: What is cheapest way of reaching the "happy" location. The answer, in this case, is 3 which is achieved by delaying for 1 time unit in l_1, taking the transition to l_2 and delaying for 1 time unit before proceeding to l_3.

A natural extension of PTA is to allow a secondary cost variable, thus arriving at dual-priced timed automata (DPTA), and pose reachability questions of the type: What is the cheapest primary cost of reaching the "happy" location

[1] Variables with these two properties are sometimes referred to as observers in the literature.

V. Sassone (Ed.): FOSSACS 2005, LNCS 3441, pp. 234–249, 2005.

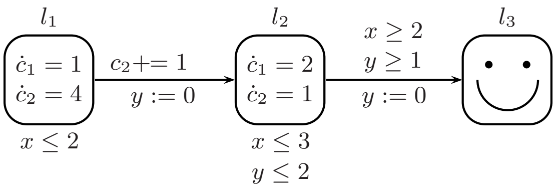

Fig. 1. Example dual-priced timed automata

under some upper bound constraint on the secondary cost? We term this *optimal conditional reachability*. There are three cases to consider, if the secondary cost is time, if the primary cost is time, and if neither the primary nor the secondary cost is time. In the first case, we can augment a PTA with time invariants corresponding to the upper bound constraint on all locations and then use minimum-cost reachability for PTA. In the second case, we can combine finding fastest traces in TA with minimum-cost reachability for PTA. The third case is the topic of this paper. Figure 1 provides a model with two cost variables for which we can pose questions of the type: What is the minimum cost for c_1 of reaching the "happy" location, while respecting $c_2 \leq 4$. The answer to this question is $\frac{11}{3}$ and is obtained by delaying for $\frac{1}{3}$ time units in l_2, then proceeding to l_2 and waiting $\frac{5}{3}$ time units before proceeding to l_3. This example illustrates that unlike minimum-cost reachability for PTA, optimal conditional reachability with two cost variables may have non-integral solutions[2].

If we generalize DPTA to allow any finite number of cost variables, we arrive at *multi-priced timed automata* (MPTA). Optimal conditional reachability for MPTA concerns minimizing the first cost variable while respecting upper bound constraints on the rest. The main contribution of this paper is the decidability of optimal conditional reachability for MPTA.

Relevant work on MPTA include the model checking problem of MPTA with respect to weighted CTL which has been studied by Brihaye et al., [6], and proven undecidable, even with discrete time.

The discrete version of conditional reachability is called multi-constrained routing and is well-known to be NP-complete, [7]. Recently, the problem has been reconsidered by Puri and Tripakis in [11] where several algorithms are proposed for solving the problem, both exactly and approximately.

For simplicity of the proofs, we prove decidability of optimal conditional reachability for MPTA, by proving the decidability for the simpler DPTA model. To show that the result can be lifted to from DPTA to MPTA, we provide, throughout the paper, descriptions of how important aspects are extended from pairs of costs to k-tuples of costs.

The rest of this paper is organized as follows. In Section 2, we give an abstract framework for symbolic optimal conditional reachability in terms of dual-priced

[2] The simple model in Figure 1 is acyclic, so optimal conditional reachability can be reduced to linear programming.

transition systems, including a generic algorithm for conditional optimal reachability. In Section 3, we introduce dual-priced timed automata as a syntactic model for dual-priced transition systems. In section 4, we introduce dual-priced zones as the main construct for dual-priced symbolic states. In Section 5, we define a successor operator on the constructs of the previous section. In Section 6, we discuss termination of our algorithm. Finally, we conclude the paper in Section 7 and point out directions for future research.

2 Conditional Optimal Reachability

The notation defined in this section aims at being consistent with that of [11].

The partial order, $\preceq$, over $\mathbb{R}^2_+$ defined such that $(a,b) \preceq (c,d)$ iff $a \leq c$ and $b \leq d$ is called a *domination order*. Given a set of points, $A \subseteq \mathbb{R}^2_+$, an element, $(c,d) \in A$, is said to be *redundant* if there exists another element $(a,b) \in A$ such that $(a,b) \preceq (c,d)$. We extend domination to sets, $A, B \in 2^{\mathbb{R}^2_+}$ such that $A \preceq B$ iff every $(a,b) \in B$ is redundant in A. Figure 2 depicts a set of points with black and white bullets denoting, respectively, redundant and non-redundant nodes.

A dual-priced transition system is a structure $\mathcal{T} = (S, s_0, \Sigma, \rightarrow)$ where S is a, possibly infinite, set of states, s_0 is the initial state, Σ is an alphabet of labels, and $\rightarrow$ is partial function with signature $S \times \Sigma \times S \hookrightarrow \mathbb{R}^2_+$. For brevity, we will use the notation $s \rightarrow s'$ to denote that $\exists c_1, c_2, a : \rightarrow (s, a, s') = (c_1, c_2)$, we use $s \xrightarrow[c_1,c_2]{a}$ whenever $\rightarrow (s,a,s') = (c_1,c_2)$, the notation $s \xrightarrow{c_1,c_2} s'$ for $\exists a \in \Sigma : s \xrightarrow[c_1,c_2]{a} s'$, and the notation $s \xrightarrow{a} s'$ for $\exists c_1, c_2 : s \xrightarrow[c_1,c_2]{a} s'$. The components of a cost pair are denoted primary, respectively, secondary costs.

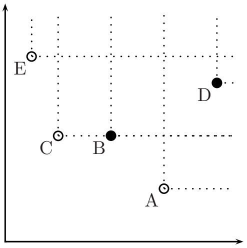

Fig. 2. Domination in $\mathbb{R}^2_+$

An execution ε of $\mathcal{T}$ is a sequence $\varepsilon = s_0 \xrightarrow[c_1^1,c_2^1]{a_1} s_1 \xrightarrow[c_1^2,c_2^2]{a_2} \cdots \xrightarrow[c_1^n,c_2^n]{a_n} s_n$. The cost, $\mathsf{Cost}(\varepsilon)$, of an execution ε is given as $\mathsf{Cost}(\varepsilon) = (\mathsf{Cost}_1(\varepsilon), \mathsf{Cost}_2(\varepsilon))$, where $\mathsf{Cost}_j(\varepsilon) = \sum_{i=1}^n c_j^i$ for $j \in \{1,2\}$.

The minimal primary cost of reaching a set of goal states, $G \subseteq S$, under an upper bound, p, on the secondary cost is termed the conditional optimal cost and given as:

$$\mathsf{mincost}_{\leq p}(G) = \inf\{\mathsf{Cost}_1(\varepsilon) \mid \varepsilon = s_0 \xrightarrow{c_1^1,c_2^1} \cdots \xrightarrow{c_1^n,c_2^n} s \in G,\ \mathsf{Cost}_2(\varepsilon) \leq p\} \quad (1)$$

In order to effectively analyze dual-priced transition systems we suggest dual-priced symbolic states of the form (A,π) where $A \subseteq S$ and $\pi : A \rightarrow 2^{\mathbb{R}^2_+}$. Intuitively, reachability of the dual-priced symbolic state (A,π) has the interpretation that all s of A are reachable with costs arbitrarily close to all $\pi(s)$.

To express successors of dual-priced symbolic states we use the Post operator $\mathsf{Post}_a(A, \pi) = (B, \eta)$ where:

$$B = \{s' \mid \exists s \in A : s \xrightarrow{a} s'\}, \text{ and} \tag{2}$$

$$\eta(s) = \{(c_1 + c, c_2 + c') \mid \exists s' \in A : s' \xrightarrow[c_1,c_2]{a} s \text{ and } \pi(s') = (c, c')\} \tag{3}$$

A symbolic execution ξ of a dual-priced transition system is a sequence $\xi = (A_0, \pi_0), \ldots, (A_n, \pi_n)$, where $A_0 = \{s_0\}$, $\pi(s_0) = (0, 0)$ and for $1 \leq i \leq n$ we have $(A_i, \pi_i) = \mathsf{Post}_a(A_{i-1}, \pi_{i-1})$, for some $a \in \Sigma$. The correspondence between executions and symbolic executions is captured below:

- For each execution ε of $\mathcal{T}$ ending in s there is a symbolic execution ξ ending in (A, π) such that $s \in A$ and $\mathsf{Cost}(\varepsilon) \in \pi(s)$.
- Let ξ be a symbolic execution of $\mathcal{T}$ ending in (A, π), then for each s and $(c, c') \in \pi(s)$ there is an execution ε ending in s such that $\mathsf{Cost}(\varepsilon) = (c, c')$.

From the above it follows that symbolic states accurately capture conditional optimal reachability in the sense that:

$$\mathsf{mincost}_{\leq p}(G) = \inf\{minCost_{\leq p}(A \cap G, \pi) \mid (A, \pi) \text{ is reachable}\}, \tag{4}$$

where $minCost_{\leq p}(A, \pi)$ is defined as $\inf\{c | \exists s \in A : (c, c') \in \pi(s) \text{ and } c' \leq p\}$. Furthermore, we define the relation $\sqsubseteq$ on dual-priced symbolic states such that $(B, \eta) \sqsubseteq (A, \pi)$ iff $A \subseteq B$ and $\eta(s) \preceq \pi(s)$ for all $s \in A$. In other words, B is bigger than A and for each state, s, in A, $\pi(s)$ is dominated by $\eta(s)$.

Based on the above result, we provide an algorithm for computing the optimal conditional reachability problem, $\mathsf{mincost}_{\leq p}(G)$, in Figure 3.

```
Cost $:= \infty$
Passed $:= \emptyset$
Waiting $:= \{(\{s_0\}, \pi_0)\}$
while Waiting $\neq \emptyset$ do
    select $(A, \pi) \in$ Waiting
    if $A \cap G \neq \emptyset$ then
        if $minCost_{\leq p}(A \cap G, \pi) \leq$ Cost then
            Cost $:= minCost_{\leq p}(A \cap G, \pi)$ fi
    fi
    if $\forall (B, \eta) \in$ Passed $: (B, \eta) \not\sqsubseteq (A, \pi)$then
        Passed $:=$ Passed $\cup \{(A, \pi)\}$
        Waiting $:=$ Waiting $\cup \bigcup_{a \in \Sigma} \mathsf{Post}_a(A, \pi)$ fi
    fi
od
return Cost
```

Fig. 3. General algorithm for computing the optimal conditional reachability cost, $\mathsf{mincost}_{\leq p}(G)$

The algorithm maintains two lists, a PASSED and a WAITING list, that hold the states already explored and the states waiting to be explored, respectively. Initially, the PASSED list is empty and the WAITING list contains only the initial state. The algorithm iterates as long as the WAITING list in non-empty.

At each iteration the algorithm select a state, (A, π), from the WAITING list. The set of states, A, is checked for intersection with the set of goal states. If the intersection is non-empty, the minimum primary cost of any goal state satisfying the constraint on the secondary cost is computed and compared to COST, and if the computed cost is the smaller of the two, COST is updated appropriately.

Whether A intersects with the goal states or not, we go through the PASSED list and check whether it contains any (B, η) such that $(B, \eta) \sqsubseteq (A, \pi)$. If it does, (A, π) is discarded as it is dominated by (B, η), otherwise we add all successors of (A, π) to the WAITING list and add itself to the PASSED list.

The algorithm terminates when the WAITING list is empty and at this point, COST holds $\mathsf{mincost}_{\leq p}(G)$. Termination of the algorithm is guaranteed if $\sqsubseteq$ is a well-quasi ordering on dual-priced symbolic states.

For optimization of the algorithm, further pruning of elements in the WAITING list can be performed simultaneously with the inclusion check, $\sqsubseteq$, e.g. keeping only elements where the set of states with primary cost smaller than COST and secondary cost smaller than p. This is correct since both primary and secondary costs increase monotonically in any trace. Furthermore, for any encountered pair (A, π) with $s \in A$ we could prune $\pi(s)$ for redundant elements.

Every aspect in this section about dual-priced transition systems, including the generic algorithm, can be directly extended to multi-priced transition systems with k-tuples of cost and optimal conditional reachability of the form $\mathsf{mincost}_{\leq p_2,\ldots p_k}(G)$. That is, minimize the primary cost under individual upper bound constraints on the $k-1$ secondary costs.

The above framework may be instantiated by providing concrete syntax for dual-priced transition systems and data structures for dual-priced symbolic states that, first, allow effective computation of the Post operator and, second, have a well-quasi ordered relation, $\sqsubseteq$. In the following sections, we provide such an instantiation of the above framework.

3 Dual-Priced Timed Automata

In this section we define dual-priced timed automata which is a proper subset of linear hybrid automata, [3], and a proper superset of priced timed automata, [9], or weighted timed automata, [4], and in turn timed automata, [2]. DPTA will serve as a concrete syntax for dual-priced transition systems. First however, we recall some basic notation from the theory of timed automata.

We work with a finite set, $\mathbb{C}$, of positive, real-valued variables called clocks. $\mathcal{B}(\mathbb{C})$ is the set of formulae obtained as conjunctions of atomic constraints of the form $x \bowtie n$, where $x \in \mathbb{C}$, $n \in \mathbb{N}$, and $\bowtie \in \{\leq, =, \geq\}$[3]. We refer to the elements of $\mathcal{B}(\mathbb{C})$ as clock constraints. $\mathcal{B}(\mathbb{C})^*$ is the set of clock constraints involving only upper bounds, i.e. $\leq$.

Clock values are represented as functions from $\mathbb{C}$ to the set of non-negative reals, $\mathbb{R}_+$, called clock valuations and ranged over by u, u' etc.

For a clock valuation, $u \in (\mathbb{C} \to \mathbb{R}_+)$, and a clock constraint, $g \in \mathcal{B}(\mathbb{C})$, we write $u \in g$ when u satisfies all the constraints of g. For $t \in \mathbb{R}_+$, we define the operation $u + t$ to be the clock valuation that assigns $u(x) + t$ to all clocks, and for $R \subseteq \mathbb{C}$ the operation $u[R \to 0]$ to be the clock valuation that agrees with u for all clocks in $\mathbb{C} \backslash R$ and assigns zero to all clocks in R. $u[x \to 0]$ is shorthand for $u[\{x\} \to 0]$. Furthermore, u_0 is defined to be the clock valuation that assigns zero to all clocks.

Definition 1 (Dual-Priced Timed Automata). *A dual-priced timed automaton is a 6-tuple $\mathcal{A} = (L, l_0, \mathbb{C}, E, I, \boldsymbol{P})$ where $\boldsymbol{P} = \{\mathcal{P}_1, \mathcal{P}_2\}$[4], L is a finite set of locations, l_0 is the initial location, $\mathbb{C}$ is a finite set of clocks, $E \subseteq L \times \mathcal{B}(\mathbb{C}) \times 2^{\mathbb{C}} \times (\mathbb{N} \times \mathbb{N}) \times L$ is the set of edges, $I : L \to \mathcal{B}(\mathbb{C})^*$ assigns invariants to locations, and $\mathcal{P}_i : L \to \mathbb{N}$ assigns prices to locations, $i \in \{1, 2\}$.*

The concrete state semantics of a DPTA, $\mathcal{A} = (L, l_0, \mathbb{C}, E, I, \boldsymbol{P})$, is given in terms of a dual-priced transition system with state set $L \times (\mathbb{C} \to \mathbb{R}_+)$, initial state (l_0, u_0), alphabet $\Sigma = E \cup \{\delta\}$, and the transition relation, $\to$, defined as:

- $(l, u) \xrightarrow[t \cdot \mathcal{P}_1(l), t \cdot \mathcal{P}_2(l)]{\delta} (l, u + t)$ if $\forall\, 0 \leq t' \leq t : u + t' \in I(l)$ and
- $(l, u) \xrightarrow[c, c']{e} (l', u')$ if $e = (l, g, R, (c, c'), l') \in E, u \in g, u' = u[R \to 0]$.

We will often write concrete states as (l, u, c_1, c_2) to denote the assumption of some underlying execution, ε, ending in (l, u) with $\mathsf{Cost}(\varepsilon) = (c_1, c_2)$.

A concrete dual-priced state (l, u, c_1, c_2) is said to dominate another state (l', u', c_1', c_2') iff $l = l'$, $u = u'$, and $(c_1, c_2) \preceq (c_1', c_2')$. In such case we write $(l, v, c_1, c_2) \preceq (l', v', c_1', c_2')$.

For convenience reasons, we assume some restrictions on the structure of the DPTA in the rest of the paper. First, any DPTA should be bounded, i.e. all locations have upper bound invariants on all clocks. Second, at least one clock is reset on every transition. Note that neither restriction compromises the generality of our result, as it is well-known that any TA can be transformed into a semantically equivalent bounded TA, and that result extends directly to DPTA. Furthermore, the reset assumption can be guaranteed by introducing an extra clock which is reset on every transition.

[3] For simplification we do not include strict inequalities, note, however, that everything covered in this paper extends directly to strict inequalities, which is why we compute infimum costs as opposed to minimum costs.

[4] If we let $\boldsymbol{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ we have MPTA with analogous semantics.

3.1 Relation to Linear Hybrid Automata

Any DPTA is a LHA where the value of the rate of each clock variable is one is every location, and the rates of the primary and secondary costs are $\mathcal{P}_1(l)$ and $\mathcal{P}_2(l)$, respectively, in location l.

Tools such as HyTech, [8], can perform forward symbolic reachability analysis on LHA over a set of variables $\boldsymbol{x}$ using symbolic state structures (l, A, b) where l is a location and $A \cdot \boldsymbol{x} \leq b$ defines a convex polyhedra of valid variable assignments. One of the main properties of this kind of reachability analysis is that the Post operators defined for LHA maintains convexity of the state set. However, the reachability problem for LHA is, in general, undecidable, so termination of the reachability algorithm is not guaranteed. However, a consequence of our result is that for the class of DPTA, HyTech will terminate when performing conditional reachability.

4 Dual-Priced Zones

Now, we propose dual-priced zones as a syntactic construct for providing a symbolic semantics for the dual-priced transition system induced by DPTA.

The constructs of our proposal for dual-priced symbolic states are *zones* and *cost functions*. Zones are well-known from the analysis of timed systems and efficient implementations of zones as difference bound matrices are used in real-time verification tools such as Kronos, [5], and Uppaal, [10]. Briefly, zones are convex collections of clock valuations that can be described solely using difference constraints of the form $x_i - x_j \leq m$ where $m \in \mathbb{Z}$ and $x_i, x_j \in \mathbb{C} \cup \{x_0\}$, where, x_0, is a special clock whose value is fixed to zero. That way, constraints of the form $x_i \geq n$ can be written as $x_0 - x_i \leq -n$, and similarly for other constraints involving a single variable. Zones are ranged over by $Z, Z_1, Z', \ldots$. When a clock valuation, u, satisfies the difference constraints of a zone, Z, we write $u \in Z$.

The second construct is a cost function, which is an affine function over $\mathbb{C}$, i.e. a cost function, d, is a function with signature $(\mathbb{C} \rightarrow \mathbb{R}_+) \rightarrow \mathbb{R}_+$ that can be written syntactically as $a_1 \cdot x_1 + \cdots + a_n \cdot x_n + b$ where $x_i \in \mathbb{C}$, $1 \leq i \leq n$, and $a_i, b \in \mathbb{Z}$. The cost of a clock valuation, u, in a cost function, d, is given by $d(u) = a_1 \cdot u(x_1) + \cdots + a_n \cdot u(x_n) + b$. We range over cost function by d, e, d_1, e_1, d', e' etc. For ease of notation we define a number of operations on cost functions. Let $m \in \mathbb{Z}$, $p \in \mathbb{N}$ and $x_i, x_j \in \mathbb{C}$, then the substitution operation $d[x_i/\varphi]$ for $\varphi \in \{m, x_j + m\}$ is defined as $d[x_i/\varphi] = a_1 \cdot x_1 + \cdots + a_i \cdot \varphi + \cdots + a_n \cdot x_n$. The delay operation $d^{\uparrow p, x_i}$ is defined as $d^{\uparrow p, x_i} = a_1 \cdot x_1 + \cdots + (p - \sum_{j \neq i} a_j) \cdot x_i + \cdots + a_n \cdot x_n$, meaning we want the sum of the coefficients to match p by assign the correct coefficient to x_i.

Let C be a set of pairs of cost functions, i.e. $C = \{(e_1, d_1), \ldots, (e_k, d_k)\}$ and u a clock valuation, then $C(u) = \{(e_1(u), d_1(u)), \ldots (e_k(u), d_k(u))\}$ is a set of points in $\mathbb{R}^2_+$. We denote by $\lambda(C(u))$ the set of all convex combinations of $C(u)$, i.e. the convex hull.

For the construction of dual-priced symbolic states we propose dual-priced zones as given in Definition 2 below.

Definition 2 (Dual-Priced Zone). *A dual-priced zone is a pair, (Z, C), where Z is a zone and C is a set of pairs of cost functions $\{(e_1, d_1), \ldots, (e_k, d_k)\}$.*

We construct dual-priced symbolic states as structures (l, Z, C) where l is a location and (Z, C) is a dual-priced zone. A dual-priced symbolic state (l, Z, C) contains all concrete states (l', u, c_1, c_2) where $l' = l$, $u \in Z$, and $(c_1, c_2) \in \lambda(C(u))$. Not that dual-priced zones extend directly to multi-priced zones with k-tuples of cost functions and, in turn, multi-priced symbolic states.

In [9], efficient data structures for symbolic minimum-cost reachability for priced timed automata (PTA) are provided. These are so-called priced zones which effectively are zones, Z, with an associated cost function, e. For representing cost in the discrete case described in [11], subsets of $\mathbb{N} \times \mathbb{N}$ are used for representing reachability costs.

The immediate combination of the two suggest the use of zones together with sets of pairs of cost function. The following example illustrates why we also need to consider convex combinations of the cost functions.

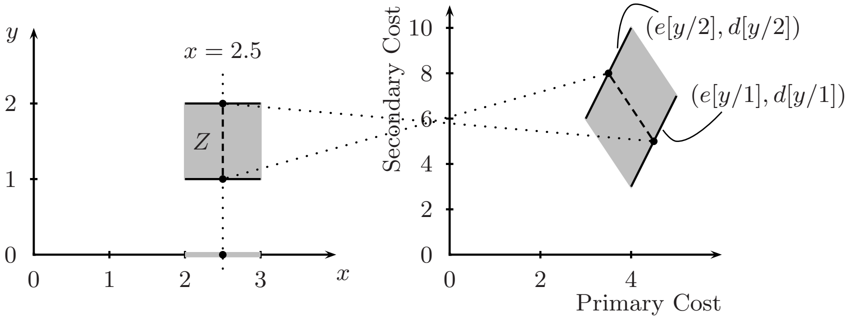

Fig. 4. The relationship between the zone, Z, defined by the constraints $2 \leq x \leq 3$ and $1 \leq y \leq 2$ with cost functions (e, d) with $e = x + y$ and $d = 4x - 3y + 1$

Consider the zone of Figure 4 described by the constraints $2 \leq x \leq 3$ and $1 \leq y \leq 2$ with the pair of cost functions (e, d) where $e = x+y$ and $d = 4x-3y+1$. Now, if we need to compute the projection of the zone onto the first axis due to a reset of y, what should the set of pairs of cost functions be to represent or dominate the possible cost values? The suggestion following the lines of reasoning from [9] would be to use the two pairs of cost functions $(e[y/2], d[y/2])$ and $(e[y/1], d[y/1])$. This choice, however, has a loss of information if we do not allow convex combinations. The point $(x = 2.5, y = 0)$ is obtained from Z by projection from any point satisfying $(x = 2.5, 1 \leq y \leq 2)$ corresponding to costs given by any convex combination between $(3.5, 8)$ and $(4.5, 5)$. However, maintaining only the these two points is incorrect, as neither of the points dominate any point in their convex combination.

5 Post Operator

The projection operation in the previous section serves as a first step towards a Post operator. Consider, again, the zone in Figure 4 and assume it is, now, associated with two pairs of cost functions (e_1, d_1) and (e_2, d_2), between which we allow arbitrary convex combinations. Now, if we perform a projection onto the first axis we split each pair of cost functions in two, i.e. (e_i^L, d_i^L) and (e_i^U, d_i^U), $i \in \{1, 2\}$, corresponding to the lines $L : y = 1$ and $U : y = 2$, respectively, giving four cost functions. Originally, for any clock valuation, u, in the zone and $0 \leq \alpha \leq 1$, the convex combination between $(e_1(u), d_1(u))$ and $(e_2(u), d_2(u))$ wrt. α is a valid cost pair. However, when we split the cost functions, the cost corresponding to e.g. $(e_1(u), d_1(u))$ is given by some convex combination of (e_1^L, d_1^L) and (e_1^U, d_1^U) for the clock valuation $u[y \rightarrow 0]$, and similarly for $(e_2(u), d_2(u))$ using the same convex combination. Contrary to the definition of dual-priced zones, this suggests not to allow arbitrary convex combinations between (e_1^L, d_1^L) and (e_1^U, d_1^U), (e_2^L, d_2^L) and (e_2^U, d_2^U), but rather "binary tree" convex combinations of the form: Choose the same convex combination between (e_1^L, d_1^L), (e_1^U, d_1^U) and (e_2^L, d_2^L), (e_2^U, d_2^U) and take any convex combination of the resulting pairs. However, the following key lemma states that if this set is convex, it is identical to the set of arbitrary convex combinations between the four.

Lemma 1. *Assume a set of pairs of points in* $\mathbb{R}^2_+$

$$\{(a_1, b_1), \ldots, (a_n, b_n)\}, \;\; a_i \in \mathbb{R}^2_+, b_i \in \mathbb{R}^2_+, 1 \leq i \leq n$$

For $0 \leq \alpha \leq 1$*, let:*

$$\begin{aligned} A_\alpha &= \{\alpha \cdot a_i + (1 - \alpha) \cdot b_i | 1 \leq i \leq n\} \text{ and} \\ B &= \{a_i, b_i | 1 \leq i \leq n\}. \end{aligned}$$

Now, if $\bigcup_\alpha \lambda(A_\alpha)$ *is convex (i.e.* $\bigcup_\alpha \lambda(A_\alpha) = \lambda(\bigcup_\alpha \lambda(A_\alpha))$*) then* $\bigcup_\alpha \lambda(A_\alpha) = \lambda(B)$*.*

Proof. We prove the lemma in two steps. First, we show that $\bigcup_\alpha \lambda(A_\alpha) \subseteq \lambda(B)$ and, secondly, that $\lambda(B) \subseteq \bigcup_\alpha \lambda(A_\alpha)$.
1. Let c be a convex combination of A_α for any $0 \leq \alpha \leq 1$, that is,

$$\begin{aligned} c &= \lambda_1(\alpha a_1 + (1 - \alpha) b_1) + \cdots + \lambda_n(\alpha a_n + (1 - \alpha) b_n) && (5) \\ &= \lambda_1 \alpha a_1 + \lambda_1(1 - \alpha) b_1 + \cdots + \lambda_n \alpha a_n + \lambda_n(1 - \alpha) b_n, && (6) \end{aligned}$$

where $0 \leq \lambda_i \leq 1$ and $\sum_i \lambda_i = 1$. Now, (6) is a convex combination of B, thus $c \in \lambda(B)$ and in turn $\bigcup_\alpha \lambda(A_\alpha) \subseteq \lambda(B)$.
2. Each point a_i can be given as a convex combination of A_α where $\alpha = 1$ using $\lambda_i = 1$ and $\lambda_j = 0$ for $j \neq i$. Simililarly for b_i with $\alpha = 0$. Now, since all a_i, b_i are included in the convex set $\bigcup_\alpha \lambda(A_\alpha)$, we know that $\lambda(B) \subseteq \lambda(\bigcup_\alpha \lambda(A_\alpha)) = \bigcup_\alpha \lambda(A_\alpha)$. □

Note, that the proof makes no mention of $\mathbb{R}_+^2$, thus the Lemma 1 is directly extendible to pairs of points in $\mathbb{R}_+^k$.

At first glance, $\bigcup_\alpha \lambda(A_\alpha)$ in Lemma 1 might seem universally convex, however, Figure 5 depicts the contrary where Lemma 1 does not hold. Let $P = \{(A,B),(C,D)\}$, now, $\bigcup_\alpha \lambda(P_\alpha)$ (the gray area with the dashed line) is not convex and not equal to $\lambda(\{A,B,C,D\})$, particularly, all points on the line from A to D are not included in the former.

Before defining the Post operator on dual-priced states of the form (l,Z,C), we need to introduce a number of definitions and operations. Let Z be a zone, then the delay operation $Z^\uparrow$ and the reset, $\{x\}Z$, with respect to a clock, $x \in \mathbb{C}$, are defined as $Z^\uparrow = \{u+t | u \in Z \text{ and } t \geq 0\}$ and $\{x\}Z = \{u[x \to 0] | u \in Z\}$. It is well-known from timed automata that both $Z^\uparrow$ and $\{x\}Z$ are representable as zones.

Given a zone, Z, if $x_i - x_j \leq m$ is a constraint in Z then $(Z \wedge (x_i - x_j = m))$ is a *facet* of Z, a *lower relative facet* of x_j, and an *upper relative facet* of x_i. The set of lower (resp. upper) relative facets of a clock, x_i, in a zone, Z, is denoted $LF_{x_i}(Z)$ (resp. $UF_{x_i}(Z)$).

The following lemma for facets is proven in [9].

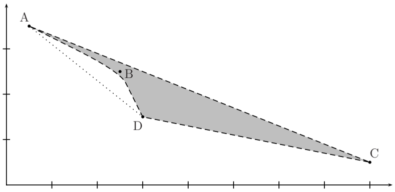

Fig. 5. Counter example

Lemma 2. *Let Z be a zone over a clock set, $\mathbb{C}$, with $x \in \mathbb{C}$, then:*

1. $Z^\uparrow = \bigcup_{F \in UF_{x_0}(Z)} F^\uparrow = Z \cup \bigcup_{F \in LF_{x_0}(Z)} F^\uparrow$ *and*
2. $\{x\}Z = \bigcup_{F \in LF_x(Z)} \{x\}F = \bigcup_{F \in UF_x(Z)} \{x\}F$.

Lemma 2.1 is most intuitively understood knowing that x_0 is fixed to zero, that way UF_{x_0} is the set of all lower bound constraints on clocks in $\mathbb{C}$ (i.e. $x \geq n$) and LF_{x_0} is the set of all upper bound constraints on clocks in $\mathbb{C}$ (i.e. $x \leq n$).

Definition 3. *Given a zone, Z, and a clock, x, $LUF_x(Z)$ is the unique, smallest collection of pairs $\{(L_1,U_1),\ldots,(L_n,U_n)\}$, such that for all $1 \leq i,j \leq n, i \neq j$ we have (i) $L_i \cap L_j = U_i \cap U_j = \emptyset$, (ii) $\{x\}L_i = \{x\}U_i$, and (iii) $L_i \subseteq F$, $U_i \subseteq F'$ for some $F \in LF_x(Z)$ and $F' \in UF_x(Z)$.*

We call the elements of $LUF_x(Z)$ partial relative facets with regard to x. Figure 6 illustrates the concept of partial relative facets.

Let d be a cost function and let F be a relative facet of a zone in the sense that $x_i - x_j = m$ (or $x_i = m$) is a constraint in F, then we use the shorthand notation d^F for $d[x_i/x_j + m]$ (or $d[x_i/m]$).

Definition 4 (Post Operator). *Let $\mathcal{A} = (L, l_0, \mathbb{C}, E, I, \boldsymbol{P})$ be a DPTA with $l \in L$ and $e = (l, g, \{x\}, (c, c'), l') \in E$[5], let Z be zone, let Z' be a zone where*

[5] For the general case with multiple resets, we consecutively split the pairs of cost functions for each clock that is reset.

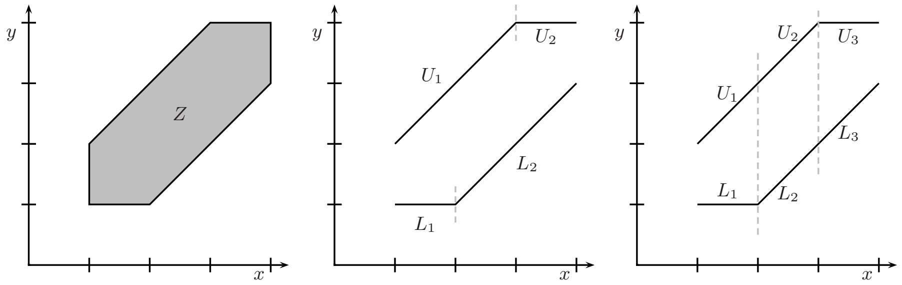

Fig. 6. From left to right (i): a zone, Z, (ii): $LF_y(Z) = \{L_1, L_2\}$ and $UF_y(Z) = \{U_1, U_2\}$ (iii): $LUF_y = \{(L_1, U_1), (L_2, U_2), (L_3, U_3)\}$

$x \in \mathbb{C}$ is fixed at zero, and let $C = \{(e_1, d_1), \dots, (e_k, d_k)\}$ be a set of pairs of cost functions, then

$$\mathsf{Post}_\delta(l, Z', C) = \Big\{(l, (Z' \wedge I(l))^\uparrow \wedge I(l), \{(e_i^{\uparrow \mathcal{P}_1(l),x}, d_i^{\uparrow \mathcal{P}_2(l),x}) | 1 \le i \le k\})\Big\}$$

$$\mathsf{Post}_e(l, Z, C) = \bigcup_{(L,U) \in LUF_x(Z \wedge g)} \Big\{(l', \{x\}(U), C')\Big\}$$

where $C' = \{(e_i^L + c, d_i^L + c'), (e_i^U + c, d_i^U + c') | 1 \le i \le k\}$.

The simplification of the Post_δ operator is no restriction given the reset assumption we made in Section 3, we simply just allow Post_δ after a Post_e, which is, actually, how symbolic reachability is performed in tools such as Uppaal and Kronos. The Post operator as given above extends directly to multi-priced zones and the binary split in Post_e remains binary.

As shorthand notation, we write $(l, u, c_1, c_2) \in \mathsf{Post}_e(l, Z, C)$ to indicate that $(l, u, c_1, c_2) \in (l', Z', C')$ for some $(l', Z', C') \in \mathsf{Post}_e(l, Z, C)$.

Before we prove the soundness and completeness of the Post operator, we illustrate, in Figure 7, its behavior on the running example of Figure 1.

Lemma 3. *Given dual-priced symbolic state (l, Z, C) where $C = \{(e_1, d_1), \dots, (e_k, d_k)\}$ and $a \in \{e, \delta\}$ where $e = (l, g, \{x\}, (c, c'), l')$ we have*

$$(l', u', c_1', c_2') \in \mathsf{Post}_a(l, Z, C) \Longleftrightarrow$$
$$\exists (l, u, c_1, c_2) \in (l, Z, C) : (l, u, c_1, c_2) \xrightarrow{a} (l', u', c_1', c_2')$$

Proof. We choose only to prove the lemma for Post_e as the analogous proof for Post_δ is straightforward since each concrete successor has a unique concrete predecessor, given the requirement that Post_δ is always applied after a clock reset. We prove each direction of the bi-implication separately.

$\Longleftarrow$ - *Completeness*: Let $(l, u, c_1, c_2) \in (l, Z, C)$. The costs (c_1, c_2) are given as a convex combination of $C(u)$, i.e there are $0 \le \lambda_i \le 1$ and $\sum_i \lambda_i = 1$ for $1 \le i \le k$ such that:

$$(c_1, c_2) = \sum_i \lambda_i \cdot (e_i(u), d_i(u)). \qquad (7)$$

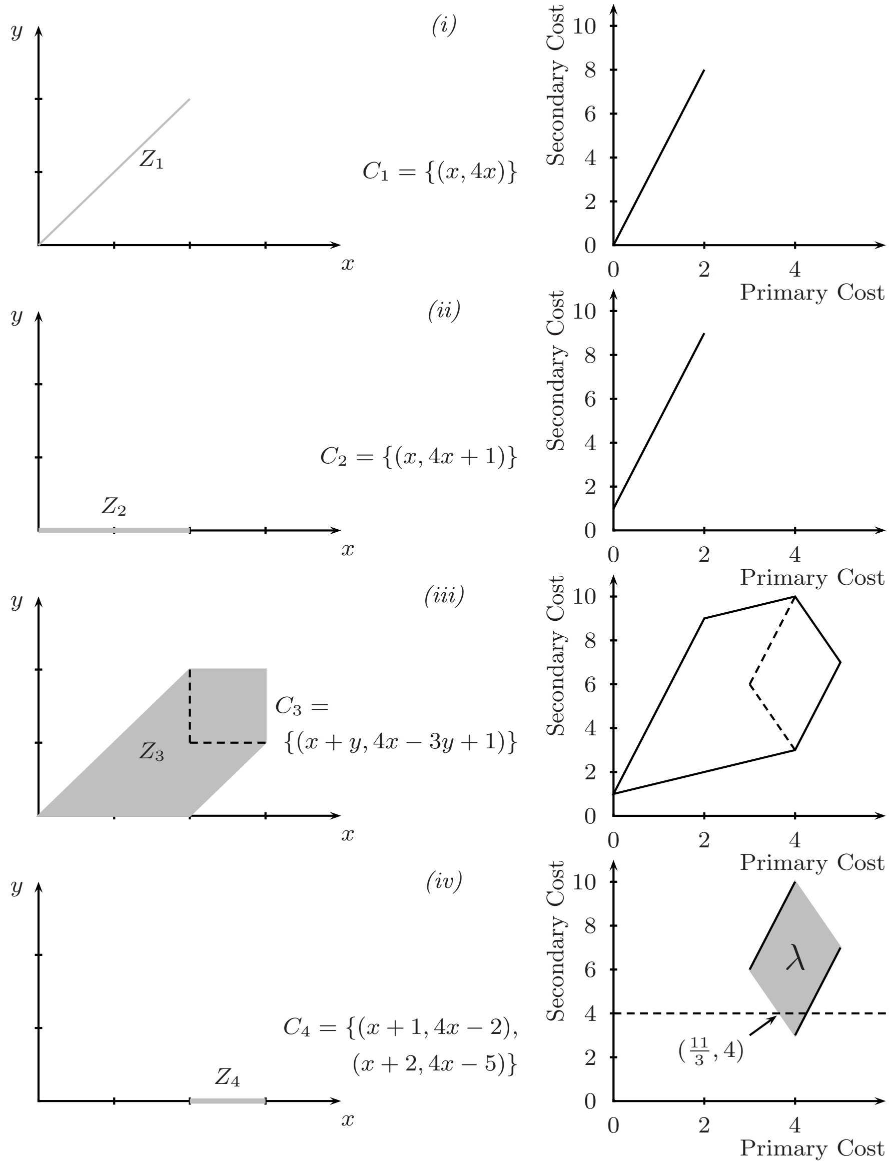

Fig. 7. Reachability analysis for $\mathsf{mincost}_{\leq 4}(\{l_3\})$ on the DPTA in Figure 1 starting from the initial state (l_1, Z_0, C_0). Areas inclosed by black lines in the cost part indicate all cost pairs computable from the cost functions. *(i)* $(l_1, Z_1, C_1) = \mathsf{Post}_\delta(l_1, Z_0, C_0)$ *(ii)* $(l_2, Z_2, C_2) = \mathsf{Post}_e(l_1, Z_1, C_1)$ where $e = (l_1, -, \{y\}, (0, 1), l_2)$ *(iii)* $(l_2, Z_3, C_3) = \mathsf{Post}_\delta(l_2, Z_2, C_2)$. The dashed area indicates the subset of the zone satisfying the guard of $e' = (l_2, x \geq 2 \wedge y \geq 1, \{y\}, (0, 0), l_3)$ *(iv)* $(l_3, Z_4, C_4) = \mathsf{Post}_{e'}(l_2, Z_3, C_3)$. The gray area in the cost part indicate the convex combinations between the lines describing the two cost functions. The cost pairs below the dashed line are the ones satisfying the constraint on the secondary cost. Note that $\mathsf{mincost}_{\leq 4}(\{l_3\}) = \frac{11}{3}$

The discrete successor of (l, u, c_1, c_2) with respect to e is given as $(l', u[x \rightarrow 0], c_1 + c, c_2 + c')$, which we will now prove is contained in $\mathsf{Post}_e(l, Z, C)$.
Let $(L, U) \in LUF_x(Z)$ such that $u[x \rightarrow 0] \in \{x\}L$. Given the convexity of zones there exist unique $v \in L$ and $w \in U$ where $v(x) \leq u(x) \leq w(x)$ and $u(y) = v(y) = w(y)$ for $y \neq x$, i.e. $u(x) = \alpha \cdot v(x) + (1 - \alpha) \cdot w(x)$ for some $0 \leq \alpha \leq 1$. Furthermore, the affinity of cost functions provide us with

$$(e_i(u), d_i(u)) = \alpha \cdot (e_i(v), d_i(v)) + (1 - \alpha) \cdot (e_i(w), d_i(w)), \tag{8}$$

for all $1 \leq i \leq k$ and the same α as above.
Now, choose $(l', u', c_1', c_2') \in \mathsf{Post}_e(l, Z, C)$ where $u' = u[x \rightarrow 0]$ and (c_1', c_2') is given by (9), which we can rewrite as:

$$\sum_i \lambda_i \cdot (\alpha \cdot (e_i^L(u') + c, d_i^L(u') + c') + (1 - \alpha) \cdot (e_i^U(u') + c, d_i^U(u') + c')) \tag{9}$$

$$= (c, c') + \sum_i \lambda_i \cdot (\alpha \cdot (e_i^L(u'), d_i^L(u')) + (1 - \alpha) \cdot (e_i^U(u'), d_i^U(u'))) \tag{10}$$

$$= (c, c') + \sum_i \lambda_i \cdot (\alpha \cdot (e_i(v), d_i(v)) + (1 - \alpha) \cdot (e_i(w), d_i(w))) \tag{11}$$

$$= (c, c') + \sum_i \lambda_i \cdot (e_i(u), d_i(u)) = (c_1 + c, c_2 + c') \tag{12}$$

The step from (10) to (11) follows from the definition of e_i^L, d_i^L, e_i^U, and d_i^U, and the step from (11) to (12) uses (8). Thus, the discrete successor of each concrete state in (l, Z, C) is contained in $\mathsf{Post}_e(l, Z, C)$.
$\Longrightarrow$ - *Soundness*: Let $(l', u', c_1', c_2') \in \mathsf{Post}_e(l, Z, C)$ such that $u' \in \{x\}L$ for some $(L, U) \in LUF_x(Z)$. Assume that:

$$(c_1', c_2') = \sum_i \lambda_i \cdot (\alpha \cdot (e_i^L(u') + c, d_i^L(u') + c') + (1 - \alpha) \cdot (e_i^U(u') + c, d_i^U(u') + c')) \tag{13}$$

for some $0 \leq \alpha, \lambda_i \leq 1$ and $\sum_i \lambda = 1$.
Let $v \in L$ and $w \in U$ be the unique clock valuations in Z where $u'(y) = v(y) = w(y)$ for $y \neq x$. $u \in Z$ is then the unique clock valuation with $u(y) = \alpha \cdot v(y) + (1 - \alpha) \cdot w(y)$ for all y with the same α as above. Choose the cost pair $(c_1, c_2) = \sum_i \lambda_i \cdot (e_i(u), d_i(u))$. Now, $(l, u, c_1, c_2) \in (l, Z, C)$ and the proof of completeness gives us that $(l, u, c_1, c_2) \xrightarrow{e} (l', u', c_1', c_2')$.

Now, we have that all e-successors and only e-successors of concrete states in (l, Z, C) are in the subset of $\mathsf{Post}_e(l, Z, C)$ with costs that can be written according to (13). Since DPTA are a subset of linear hybrid automata, we know that e-successors maintain convexity. So, since (l, Z, C) is, by definition, convex we know that the set of concrete states $(l', u', c_1', c_2') \in \mathsf{Post}_e(l, Z, C)$ with costs according to (13) is convex. Lemma 1 now states that this set is identical to all concrete states in $\mathsf{Post}_e(l, Z, C)$.□

If allowing k-tuples of costs as opposed to pairs, the proof of Lemma 3 is analogous, whenever we choose concrete states using α and $(1 - \alpha)$, we instead use $\alpha_1, \ldots \alpha_k$ with $\sum_i \alpha_i = 1$.

Lemma 3 states that the properties of our proposed Post operator corresponds to the requirements of Post defined in Section 2.

6 Termination

In this section, we first define the ordering $\sqsubseteq$ on the structure of locations with dual-priced zones and then prove that it is a well-quasi order.

Note that given a zone, Z, with m corner points, any cost function, e, associated with Z can be represented as an element of $\mathbb{N}^m$ giving the cost at each of corner points since any corner point of a zone have integral values. Thus, we can view the set of cost function pairs, C, of a dual-priced symbolic state, (l, Z, C) as a subset of $2^{\mathbb{N}^m \times \mathbb{N}^m}$ if Z has m corner points, and whenever we refer to this representation, we write C_Z. Given a pair, $(\bar{e}, \bar{d})$, of m-vectors in C_Z, we write $\bar{e} \leq \bar{d}$, if $\bar{e}$ is component-wise less than or equal to $\bar{d}$.

Definition 5 ($\sqsubseteq$). *Given two dual-priced symbolic states (l, Z, C), (l', Z', C'), we write $(l, Z, C) \sqsubseteq (l', Z', C')$ iff (i) $l = l'$ (ii) $Z' \subseteq Z$ and (iii) for all $(\bar{e}', \bar{d}') \in C'_Z$, there exists a $(\bar{e}, \bar{d}) \in C_{Z \wedge Z'}$ such that $\bar{e} \leq \bar{e}'$ and $\bar{d} \leq \bar{d}'$.*

The order $\sqsubseteq$ on k-tuples of costs are defined analogously. Note that $(l, Z, C) \sqsubseteq (l', Z', C')$ implies that for all $u \in Z'$, $\lambda(C(u)) \preceq \lambda(C'(u))$, but not the reverse, i.e. our $\sqsubseteq$ is stronger than domination, however, the above definition suffices to guarantee termination.

Lemma 4. *$\sqsubseteq$ is a well-quasi ordering.*

The proof of Lemma 4 follows directly from the fact that $(\mathbb{N}, \leq)$ is a better-quasi ordering, [1], and better-quasi orderings are closed under Cartesian product and power sets, and, finally, better-quasi orderings imply well-quasi orderings. For k-tuples of cost, the proof is identical as we consider k Cartesian products on $\mathbb{N}^m$ instead of pairs.

Now, we have fully instantiated the framework defined in Section 2 with syntax, data structures, a Post operator, and a well-quasi order. Based on this, we can conclude that, with this instantiation, the algorithm in Figure 3 computes optimal conditional reachability for DPTA. The result is summarized in the following theorem.

Theorem 1. *Optimal conditional reachability for DPTA is decidable.*

Along with the definitions of the framework of dual-priced transitions systems, DPTA, data structure for dual-priced symbolic states, the Post operator, and $\sqsubseteq$ we have discussed the straightforward extension to k-tuples of cost, and thus MPTA. Based on this we state the following corollary of Theorem 1.

Corollary 1. *Optimal conditional reachability for MPTA is decidable.*

7 Conclusion and Future Work

We have proven the decidability of optimal conditional reachability for multi-priced timed automata. The results are obtained from a zone-based algorithm for computing optimal conditional reachability which, in turn, might lead to an efficient implementation.

The example of Figure 1 illustrates that integral solution are not guaranteed, thus the immediate discrete time semantics for MPTA will not, in general, give correct results. However, discrete analysis of MPTA can be applied, but a correct time granularity must be chosen beforehand. In the case of Figure 1 a valid time granularity is $\frac{1}{3}$. However, a valid choice of granularity is non-trivial.

Except implementation of conditional reachability in the tool Uppaal, future research includes considering approximations along the lines of the ones proposed by Puri and Tripakis in [11]. Also, the complexity and efficiency of the algorithm in Figure 3 should be analyzed. Finally, related conditional reachability problems such as minimization under lower bound constraints and maximization under lower as well as upper bound constraints deserve investigation.

Acknowledgements. The authors would like to thank Stavros Tripakis for introducing them to multi-constrained optimization problems.

References

1. Parosh Aziz Abdulla and Aletta Nylén. Better is better than well: On efficient verification of infinite-state systems. In *Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science*, page 132. IEEE Computer Society, 2000.
2. R. Alur and D. Dill. Automata for modelling real-time systems. In *Proc. of Int. Colloquium on Algorithms, Languages and Programming*, number 443, pages 322–335, July 1990.
3. Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, pages 209–229, 1992.
4. Rajeev Alur, Salvatore La Torre, and George J. Pappas. Optimal paths in weighted timed automata. *Lecture Notes in Computer Science*, 2034:pp. 49–??, 2001.
5. M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine. Kronos: A model-checking tool for real-time systems. In A. J. Hu and M. Y. Vardi, editors, *Proc. 10th International Conference on Computer Aided Verification, Vancouver, Canada*, volume 1427, pages 546–550. Springer-Verlag, 1998.
6. Thomas Brihaye, Véronique Bruyére, and Jean-François Raskin. Model-checking weighted timed automata. In *the proc. of FORMATS'04*, Lecture Notes in Computer Science. Springer-Verlag, 2004.
7. Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.
8. Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. HYTECH: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, 1(1–2):110–122, 1997.

9. Kim Larsen, Gerd Behrmann, Ed Brinksma, Ansgar Fehnker, Thomas Hune, Paul Pettersson, and Judi Romijn. As cheap as possible: Efficient cost-optimal reachability for priced timed automata. *Lecture Notes in Computer Science*, 2102:pp. 493+, 2001.
10. Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a nutshell. *Int. Journal on Software Tools for Technology Transfer*, 1(1-2):134–152, 1997.
11. Anuj Puri and Stavros Tripakis. Algorithms for the multi-constrained routing problem. In *Proceedings of the 8th Scandinavian Workshop on Algorithm Theory*, pages 338–347. Springer-Verlag, 2002.