Revisiting Decidability and Optimum Reachability for Multi-Priced Timed Automata*

Martin Fränzle and Mani Swaminathan

University of Oldenburg
{fraenzle,mani.swaminathan}@informatik.uni-oldenburg.de

Abstract. We investigate the optimum reachability problem for Multi-Priced Timed Automata (MPTA) that admit both positive and negative costs on edges and locations, thus bridging the gap between the results of Bouyer et al. (2007) and of Larsen and Rasmussen (2008). Our contributions are the following: (1) We show that even the location reachability problem is undecidable for MPTA equipped with both positive and negative costs, provided the costs are subject to a bounded budget, in the sense that paths of the underlying Multi-Priced Transition System (MPTS) that operationally exceed the budget are considered as not being viable. This undecidability result follows from an encoding of Stop-Watch Automata using such MPTA, and applies to MPTA with as few as two cost variables, and even when no costs are incurred upon taking edges. (2) We then restrict the MPTA such that each viable quasi-cyclic path of the underlying MPTS incurs a minimum absolute cost. Under such a condition, the location reachability problem is shown to be decidable and the optimum cost is shown to be computable for MPTA with positive and negative costs and a bounded budget. These results follow from a reduction of the optimum reachability problem to the solution of a linear constraint system representing the path conditions over a finite number of viable paths of bounded length.

1 Introduction and Related Work

Formal models for hard real-time systems, paired with automatic analysis procedures determining their dynamic properties, are being considered as a means for rigorously ensuring that such systems function as desired. The classical *Timed Automaton* (TA) [1] has emerged as a well-studied model in this context. The (un-)decidability frontier between TA, for which location reachability and related properties are decidable, and Linear Hybrid Automata (LHA) [2], for which these properties happen to be undecidable, has been investigated through analysis of various moderate extensions of the original TA framework. Some of these extensions are interesting in their own right, as they provide valuable enhancements

* This work has been partially funded by the German Research Council (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS, www.avacs.org).

J. Quaknine and F. Vaandrager (Eds.): FORMATS 2009, LNCS 5813, pp. 149–163, 2009.

to the expressiveness of the TA framework, thus enabling the analysis and optimization of phenomena such as scheduling, which are beyond the scope of TA.

One such extension is that of Linear Priced Timed Automata ((L)PTA) or, synonymously, Weighted Timed Automata [3,4,5] for modelling real-time systems subject to some *budgetary constraints* on resource consumption. LPTA have -in addition to the real-valued *clocks* of classical TA - a *cost-function* mapping locations and edges to non-negative integers, whereupon a certain cost is incurred by staying in a location, or by taking an edge. The *minimum (infimum) cost reachability problem* for LPTA computes the minimum (infimum) cost of reaching a given *goal-location.* The minimum / infimum cost reachability problem for LPTA has been shown to be decidable and computable [3,4,5], leading to efficient tool-support through UPPAAL CORA along with applications to real-time scheduling [6]. A key factor for the decidability of location reachability in LPTA is that the cost variable is a monotonically increasing *observer* in the following sense: the cost variable cannot be reset, and testing the cost is forbidden in both guards of edges and invariants of locations, thereby restricting the expressive power of the model wrt. LHA, or equivalently, wrt. Stop-Watch Automata (SWA) [7]. This preservation of decidability has attracted an immense amount of research on (L)PTA in recent years (see [8] and Chapter 5 of [9] for surveys), among which we take a closer look at the following enhancements to the original LPTA model:

- The *optimum reachability problem* is considered in [10] for LPTA having a single cost variable, with *both positive and negative integer costs* being allowed on edges and locations. The optimality here refers to the computation of both *infimum and supremum* cost, which is shown to be PSPACE-COMPLETE, with optimum paths of the underlying transition system consisting of time-transitions occurring at time instants arbitrarily close to integers.
- The optimum (conditional) reachability problem for *Multi-Priced Timed Automata* (MPTA) with multiple cost variables is considered in [11], with only *non-negative costs* being allowed on edges and locations. The decidability of the minimum- and maximum- cost reachability problems is shown through exact symbolic (zone-based) algorithms that are guaranteed to terminate. Termination of the symbolic algorithm for computing the maximum cost reachability is subject to a divergence condition on costs, where the accumulation of each of the costs diverges along all infinite paths of the underlying Multi-Priced Transition System (MPTS).
- MPTA with *both positive and negative costs* are considered in [12]. More specifically, [12] investigates *Dual-Priced Timed Automata (DPTA)* with two observers (one observer termed as cost and the other as reward) in the context of optimum infinite scheduling, where the reward takes on only non-negative rates and is “strongly diverging” in the following sense: the accumulated reward diverges along every infinite path in the underlying transition system of the equivalent closed DPTA (obtained as usual by

making all inequalities in guards and invariants non-strict). There are no such restrictions on the cost observer, which can take on both positive and negative values. Optimum infinite schedules (that minimize or maximize the cost/reward ratio) are shown to be computable for such DPTA via *corner-point abstractions*.

Nevertheless, an exact characterization of the conditions for (un-)decidability and computation of the *optimum reachability problems* for MPTA having both positive and negative costs remains - to the best of our knowledge - unclear. We therefore attempt here to bridge the gap between the results of [10] and [11] by formulating conditions for (un-)decidability and computatibility wrt. the optimum reachability problems for such MPTA, through the following contributions:

1. We first show that *Stopwatch Automata* (SWA) [13,14] can be encoded using MPTA with two cost variables per stopwatch, allowing both positive and negative costs on edges and locations, with each of the costs being subject to individual upper and lower bounds that are to be respected along all viable paths of the underlying transition system. Since location reachability is undecidable in SWA with just one stop-watch, an immediate consequence of such an encoding is that even location reachability becomes undecidable for DPTA with two cost variables, admitting both postive and negative costs in locations and edges. Moreover, this undecidability result holds even when no costs are charged upon taking edges.
2. We then consider MPTA with both positive and negative costs on locations and edges, with individual bounds on each cost variable, and restrict the underlying MPTS such that a minimum absolute cost is incurred along all quasi-cyclic viable paths. Under such a restriction, we show that the reachability problem is decidable and that the optimum cost is computable for such MPTA. These results are derived from a reduction of the optimum reachability problem to the solution of a linear constraint system representing the path conditions over a finite number of viable paths, with the finiteness here being obtained from the boundedness and the (quasi-)cycle conditions on the costs.

Our contributions may thus be viewed as an additional step towards the precise characterization of the (un-)decidability frontiers between various semantic models for richer classes of real-time systems.

The remainder of this paper is organized as follows: Section 2 introduces MPTA and MPTS. Section 3 illustrates the encoding of SWA through MPTA admitting both positive and negative (bounded) costs in locations / edges, thereby demonstrating that even location reachability is undecidable for such MPTA with as few as two cost variables. Section 4 describes the computation of optimum cost for MPTA with both positive and negative costs in locations and edges, but subject to the cost boundedness and (quasi-)cycle conditions mentioned above. Section 5 concludes the paper along with directions for future research.

2 Multi- Priced Timed Automata (MPTA)

The notation and definitions used in this section partly mirror those in [11,12]. Given a finite set C of *clocks*, a *clock valuation* over C is a map $v : C \to \mathbb{R}_{\geq 0}$ that assigns a non-negative real value to each clock in C. If n is the number of clocks, a clock valuation is basically a point in $\mathbb{R}^n_{\geq 0}$, which we henceforth denote by $\boldsymbol{u}, \boldsymbol{v}$ etc.

Definition 1. *A* zone *over a set of clocks C is a system of constraints defined by the grammar $g ::= x \triangleright d \mid x - y \triangleright d \mid g \wedge g$, where $x, y \in C$, $d \in \mathbb{N}$, and $\triangleright \in \{<, \leq, >, \geq\}$. The set of zones over C is denoted $Z(C)$.*

A *closed zone* is one in which $\triangleright \in \{\leq, \geq\}$, and we denote the set of closed zones over C by $Z_c(C)$. A zone with no bounds on clock differences (i.e., with no constraint of the form $x - y \triangleright d$) is said to be *diagonal-free*, and we denote the corresponding set of zones by $Z_d(C)$. The set $Z_{cd}(C)$ denotes zones that are *both closed and diagonal-free.* The set $Z_{cdU}(C)$ denotes the set of *closed, diagonal-free zones having only upper bounds on the clocks.*

Definition 2. *An MPTA is a tuple $A = (L, C, (l_0, \mathbf{0}), E, I, \boldsymbol{P})$, with*

- *a finite set L of* locations *and a finite set C of* clocks*, with $|C| = n$.*
- *An* initial *location $l_0 \in L$ together with the initial clock-valuation $\mathbf{0}$ where all clocks are set to 0.*
- *a set $E \subseteq L \times Z_{cd}(C) \times 2^C \times L$ of possible* edges *between locations. An edge $e = (l, g, Y, l')$ between two locations l and l' is denoted $l \xrightarrow{e} l'$, and involves a* guard *$g = G(e) \in Z_{cd}(C)$, a* reset set *$Y = Res_e \subseteq C$.*
- *$I : L \to Z_{cdU}(C)$ assigns* invariants *to locations*
- *$\boldsymbol{P}$ is an indexed set of* prices *$\{p_1, \ldots, p_{n'}\}$ where each $p_i : (L \cup E) \to \mathbb{Z}$ assigns* price-rates *to locations and* prices *(or* costs*) to edges*

In the sequel, we will denote by m the *clock ceiling* of the MPTA A under investigation, which is the largest constant among the clock constraints of A. For ease of presentation, we assume that the guards and invariants of the automaton are *closed and diagonal-free zones*. We further assume that the clock-values are upper-bounded by m through the invariants at each location. These are not real restrictions, as every (P)TA can be transformed into an equivalent bounded and diagonal-free one (as in Section 5.3 of [10]). Boundedness likewise does not confine the expressiveness of Stop-Watch Automata discussed in the next section.

The concrete semantics of such an MPTA is given by a corresponding *Multi-Priced Transition System (MPTS)* with states $(l, \boldsymbol{u}) \in (L, \mathbb{R}^n_{\geq 0})$ where $\boldsymbol{u} \models I(l)$, with initial state $(l_0, \mathbf{0})$, and a transition relation $\to$ defined as follows:

- *Time-transitions*: $(l, \boldsymbol{u}) \xrightarrow{\delta, \boldsymbol{c}} (l, \boldsymbol{v})$ if $\boldsymbol{c} = \boldsymbol{P}(l) \cdot \delta$ and $\forall 0 \leq t \leq \delta : \boldsymbol{u} + t \models I(l)$ where $\boldsymbol{u} + t$ denotes the addition of t to each component of $\boldsymbol{u}$.
- *Switch-Transitions*: $(l, \boldsymbol{u}) \xrightarrow{e, \boldsymbol{c}} (l', \boldsymbol{v})$ if $\exists e = (l, g, Y, l') \in E : \boldsymbol{u} \models g, \boldsymbol{v} = [Y \leftarrow 0]\boldsymbol{u}$, $\boldsymbol{v} \models I(l'), \boldsymbol{c} = \boldsymbol{P}(e)$

Definition 3. *A* canonical initialized path *[10] π of an MPTS is a (possibly infinite) sequence of states s_i (each state s_i being a location-plus-clock-valuation pair of the form $(l, \boldsymbol{u})$), which hich starts from the initial state and alternates between time- and switch-transitions $\pi = s_0 \xrightarrow{\delta,\boldsymbol{c^0}} s_1 \xrightarrow{e_1,\boldsymbol{c^1}} s_2 \ldots$. The set of all possible canonical initialized paths is denoted Π. For a finite path $\pi in \Pi$ of length $|\pi| = k$, its* accumulated cost-vector *is defined as: $\boldsymbol{Cost(\pi)} = \sum_{i=0}^{k-1} \boldsymbol{c^i}$, with the summation here being performed component-wise for each cost-vector $\boldsymbol{c^i}$, with $Cost_j(\pi) = \sum_{i=0}^{k-1} c_j^i$ for each cost-component.*

For $\pi \in \Pi$, let π_k denote its finite prefix of length k. Then the corresponding accumulated cost along π is given by $\boldsymbol{Cost(\pi)} = lim_{k\to\infty}\boldsymbol{Cost(\pi_k)}$, if the latter exists.

The accumulated cost of $\pi \in \Pi$ wrt. a (set of) goal *state(s) G is defined as:*

$$\boldsymbol{Cost_G(\pi)} = \begin{cases} \infty & \text{if } \forall i \geq 0 : s_i \notin G \ , \\ \sum_{i=0}^{k} \boldsymbol{c^i} & \text{if } \exists k \geq 0 : (s_k \in G \land \forall i < k : s_i \notin G) \ . \end{cases}$$

$\boldsymbol{Cost_G(\pi)}$ for $\pi \in \Pi$ therefore yields the accumulated cost-vector along the shortest prefix of π ending in a goal state.

Cost-Boundedness Constraint. We assume in this paper that the permissible *cost charging is bounded by budgetary constraints*, in the sense that paths of the MPTS exceeding this budget (e.g., exhausting the battery capacity) are considered unviable and thus irrelevant to the optimization problem, even if the budget is exceeded only temporarily. The *budgetary constraint* is given formally as follows: For each cost variable, there is a lower bound $L_j \in \mathbb{Z}_{\leq 0}$ and an upper bound $U_j \in \mathbb{Z}_{\geq 0}$ which all viable paths have to obey throughout. Thus, a path π is called *viable* iff

$$\forall \pi' \text{ non-empty canonical prefix of } \pi : \forall j \in \{1, \ldots, n'\} : L_j \leq Cost_j(\pi') \leq U_j$$

holds.

We further designate Ω as the linear objective function that we wish to *optimize* wrt. reaching a set of *goal locations* under such budgetary constraints. Ω can be an arbitrary linear combination of prices drawn from $\boldsymbol{P}$. The objective of this paper is to formulate the conditions for (un-)decidability of reaching G under the budgetary constraints and for computatibility of the minimum value of Ω when viably reaching G. We call the latter the optimum-cost reachability problem, formally given below.

Problem 1. Given an MPTA $A = (L, C, (l_0, \mathbf{0}), E, I, \boldsymbol{P})$ having a set Π of canonical initialized paths in its corresponding MPTS, and given a set $G \subseteq L$ of goal locations plus a linear objective function Ω, as well as budgetary constraints (L_j, U_j) for the accumulation of each cost function p_j along all viable paths in Π, the *optimum-cost reachability problem* is to compute

$$\min\{\Omega(\boldsymbol{Cost_G}(\pi)) \mid \pi \in \Pi, \pi \text{ viable}\} \ .$$

Note that as Ω is an arbitrary linear combination of the prices accumulated in A, this problem — despite being formulated as a minimization problem — incorporates maximum-cost reachability also.

MPTA having two cost variables only are termed *Dual-Priced Timed Automata (DPTA)* in the remainder. We now proceed to show in Section 3 that DPTA with a boundedness condition on costs as above can be used to encode Stop-Watch Automata (SWA) with one stopwatch, for which even location reachability is undecidable [14]. It therefore follows that Problem 1 is undecidable for such cases. We however show in Section 4 that Problem 1 is decidable and computable even for MPTA when one imposes suitable conditions on viable quasi-cyclic paths of the corresponding MPTS.

3 Encoding of Stop-Watch Automata Using Bounded MPTA

Stopwatch automata (SWA) are an extension of timed automata where advance of individual clocks can be stopped in selected locations. It has been shown in [13,14] that location reachability is undecidable even for simple SWA (in the sense of all clock constraints being diagonal-free), and even when both the clocks and the stopwatches are confined to bounded range. The result, which is based on encoding two-counter machines, applies to SWA as small as a single stopwatch and four clocks. In the sequel, we will provide an encoding of stopwatch automata with n bounded clocks and n'' bounded stopwatches by MPTA with $n+1$ clocks and $2n''$ cost variables. This shows that location reachability is undecidable for bounded dually priced 5-clock MPTA.

As it suffices for our undecidability result, and as the generalization is straightforward, we demonstrate our reduction on 1-stopwatch SWA only. Let $m \in \mathbb{N}$ be a common upper bound on all clocks $C = \{x_1, \ldots, x_n\}$ and the single stopwatch sw occurring in the SWA A, i.e. m dominates the individual range bounds on clocks and sw. We construct an MPTA with two cost variables s and S, both with bounded range $[0, 2m]$, and $n+1$ (bounded) clocks $C \cup \{h\}$, where h is a fresh helper clock. W.l.o.g, we assume that the SWA A to be encoded does not contain guard conditions on its stopwatch, as these can always be replaced by invariants imposed in urgent transient states.

The central idea of the encoding is that s watches the lower bounds while S watches the upper bounds imposed on sw. Therefore,

1. the prices s and S do generally evolve with the same rate as the stopwatch sw they simulate,
2. $s \leq S$ holds throughout,
3. when sw is subject to an invariant imposing a lower bound of $l \geq 0$ then $s = sw - l$,[1]

[1] Note that whenever there is no explicit invariant enforcing a stronger lower bound, sw still is subject to the invariant $sw \geq 0$. Moreover, sw is always subject to an upper bound as it is generally confined to the range $[0, m]$. The invariants $s \geq 0$ (uniform over all locations) and $sw \geq l$ mutually reinforce each other.

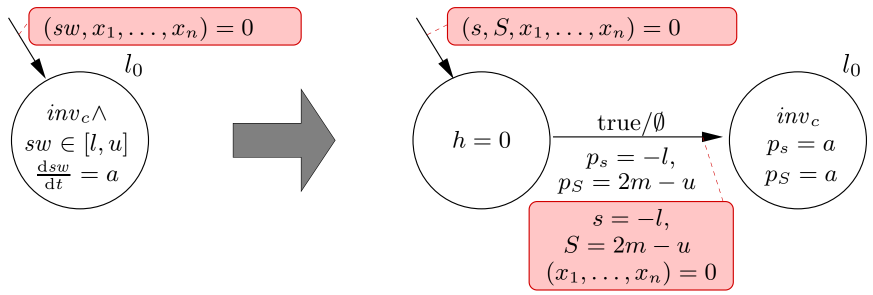

Fig. 1. Initializing the cost variables s and S simulating the stopwatch such that they enforce the invariant on the stopwatch sw. Here and in the remainder, inv_c (as well as inv'_c) refers to the parts of the invariant not dealing with the stopwatch. a is the slope of the stopwatch in l_0, which can be 0 or 1. Here and in all subsequent figures, formulae in shaded boxes are not part of the automaton, but collect invariant properties of the MPTS guaranteed along simulating runs.

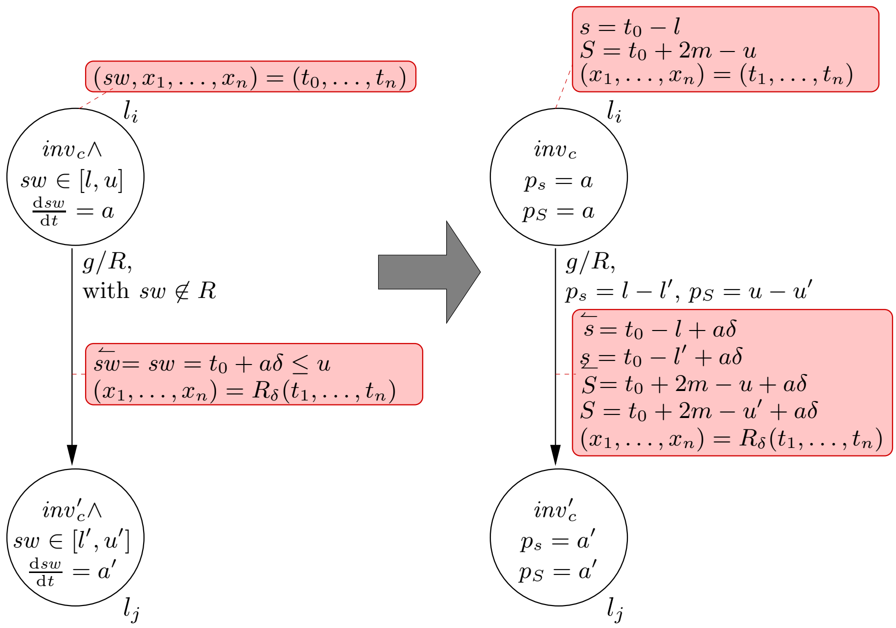

Fig. 2. Implementing change of invariant on the stopwatch in case the stopwatch is not reset by the switch transition. W.l.o.g., we assume that the guard g does not mention the stopwatch, as the pertinent conditions can be moved to invariants. $\overleftarrow{x}$ denotes the value of x before the transition while x denotes its value thereafter. δ represents the time spent in l_i and $R_\delta(t_1, \ldots, t_n)$ abbreviates the result of applying the reset R to $(x_1, \ldots, , x_n) = (t_1 + \delta, \ldots, t_n + \delta)$.

4. when sw is subject to an invariant imposing an upper bound of $u \leq m$ then $S = sw + 2m - u$.

Note that maintaining properties 3 and 4 leads to the bound $[0, 2m]$ on s and S enforcing the original invariant on sw, as $s = sw - l \wedge s \geq 0$ implies $sw \geq l$ and $S = sw + 2m - u \wedge S \leq 2m$ implies $sw \leq u$. Thus, the general bounds on s and S enforce the invariants on sw without any need for explicit invariants on s and S. All that has to be done is to, first, initialize s and S such that 3 and 4 hold, which is achieved by replacing the initial state by two states as in Fig. 1 and, second, update them accordingly upon a change of the invariant mediated by a location change in the SWA, which is shown in Fig. 2. Note that in both cases in accordance with property 1, the cost rates p_s and p_S coincide to the slope a of sw.

Resetting the stopwatch requires a slightly more complex construction, as we need to force s to value 0 (assuming that the invariant of the location following the reset does not enforce a lower bound on the just reset stopwatch, which would render the switch transition infeasible) and S to value $2m - u'$, where u' is the upper bound on sw in the target location of the resetting switch transition. To achieve this, we simulate the (instantaneous) switch transition by a run of duration $2m$. Within this run, we let s and S run to value m, which we test by substracting $-2m$ in a subsequent switch transition. We then adjust the values as desired. Furthermore, we employ the wrapping automaton construction of [13,14] to preserve the clock values. The complete automaton fragment is depicted in Fig. 3.

Glueing together the above MPTA fragments at the like-named locations, one obtains an MPTA which is equivalent to the encoded SWA wrt. location reachability. Due to the undecidability of location reachability for SWA [13,14], this reduction yields the following result:

Theorem 1. *Location reachability is undecidable for MPTA with $n \geq 1$ clocks and $\max(2, 14 - 2n)$ bounded cost variables. In particular, it is undecidable for MPTA with 6 clocks and 2 bounded cost variables, as well as for 1 clock and 12 bounded cost variables.*

Proof. The invariance properties mentioned in the shaded boxes in Figures 1 to 3, which are straightforward to establish based on the semantics of stopwatch automata and MPTA, show that the stopwatch automaton A has a path reaching location l_i with clock readings $(x_1, \ldots, x_n) = (t_1, \ldots, t_n)$ and stopwatch reading t_0 iff the encoding MPTA M has a viable path reaching location l_i with the same clock readings $(x_1, \ldots, x_n) = (t_1, \ldots, t_n)$ and costs $s = t_0 - l$ and $S = t_0 + 2m - u$. Hence, A can reach a given target location l_{target} iff M can reach the corresponding location.

According to [13,14], location reachability is undecidable for simple SWA with bounded clocks and stopwatches. The reduction to two-counter machines used in their proof yields SWA with five clocks and one stopwatch. As clocks are special cases of stopwatches, location reachability is thus undecidable for SWA with $n \geq 0$ clocks and $max(1, 6 - n)$ stopwatches. Our reduction encodes such SWA by a bounded MPTA with $n \geq 1$ clocks and $max(2, 14 - 2n)$ bounded prices. □

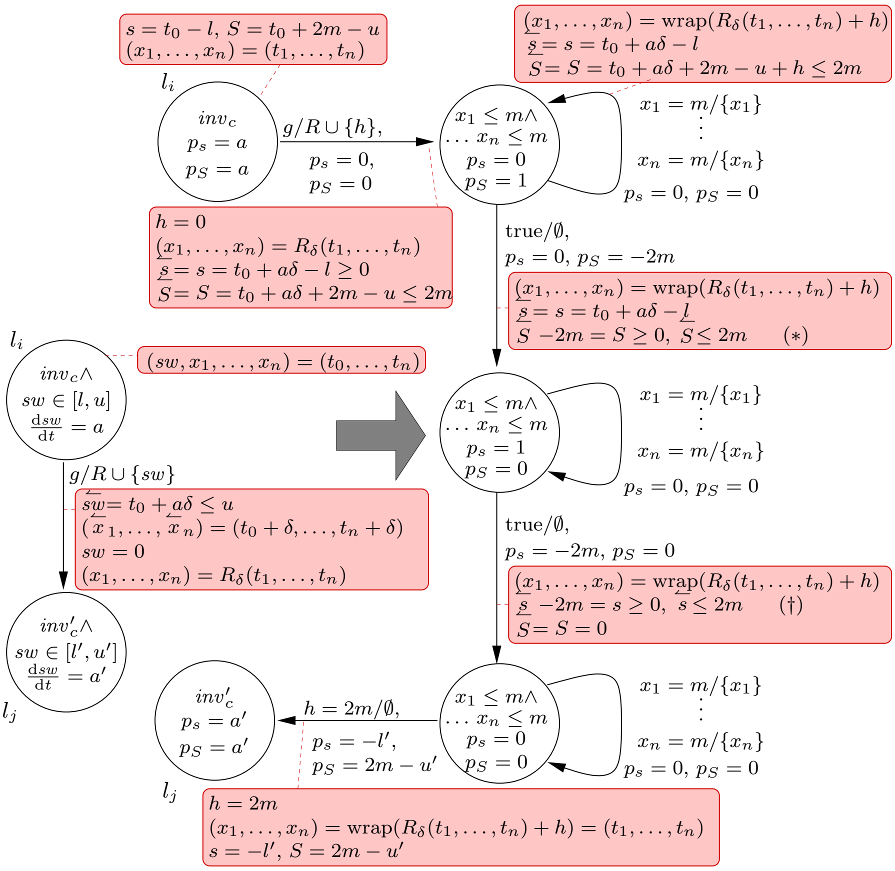

Fig. 3. Resetting the stopwatch, i.e. setting $s = -l$ and $S = 2m - u'$ while preserving the clocks. Here, $\mathrm{wrap}(x) := x - m\lfloor \frac{x}{m} \rfloor$. Note that $(*)$ and $(\dagger)$ imply $S = 0$ and $s = 0$.

An immediate consequence is

Corollary 1. *Optimum cost is not effectively computable for bounded MPTA.*

Proof. As the optimum cost is infinite iff the target location is unreachable, computing the optimum cost entails solving the reachability problem, which is undecidable for bounded PTA with more than one price according to Theorem 1. □

Note that these results can easily be strengthened wrt. the operations permitted on costs:

Corollary 2. *Location reachability remains undecidable and optimum cost uncomputable even if no costs can be charged on edges.*

Proof. It is straightforward to simulate an instantaneous price update by a durational update of fixed duration and slope using the wrapping construction to retain the clock values. □

When interpreting the above negative results concerning effectiveness, it should be noted, however, that the encoding relies crucially on closedness as well as on the universally binding character of the budgetary constraints. It is currently unclear whether similar undecidability results hold under an open (in the sense of strict bounds on the cost) budget or the permission to temporarily overdraw the budget by small amounts.

4 Optimum Cost Reachability for MPTA under Cost-Charging Quasi-Cycles

In this section, we investigate the optimum cost problem for bounded MPTA subject to the additional assumption that non-trivial cost is charged upon each quasi-cyclic viable path, where a quasi-cycle is a sequence of states returning close to its origin. This property is captured by the following definition:

Definition 4. *We call an MPTA* cost-charging on quasi-cycles *if there exists* $\varepsilon > 0$ *such that for each canonical viable path* $\pi = s_0 \xrightarrow{\delta_0, \mathbf{c^0}} s_1 \xrightarrow{e_1, \mathbf{c^1}} s_2 \ldots \xrightarrow{\delta_i, \mathbf{c}^i} s_i$ *with* $d(s_0, s_i) \leq \varepsilon$ *and path-length* $i \geq 2$*, i.e. the path contains at least one jump, it holds that* $|Cost_k(\pi)| \geq \varepsilon$ *for some cost-component. I.e., the MPTA is cost-charging on quasi-cycles iff there is no infinitesimally cheap return to a close vicinity of a state once this vicinity has been left.*

Hereby, we define the distance $d((l, \boldsymbol{u}), (l', \boldsymbol{u}'))$ *between two states* $(l, \boldsymbol{u})$ *and* $(\boldsymbol{l}', u)$ *to be*

$$d((l, \boldsymbol{u}), (l', \boldsymbol{u}')) = \begin{cases} \infty & \text{if } l \neq l', \\ ||\boldsymbol{u} - \boldsymbol{u}'|| & \text{if } l = l', \end{cases}$$

where $||\cdot||$ *is the maximum norm.*

Note that we neither demand a constant sign for the cost incurred nor fix the cost variable that incurs non-trivial cost upon quasi-return. Hence, some cost variable may well incur cost ε on some path from $(l, \boldsymbol{u})$ to some $(l, \boldsymbol{u}')$ in its ε-vicinity and cost $-\varepsilon$ on the same or another cost variable when proceeding from $(l, \boldsymbol{u}')$ to another $(l, \boldsymbol{u}'')$ in the ε-vicinity of $(l, \boldsymbol{u}')$.

Nevertheless, together with compactness of the state space as implied by boundedness, cost-charging on quasi-cycles is strong enough a condition of finiteness on all viable, i.e. bound-respecting, paths. [2]

Lemma 1. *Let A be a bounded MPTA which is cost-charging on quasi-cycles. Then the length of canonical viable paths in A is finitely bounded.*

[2] Note that any constant bound $\delta(\varepsilon) > 0$ suffices as a minimum lower bound on the absolute cost incurred along quasi-cyclic paths in order to obtain the finiteness condition. We have however chosen a single parameter ε for ease of presentation.

Proof. Let $\varepsilon > 0$ be the constant from Def. 4. Let L be the set of locations and $\{x_1, \ldots, x_n\}$ be the set of clocks in A and let D_i for $i = 1, \ldots, n$ be their respective bounded domains. Let $\{p_1, \ldots, p_{n'}\}$ be the set of cost variables in A and let P_i for $i = 1, \ldots, n'$ be their respective bounded domains. As the domains are bounded, the topological closure $\overline{V}$ of the combined clock-and-cost space $V = \prod_{i=1}^{n} D_i \times \prod_{j=1}^{n'} P_j$ is compact. Hence, V can only contain finitely many ε-separated points.

Let k be an upper bound on the maximum number of ε-separated points in V. Let $\pi = s_0 \xrightarrow{\delta_0, \boldsymbol{c}^0} s_1 \xrightarrow{e_1, \boldsymbol{c}^1} s_2 \ldots \xrightarrow{\delta_K, \boldsymbol{c}^K} s_K$ be a viable canonical path and let $\boldsymbol{C}^i = \sum_{j=0}^{i} \boldsymbol{c}^i$ be the accumulated costs until step i. As A is cost-charging on quasi-cycles, $||\boldsymbol{u}_i - \boldsymbol{u}_j|| \geq \varepsilon \vee ||\boldsymbol{C}_i - \boldsymbol{C}_j|| \geq \varepsilon$, which is equivalent to $||(\boldsymbol{u}_i, \boldsymbol{C}_i) - (\boldsymbol{u}_j, \boldsymbol{C}_j)|| \geq \varepsilon$, for each $i, j \leq K$ with $l_i = l_j$. As $(\boldsymbol{u}_i, \boldsymbol{C}_i) \in V$ for each $i \leq K$, it follows that $\forall l \in L : |\{(l_i, u_i, C_i) \mid i \leq K \wedge l_i = l\}| \leq k$. Consequently, $K \leq k \cdot |L|$ holds for the length K of the canonical viable path π. □

Given the fact expressed in Lemma 1 that all canonical viable paths have a uniform finite bound on their lengths, a consequence is that the optimum reachability problem for MPTA becomes an instance of *bounded model-checking* (BMC) [15] that is solvable for a rich class of hybrid systems. In particular, we encode the optimum bounded reachability problem up to depth k for MPTA as a Mixed Integer Linear Program (MILP). In the remainder of this section, we provide a corresponding algorithm, which is based upon reducing Problem 1 for cost-charging MPTA to a Mixed Integer Linear Program, along the lines of [16] illustrating BMC for acyclic LPTA.

This MILP will then be used twofold: First, as it expresses feasibility of a path of length k, versions with increasingly larger k will be used to determine the upper bound K on path length. Once this has been found, a version of depth K equipped with the cost term as an objective function will be used for determining the optimum cost.

Given an MPTA $A = (L, C, (l_0, \mathbf{0}), E, I, \boldsymbol{P})$, a set $G \subseteq L$ of goal-locations, and an arbitrary linear combination Ω of prices, we generate the following MILP for the BMC problem of depth k:

- For each discrete location $l \in L$ we take $k + 1$ zero-one variables l^i, where each l^i takes on either of the values 0 or 1, with $0 \leq i \leq k$. The value of l^i encodes whether A is in location l in step i as follows: $l^i = 1$ iff A is in location l in step i. Thus, for any $i \leq k$, there should be exactly one $l \in L$ such that $l^i = 1$, which can be enforced by requiring $\sum_{l \in L} l^i = 1$ in the MILP for each $i \in \{0, \ldots, k\}$.
- For each edge $e \in E$ we take k zero-one variables e^i, with $1 \leq i \leq k$. The value of e^i encodes whether A's ith move in the run was transition e. Again, one enforces that exactly one transition is taken in each step by adding the constraint $\sum_{e \in E} l^i = 1$ in the MILP for each $i \in \{1, \ldots, k\}$.
- For each clock $c \in C$ we take k real-valued variables c^i, with $0 \leq i \leq k - 1$. The value of c^i encodes c's value immediately *after* the ith transition in the run.

- For each $i \leq k$ we take one real-valued variable δ^i representing the time spent in the ith location along the run.
- We add constraints describing the initial state, i.e. enforcing $l_0^0 = 1$ and $c^0 = 0$ for each $c \in C$.
- We add constraints describing the relationship between discrete locations and transitions, i.e. guaranteeing that $e^i = 1$ implies $l^{i-1} = 1$ and $\widetilde{l}^i = 1$ for $(l, \widetilde{l}) \in e$. This can be encoded as a linear constraint via

$$l^{i-1} \geq e^i \ \wedge \ l^i \geq e^i \ .$$

- We add constraints enforcing the location invariants, i.e. checking for each $i \leq k$ that $l^i = 1 \implies I(l^i)[c_1^i, \dots, c_n^i / c_1, \dots, c_n]$ and that $l^i = 1 \implies I(l^i)[c_1^i + \delta^i, \dots, c_n^i + \delta^i / c_1, \dots, c_n]$, where $c_1, \dots, c_n$ are the clocks (i.e., $\{c_1, \dots, c_n\} = C$) and $\phi[y/x]$ denotes substitution of y for x in ϕ.

  As all clocks are bounded by m, the implications can be realized using the *switch variable encoding*. E.g., for an upper bound $x \leq u$ in the invariant, $m \cdot l_i + x^i + \delta^i \leq m + u$ implements the implication $l_i = 1 \implies x^i + \delta^i \leq u$.
- Using the same encoding, we add constraints enforcing guards, i.e. guaranteeing for each $0 \leq i \leq k-1$ that

$$e^{i+1} = \mathit{active} \implies g(e)[c_1^i + \delta^i, \dots, c_n^i + \delta^i / c_1, \dots, c_n] \ ,$$

  where $g(e)$ denotes the guard of edge e.
- We add constraints dealing with resets, i.e. enforcing for each $0 \leq i \leq k-1$ that

$$e^{i+1} = \mathit{active} \implies \begin{cases} c^{i+1} = c^i + \delta^i & \text{iff } c \notin Y(e) \ , \\ c^{i+1} = 0 & \text{iff } c \in Y(e) \ , \end{cases}$$

  where $Y(e)$ is the reset map associated to edge e.
- For each price variable $p \in \boldsymbol{P}$, we define $k+1$ auxiliary variables p_t^i recording the step price incurred by p in step $i \leq k$ and $k+1$ auxiliary variables p_d^i recording the price incurred by staying in the location during step $i \leq k$. Using the switch variable encoding, we enforce

$$\bigwedge_{l \in L} \bigwedge_{i=0}^{k} l_i = 1 \implies p_d^i = p(l) \cdot \delta^i$$

  and

$$p_t^0 = 0 \wedge \bigwedge_{e \in E} \bigwedge_{i=1}^{k} e_i = 1 \implies p_t^i = p(e) \ ,$$

  where $p : E \cup L \to \mathbb{Z}$ is the cost assignment of A.
- Adding k further variables p^i for each price $p \in P$, we can record the price $Cost_p$ accumulated so far by defining

$$p^0 = 0 \ ,$$
$$p^{j+1} = p^j + p_d^j + p_t^j$$

for each $j < k$. Viability of the paths is enforced by additionally demanding

$$\begin{array}{l} L_p \leq p^j \qquad \leq U_p \ , \\ L_p \leq p^j + p_d^j \leq U_p \end{array}$$

for each $j \leq k$.

This encoding, which is a standard MILP encoding suitable for bounded model-checking, can now be used in two ways:

1. For checking whether viable paths of length k exist, the above system is simply build for the desired depth k and checked for feasibility using an MILP solver. The resulting MILP is feasible iff paths of length $\geq k$ exist.
2. For determining the minimum cost for reaching the goal states within k steps, we, first, modify the goal states to become sinks by decorating them with cost-free and always enabled loops, second, build the above constraint system to depth k, third, add constraints enforcing the goal-locations to be visited by constraint

$$\sum_{i=0}^{k} \sum_{l \in G} l^i \geq 1$$

and, finally, use the linear expression

$$\Omega[\boldsymbol{P}^k/\boldsymbol{P}]$$

as an objective function to be minimized by the MILP solver. The MILP solver will either report the system to be infeasible, in which case the minimum cost of reaching G along canonical initialized paths of length at most k is infinite, or it will report the minimum cost of reaching G along canonical initialized paths of length at most k as the optimum value of its objective function. An optimum path can then be retrieved from the variables in the MILP that represent the MPTA state at the various steps.

Combining these two steps, we can solve Problem 1, i.e. the optimum-cost reachability problem, effectively by iteratively performing step 1 for increasing k until no viable path of length k^* exists and then performing step 2 for $k^* - 1$. Based on this procedure, we obtain the following positive result concerning effectiveness of cost optimal reachability in MPTA:

Theorem 2. *For bounded MPTA A which are cost-charging on quasi-cycles, the following two properties hold:*

1. *It is decidable whether A has a viable (i.e. obedient to the budgetary constraints) initialized path to some goal state.*
2. *The optimum cost for A reaching a goal state via a viable initialized path is computable for any linear cost function.* □

5 Conclusion and Future Research

We have investigated conditions for (un-)decidability and computability of the optimum reachability problem for MPTA admitting both positive and negative costs on locations and edges. Our encoding of SWA using cost-bounded MPTA however critically depends on the fact that the bounds on the cost variables are *closed* (i.e., non-strict), and that these bounds are to be respected for each cost variable along all viable paths of the underlying transition system. The question of whether this undecidability result holds even when cost bounds are strict, or when some path is allowed to temporarily overshoot the budget wrt. a cost variable, or when subject to other forms of budgetary constraints (as motivated, for instance, in [17]) remains currently open. Moreover, given that the undecidability result for cost-bounded MPTA holds for $n \geq 1$ clocks and $max(2, 14-2n)$ bounded cost variables (cf. Theorem 1), another natural question would be the validity of this result when one correspondingly restricts the number of clocks and bounded cost variables.

The cost-charging (quasi-)cycle assumption used to validate the BMC procedure of Section 4 is related to the divergence assumptions made in [11,12]. Due to the limited ability of TA to discriminate between states, as apparent from the region graph construction, it however seems that a few other cycle conditions may be equivalent to the above-mentioned condition on quasi-cycles, thus providing a potentially fruitful line of attack.

The presently proposed procedure of iteratively increasing the depth (for which BMC is performed) is not efficient, particularly for a small bound ε on the minimum cost incurred along quasi-cyclic viable paths. We therefore also plan to investigate efficient (symbolic) algorithms that exploit other realistic path conditions, and apply them to decision problems for the Duration Calculus [18], thereby fully implementing the decision procedure for a rich fragment of DC from [19] and extending it to an even richer subset, accommodating arbitrary linear combinations of durations. The latter requires algorithms for deciding budget-constrained reachability in MPTA with positive and negative cost rates, as negatively weighted durations map to prices with negative rates when extending the construction of [19].

Acknowledgements. We wish to thank Ernst-Rüdiger Olderog and the reviewers for useful comments that helped in improving the presentation of this paper.

References

1. Alur, R., Dill, D.: A Theory of Timed Automata. Theoretical Computer Science 126(2), 183–235 (1994)
2. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The Algorithmic Analysis of Hybrid Systems. Theoretical Computer Science 138(1), 3–34 (1995)
3. Alur, R., La Torre, S., Pappas, G.J.: Optimal paths in weighted timed automata. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 49–62. Springer, Heidelberg (2001)

4. Behrmann, G., Fehnker, A., Hune, T., Larsen, K.G., Pettersson, P., Romijn, J.M.T., Vaandrager, F.W.: Minimum-cost reachability for priced timed automata. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 147–161. Springer, Heidelberg (2001)
5. Larsen, K.G., Behrmann, G., Brinksma, E., Fehnker, A., Hune, T., Pettersson, P., Romijn, J.M.T.: As cheap as possible: Efficient cost-optimal reachability for priced timed automata. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 493–505. Springer, Heidelberg (2001)
6. Behrmann, G., Larsen, K.G., Rasmussen, J.I.: Optimal Scheduling using Priced Timed Automata. SIGMETRICS Performance Evaluation Review 32(4), 34–40 (2005)
7. Cassez, F., Larsen, K.G.: The impressive power of stopwatches. In: Palamidessi, C. (ed.) CONCUR 2000. LNCS, vol. 1877, pp. 138–152. Springer, Heidelberg (2000)
8. Bouyer, P.: Weighted Timed Automata: Model-Checking and Games. In: Proc. of MFPS 2006. ENTCS, vol. 158, pp. 3–17. Elsevier, Amsterdam (2006)
9. Bouyer, P.: From Qualitative to Quantitative Analysis of Timed Systems. Mémoire d'habilitation, Université Paris 7 (2009)
10. Bouyer, P., Brihaye, T., Bruyère, V., Raskin, J.-F.: On the Optimal Reachability Problem of Weighted Timed Automata. Formal Methods in System Design 31(2), 135–175 (2007)
11. Larsen, K.G., Rasmussen, J.I.: Optimal Reachability for Multi-Priced Timed Automata. Theoretical Computer Science 390(2–3), 197–213 (2008)
12. Bouyer, P., Brinksma, E., Larsen, K.G.: Optimal Infinite Scheduling for Multi-Priced Timed Automata. Formal Methods in System Design 32(1), 3–23 (2008)
13. Čerāns, K.: Algorithmic Problems in Analysis of Real Time System Specifications. PhD thesis, University of Latvia (1992)
14. Henzinger, T., Kopke, P., Puri, A., Varaiya, P.: What's Decidable about Hybrid Automata. Journal of Computer and System Sciences 57(1), 94–124 (1998)
15. Biere, A., Cimatti, A., Clarke, E.M., Strichman, O., Zhu, Y.: Bounded Model Checking. Advances in Computers 58, 118–149 (2003)
16. Panek, S., Stursberg, O., Engell, S.: Optimization of Timed Automata Models Using Mixed-Integer Programming. In: Larsen, K.G., Niebert, P. (eds.) FORMATS 2003. LNCS, vol. 2791, pp. 73–87. Springer, Heidelberg (2004)
17. Bouyer, P., Fahrenberg, U., Larsen, K.G., Markey, N., Srba, J.: Infinite runs in weighted timed automata with energy constraints. In: Cassez, F., Jard, C. (eds.) FORMATS 2008. LNCS, vol. 5215, pp. 33–47. Springer, Heidelberg (2008)
18. Chaochen, Z., Hoare, C.A.R., Ravn, A.P.: A Calculus of Durations. Information Processing Letters 40(5), 269–276 (1991)
19. Fränzle, M., Hansen, M.R.: Deciding an interval logic with accumulated durations. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 201–215. Springer, Heidelberg (2007)