Infinite Runs in Weighted Timed Automata with Energy Constraints
Patricia Bouyer, Uli Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, Jiří Srba

To cite this version:

HAL Id: hal-01194594
https://hal.archives-ouvertes.fr/hal-01194594
Submitted on 7 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Infinite Runs in Weighted Timed Automata with Energy Constraints

Patricia Bouyer\textsuperscript{1*}, Uli Fahrenberg\textsuperscript{2}, Kim G. Larsen\textsuperscript{2}, Nicolas Markey\textsuperscript{1*}, and Jiří Srba\textsuperscript{2**}

\textsuperscript{1} Lab. Spéci\c{s}ification et Vérification, CNRS & ENS Cachan, France
\texttt{\{bouyer,markey\}@lsv.ens-cachan.fr}
\textsuperscript{2} Dept. of Computer Science, Aalborg University, Denmark
\texttt{\{uli,kgl,srba\}@cs.aau.dk}

Abstract. We study the problems of existence and construction of infinite schedules for finite weighted automata and one-clock weighted timed automata, subject to boundary constraints on the accumulated weight. More specifically, we consider automata equipped with positive and negative weights on transitions and locations, corresponding to the production and consumption of some resource (e.g. energy). We ask the question whether there exists an infinite path for which the accumulated weight for any finite prefix satisfies certain constraints (e.g. remains between 0 and some given upper-bound). We also consider a game version of the above, where certain transitions may be uncontrollable.

1 Introduction

The overall motivation of the research underlying this paper is the quest of developing weighted (or priced) timed automata and games [3, 2] into a universal formalism useful for formulating and solving a broad range of resource scheduling problems of importance in application areas such as, e.g., embedded systems. In this paper we introduce and study a new resource scheduling problem, namely that of constructing infinite schedules or strategies subject to boundary constraints on the accumulation of resources. More specifically, we propose finite and timed automata and games equipped with positive as well as negative weights, respectively weight-rates. With this extension, we may model systems where resources are not only consumed but also occasionally produced or regained, e.g. useful in modelling autonomous robots equipped with solar-cells for energy-harvesting or with the ability to search for docking-stations when energy-level gets critically low. Main challenges are now to synthesize schedules or strategies that will ensure indefinite safe operation with the additional guarantee that energy will always be available, yet never exceeds a possible maximum storage capacity.

\textsuperscript{*} This author is partially supported by project DOTS (ANR-06-SETI-003).
\textsuperscript{**} This author is partially supported by grant no. MSM-0021622419.
As a basic example, consider the weighted timed automaton in Fig. 1a) with infinite behaviours repeatedly delaying in \(\ell_0\), \(\ell_1\) and \(\ell_2\) for a total of precisely one time-unit. The negative weights \((-3\) and \(-6) in \(\ell_0\) and \(\ell_2\) indicate the rates by which energy will be consumed, and the positive rate \(+6\) in \(\ell_1\) indicates the rate by which energy will be gained. Thus, for a given iteration the effect on the amount of energy remaining will depend highly on the distribution of the one time-unit over the three locations. Let us observe the effect of lower and upper constraints on the energy-level on so-called bang-bang strategies, where the behaviour remains in a given location as long as permitted by the given bounds. Fig. 1b) shows the bang-bang strategy given an initial energy-level of 1 with no upper bound (dashed line) or 3 as upper bound (solid line). In both cases, it may be seen that the bang-bang strategy yields an infinite behaviour.

In Fig. 1c) and d), we consider the upper bounds 2 and 1, respectively. For an upper bound of 2, we see that the bang-bang strategy reduces an initial energy-level of 1 to 1 (solid line), and yet another iteration will reduce the remaining energy-level to 0. In fact, the bang-bang strategy—and it may be argued, any other strategy—fails to maintain an infinite behaviour for any initial energy-level except for 2 (dashed line). With upper-bound 1, the bang-bang strategy—and any other strategy—fails to complete even one iteration (solid line).

We also propose an alternative \emph{weak} notion of upper bounds, which does not prevent energy-increasing behaviour from proceeding once the upper bound is reached but merely maintains the energy-level at the upper bound. In this case, as also illustrated in Fig. 1d) (dashed line), the bang-bang strategy is quite adequate for yielding an infinite behaviour.

In this paper, we ask the question whether some, or all, of the infinite paths obey the property that the weight accumulated in any finite prefix satisfies certain constraints. The \emph{lower-bound problem} requires the accumulated weight never
drop below zero, the interval-bound problem requires the accumulated weight to stay within a given interval, and in the lower-weak-upper-bound problem the accumulated weights may not drop below zero, with the additional feature that the weights never accumulate to more than a given upper bound; any increases above this bound are simply truncated. We also consider a game version of the above setting, where certain transitions may be uncontrollable.

For finite weighted automata, we show that the lower- and lower-weak-upper-bound problems are decidable in polynomial time. In the game setting we prove these problems to be \(\mathcal{P}\)-hard but decidable in \(\mathcal{NP} \cap \mathcal{coNP}\). The interval-bound problem is on the other hand \(\mathcal{NP}\)-hard but decidable in \(\mathcal{PSPACE}\) and in the game setting it is \(\mathcal{EXPTIME}\)-complete. For one-clock weighted timed automata, the lower- and lower-weak-upper-bound problems remain in \(\mathcal{P}\), but the interval-bound problem in the game setting is already undecidable.

Related Work. Recently, extensions of timed automata with costs (or prices) to measure quantities like energy, bandwidth, etc. have been introduced and subject to significant research. In these models—so-called priced or weighted timed automata—the cost can be used to measure the performance of a system allowing various optimization problems to be considered. Examples include cost-optimal reachability problem in a single-cost setting [2, 3], as well as in a multi-cost setting [20], or the computation of (mean-cost or discounted) cost-optimal infinite schedules [6, 16]. Also games where players try to optimize the cost have been considered [1, 7, 13, 5, 10], and several model-checking problems have been studied [12, 5, 9, 11]. It is worth noticing that in these last frameworks (model-checking and games), most of the problems are undecidable, and they can be solved algorithmically only for the (restricted) class of one-clock automata.

However, in the priced extensions considered so far, only non-negative costs are allowed restricting the types of quantities one can measure. The only exception concerns the optimal-cost reachability problem, which has been solved even when negative costs are allowed [4], but the proof is basically not much more involved than with only non-negative costs. In the current work, we pursue this line of research, and focus on non-trivial problems that can be posed on timed automata with arbitrary costs, which allow to model interesting problems on timed systems with (bounded) resources. Due to space limitations, most proofs had to be omitted in this paper. A long version is available as [8].

2 Models and Problems

2.1 Weighted Automata and Games

A weighted automaton is a tuple \(A = (S, s_0, T)\) consisting of a set of locations (or states) \(S\), an initial location (state) \(s_0 \in S\), and a set of transitions \(T \subseteq S \times \mathbb{R} \times S\). A transition \((s, p, s')\) is customarily denoted \(s \xrightarrow{p} s'\), where \(p \in \mathbb{R}\) is the weight of that transition. We implicitly consider only non-blocking automata where every location has at least one outgoing transition. If \(S\) and \(T\) are finite sets and \(T \subseteq S \times \mathbb{Z} \times S\) then \(A\) is called a finite weighted automaton.
A run in $A$ starting from $s \in S$ is a finite or infinite sequence $s = s_1 \xrightarrow{p_1} s_2 \xrightarrow{p_2} s_3 \xrightarrow{p_3} \cdots$. We write Runs($A, s$) for the set of all runs in $A$ starting from $s$. Given a finite run $\gamma = s_1 \xrightarrow{p_1} \cdots \xrightarrow{p_{n-1}} s_n$, we write last($\gamma$) to denote its final location $s_n$.

**Definition 1.** Let $A = (S, s_0, T)$ be a weighted automaton, $c \in \mathbb{R}$, $b \in \mathbb{R} \cup \{\infty\}$, and let $\gamma = s_1 \xrightarrow{p_1} \cdots \xrightarrow{p_{n-1}} s_n \in$ Runs($A, s_1$). The accumulated weight with initial credit $c$ under weak upper bound $b$ is $p_{c:b}(\gamma) = r_n$, where $r_1, \ldots, r_n \in \mathbb{R}$ are defined inductively by $r_1 = \min(c, b)$, $r_{i+1} = \min(r_i + p_i, b)$.

So for computing $p_{c:b}(\gamma)$, costs are accumulated along the transitions of $\gamma$, but only up to a maximum accumulated cost $b$; possible increases above $b$ are simply discarded. The case $b = \infty$ is special; we will denote $p_c(\gamma) = p_{c:\infty}(\gamma) = c + \sum_{i=1}^{n-1} p_i$. Also, we will write $p(\gamma) = p_0(\gamma)$.

A **weighted game** is a tuple $G = (S_1, S_2, s_0, T)$ where $S_1$ and $S_2$ are two disjoint sets of locations, and $A_G = (S_1 \cup S_2, s_0, T)$ is a weighted automaton. Note that we only introduce turn-based weighted games here.

A run of $G$ is a run of $A_G$, and we write Runs($G$) for Runs($A_G$). A strategy $\sigma$ for Player $i$, where $i = 1$ or $i = 2$, maps each finite run $\gamma$ ending in $S_i$ to a transition departing from last($\gamma$). Given a location $s$ in $G$ and a strategy $\sigma$ for Player $i$, an outcome of $\sigma$ from $s$ is any run $s_1 \xrightarrow{p_1} s_2 \xrightarrow{p_2} \cdots \xrightarrow{p_{n-1}} s_n$ starting in $s$ such that for any $k$, if $s_k \in S_i$, then $s_k \xrightarrow{p_k} s_{k+1} = \sigma(s_1 \xrightarrow{p_1} s_2 \xrightarrow{p_2} \cdots \xrightarrow{p_{k-1}} s_k)$.

We are now able to formulate the problems which we shall be concerned with.

The **lower-bound problem**: Given a weighted game $G$ and an initial credit $c$, does there exist a strategy for Player 1 such that any infinite outcome $\gamma$ from the initial state of $G$ has $p_{c}(\gamma) \geq 0$ for all finite prefixes $\gamma'$ of $\gamma$?

The **lower-weak-upper-bound problem**: Given a weighted game $G$, a weak upper bound $b$ and an initial credit $c \leq b$, does there exist a strategy for Player 1 such that any infinite outcome $\gamma$ from the initial state of $G$ has $p_{c:b}(\gamma') \geq 0$ for all finite prefixes $\gamma'$ of $\gamma$?

The **interval-bound problem**: Given a weighted game $G$, an upper bound $b$, and an initial credit $c \leq b$, does there exist a strategy for Player 1 such that any infinite outcome $\gamma$ from the initial state of $G$ has $0 \leq p_{c}(\gamma') \leq b$ for all finite prefixes $\gamma'$ of $\gamma$?

Note that by Martin’s determinacy theorem [21], our games are determined: Player 1 has a strategy for winning one of the above games if, and only if, Player 2 does not have a strategy for making Player 1 lose.

Special variants of the above problems are obtained when one of the sets $S_1$ and $S_2$ is empty. In case $S_2 = \emptyset$, they amount to asking for the existence of an infinite path adhering to the given bounds; in case $S_1 = \emptyset$, one asks whether all infinite paths stay within the bounds. The former problems will be referred to as existential problems, the latter as universal ones.

### 2.2 The Timed Setting

A **weighted timed automaton** is a tuple $A = (Q, q_0, C, I, E, \text{rate})$, with $Q$ a finite set of locations, $q_0 \in Q$ the initial location, $C$ a finite set of clocks, $I : Q \rightarrow \Phi(C)$
location invariants, \( E \subseteq Q \times \Phi(C) \times 2^C \times Q \) a finite set of transitions, and rate: \( Q \rightarrow \mathbb{Z} \) location weight-rates. Here the set \( \Phi(C) \) of clock constraints \( \varphi \) is defined by the grammar \( \varphi := x \varphi_1 \varphi_2 \mid \varphi_1 \wedge \varphi_2 \) with \( x \in C, k \in \mathbb{Z} \) and \( \sqsubseteq \in \{ \leq, <, \geq, >, = \} \). The semantics of a weighted timed automaton \( A \) is given by a weighted automaton \([A]\) with states \((q,v)\), where \( q \in Q \) and \( v: C \rightarrow \mathbb{R}_{\geq 0} \), and transitions of two types:

- **Delay transitions** \( (q,v) \xrightarrow{p\cdot t} (q,v+t) \) for some \( t \in \mathbb{R}_{\geq 0} \). Such a transition exists whenever \( v+\lambda t \models I(q) \) for all \( t' \in [0,t] \), and its weight is \( p = t \cdot \text{rate}(q) \).
- **Switch transitions** \( (q,v) \xrightarrow{e} (q',v') \), where \( e = (q,g,r,q') \) is a transition of \( A \). Such a switch transition exists whenever \( v \models g \wedge I(q) \), \( v' = v|_{r=0} \), and \( v' \models I(q') \). Switch transition have always the weight 0.

A run of \( A \) is a sequence \((q_1,v_1) \xrightarrow{p_1} (q_1,v'_1) \xrightarrow{0} (q_2,v_2) \xrightarrow{p_2} (q_2,v'_2) \cdots \) of alternating delay and switch transitions in \([A]\). We write \( \text{Runs}(A) \) for the set of all runs of \( A \). The accumulated weight (with initial credit \( c \) and weak upper bound \( b \)) for runs of a weighted timed automaton are given by Definition 1.

A weighted timed game is a tuple \( G = (Q,q_0,C,I,E_1,E_2,\text{rate}) \) such that \( A_G = (Q,q_0,C,I,E_1 \cup E_2,\text{rate}) \) is a weighted timed automaton. States and runs of a weighted timed game are those of the underlying weighted timed automaton, and we again write \( \text{Runs}(G) \) for \( \text{Runs}(A_G) \).

Given a weighted timed game \( G \), a strategy for Player \( i \) \((i \in \{1,2\})\) is a partial function \( \sigma \) mapping each finite run of \( \text{Runs}(A_G) \) to \( E_1 \cup \{\lambda\} \), where \( \lambda \notin E_1 \cup E_2 \) is the “wait” action. For a finite run \( \gamma \) ending in \((q,v)\), it is required that if \( \sigma(\gamma) = \lambda \) then \((q,v) \xrightarrow{p\cdot t} (q,v+t) \) for some \( t > 0 \), and if \( \sigma(\gamma) = e \) then \((q,v) \xrightarrow{0} (q',v') \).

Given a state \( s = (q,v) \) of a weighted timed game \( G \) and a strategy \( \sigma \) for Player \( i \), the set \( \text{Out}(\sigma,s) \) of outcomes of \( \sigma \) from \( s \) is the smallest subset of \( \text{Runs}(G,s) \) such that

- \((q,v)\) is in \( \text{Out}(\sigma,s) \);
- if \( \gamma \in \text{Out}(\sigma,s) \) is a finite run ending in \((q_n,v_n)\), then a run of the form \( \gamma \vdash (q_{n+1},v_{n+1}) \) is in \( \text{Out}(\sigma,s) \) if either
  - \((q_n,v_n) \xrightarrow{e} (q_{n+1},v_{n+1}) \) with \( e \in E_{3-i} \),
  - \((q_n,v_n) \xrightarrow{e} (q_{n+1},v_{n+1}) \) with \( e = \sigma(\gamma) \in E_i \), or
  - \((q_n,v_n) \xrightarrow{t} (q_{n+1},v_{n+1}) \) with \( t \in \mathbb{R}_{\geq 0} \), and \( \sigma(\gamma \vdash t') (q_n,v_n+t') = \lambda \) for all \( t' \in [0,t] \);
- an infinite run is in \( \text{Out}(\sigma,s) \) if all its finite prefixes belong to \( \text{Out}(\sigma,s) \).

A weighted timed game \( G = (Q,q_0,C,I,E_1,E_2,\text{rate}) \) is said to be turn-based if the set of edges leaving any \( q \in Q \) is a subset of either \( E_1 \) or \( E_2 \). In that case, the semantics of \( G \) is a (turn-based, infinite) weighted game as introduced in the previous section. However, weighted timed games introduced above are more general than turn-based ones, and we shall later use the more general notion.

For weighted timed games, we will be interested in the same three problems as in the previous section; the existence of a strategy whose outcomes are infinite runs remaining within given bounds. As in the previous section, we have the special cases of existential and universal problems.
3 Fixed-Point Characterization

Let us introduce some terminology. For the lower-bound problem, we say that an infinite path \( \gamma \) is \( c \)-feasible for some initial credit \( c < +\infty \), if \( p_c(\gamma') \geq 0 \) for all finite prefixes \( \gamma' \) of \( \gamma \), and that \( \gamma \) is feasible if it is \( c \)-feasible for some \( c \).

Given a weighted game \((S_1, S_2, s_0, T)\) and \( b \in \mathbb{R}_{\geq 0} \), we define three predicates \( L, W_b, U_b : S \times \mathbb{R}_{\geq 0} \to \{tt, ff\} \) to be the respective maximal fixed points to the following equations:

\[
\begin{align*}
L(s, c) &= 0 \leq c \land \begin{cases} 
  s \in S_1 \implies \exists s' \overset{p}{\rightarrow} s' \in T : L(s', c + p) \\
  s \in S_2 \implies \forall s' \overset{p}{\rightarrow} s' \in T : L(s', c + p)
\end{cases} \\
W_b(s, c) &= 0 \leq c \leq b \land \begin{cases} 
  s \in S_1 \implies \exists s' \overset{p}{\rightarrow} s' \in T : W_b(s', \max(b, c + p)) \\
  s \in S_2 \implies \forall s' \overset{p}{\rightarrow} s' \in T : W_b(s', \max(b, c + p))
\end{cases} \\
U_b(s, c) &= 0 \leq c \leq b \land \begin{cases} 
  s \in S_1 \implies \exists s' \overset{p}{\rightarrow} s' \in T : U_b(s', c + p) \\
  s \in S_2 \implies \forall s' \overset{p}{\rightarrow} s' \in T : U_b(s', c + p)
\end{cases}.
\]

The right-hand side of each of these equations defines a monotone functional on the power set lattice of \( S \times \mathbb{R}_{\geq 0} \), hence indeed the maximal fixed points exist. Also, if the weighted game under investigation is image finite, i.e. if the sets \( \{(p, s') \mid s \overset{p}{\rightarrow} s' \in T\} \) are finite for all states \( s \), then it can be shown that these respective functionals are continuous, implying that the maximal fixed points can be obtained as the limits of the iterated application of the respective functionals to the maximal element \( S \times \mathbb{R}_{\geq 0} \) of the power set lattice. The proof of the lemma below is immediate from the definition of the predicates.

**Lemma 2.** Let \((S_1, S_2, s_0, T)\) be a weighted game and \( s \in S_1 \cup S_2 \), \( b, c \in \mathbb{R}_{\geq 0} \). Then \( (s, c) \in L \) (or \( (s, c) \in W_b \), or \( (s, c) \in U_b \)) if and only if there exists a strategy for Player 1 such that any infinite path \( \gamma \) from \( s \) consistent with the strategy is \( c \)-feasible for the lower-bound problem (or lower-weak-upper-bound problem, or interval-bound problem, respectively).

For the lower-bound problems, the above fixed-point characterization can be stated in a different way by defining recursively the \( \text{infimum credits} \) sufficient for feasibility—note that such a notion does not make sense for the interval-bound problem, as here the set of sufficient credits is not necessarily upward-closed. For a given weighted game \((S_1, S_2, s_0, T)\) and \( b \in \mathbb{R}_{\geq 0} \), let \( L, W_b : S \to \mathbb{R} \cup \{\infty\} \) be the functions defined as respective minimal fixed points to the following equations:

\[
\begin{align*}
L(s) &= \begin{cases} 
  \min \{L(s') - p \mid s \overset{p}{\rightarrow} s'\} & \text{if } s \in S_1 \\
  \max \{L(s') - p \mid s \overset{p}{\rightarrow} s'\} & \text{if } s \in S_2
\end{cases} \\
W_b(s) &= \begin{cases} 
  \min \{b, \min \{W(s') - p \mid s \overset{p}{\rightarrow} s'\}\} & \text{if } s \in S_1 \\
  \min \{b, \max \{W(s') - p \mid s \overset{p}{\rightarrow} s'\}\} & \text{if } s \in S_2
\end{cases}.
\]

Proposition 3. We have $L(s,c)$ if and only if $c \geq L(s)$, and $W_b(s,c)$ if and only if $W_b(s) \leq c \leq b$.

The following lemma shows that for finite weighted games, there is a pre-described upper limit for the values of $L$ and $W_b$. This implies that the fixed-point computations can be terminated after a finite number of iterations. For timed games, the second part of the next example below shows that this is not necessarily the case.

Lemma 4. Let $(S_1,S_2,s_0,T)$ be a finite weighted game and $b \in \mathbb{R}_{\geq 0}$. Let $M = \sum \{-p \mid s \xrightarrow{p} s' \in T, p < 0\}$. Then for any state $s$, $L(s) < \infty$ if and only if $L(s) \leq M$, and $W_b(s) < \infty$ if and only if $W_b(s) \leq M$.

Example 5. Consider the weighted automata in Fig. 2a) and 2b). Let us attempt to compute $L$ using iterative application of the recursive definition. For a) we find the following fixed points after two iterations: $L(s_0) = 0$ and $L(s_1) = 1$. For b) we get the following sequence of decreasing approximations: $L_{2n}(s_0) = L_{2n+1}(s_0) = 2n$ and $L_{2n}(s_1) = L_{2n-1}(s_1) = 2n + 1$. Hence clearly $L(s_0) = L(s_1) = \infty$, though this fixed point is not reached within a finite number of iteration.

In Fig. 2c) we reconsider the weighted timed automaton from Fig. 1 under the interval-bound problem with upper bound 2. If we assume that after $n$ iterations of the fixed-point computation, $U_n^s(\ell_2,1,c) = tt$ if and only if $c \in [2 - \varepsilon, 2]$ for some $\varepsilon$, then $U_{n+1}^s(\ell_0,0,c) = tt$ if and only if $c \in [2 - \frac{1}{2}\varepsilon, 2]$. The largest fixed point—$U_n^s(\ell_2,1,c) = tt$ if and only if $c = 2$—is only reached as the limit of this infinite sequence of approximations.

4 Lower-Bound Problems

In this section we treat the lower-bound and lower-weak-upper-bound problems for finite automata, one-clock timed automata, and for finite games. For one-clock timed games, these problems are open.

4.1 Finite Weighted Automata

For a given finite weighted automaton $A = (S,s_0,T)$, we denote by $\text{MinCr}(\gamma)$, for a path $\gamma$ in $A$, the minimum $c \geq 0$ for which $\gamma$ is $c$-feasible, and by $\text{MinCr}(s)$,
for a state \( s \in S \), the minimum of \( \text{MinCr}(\gamma) \) over all feasible paths \( \gamma \) emerging from \( s \).

A cycle \( \gamma = s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_0 \) in \( A \) is non-losing if \( p(\gamma) \geq 0 \) (or equivalently, \( p_\epsilon(\gamma) \geq c \) for any initial credit \( c \)). A lasso \( \lambda \) from a given state \( s_0 \) is an infinite path of the form \( \gamma_1(\gamma_2)\gamma_3 \), where \( \gamma_1 = s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_{i-1} \), \( \gamma_2 = s_i \rightarrow \cdots \rightarrow s_k \rightarrow s_i \), and \( s_{i-1} \rightarrow s_i \) are paths in \( A \), and with \( i \neq j \) whenever \( i \neq j \leq k \). It is clear that if a lasso \( \gamma_1(\gamma_2)\gamma_3 \) constitutes a feasible path then the cycle \( \gamma_2 \) must be non-losing.

**Lemma 6.** Let \( A = (S,s_0,T) \) be a finite weighted automaton and \( s \in S \). For any feasible path \( \gamma \) from \( s \), there exists a feasible lasso \( \lambda \) also from \( s \) such that \( \text{MinCr}(\lambda) \leq \text{MinCr}(\gamma) \).

**Proof.** Let \( c = \text{MinCr}(\gamma) \), and assume first that there is a cycle \( \pi \) in \( \gamma \) such that \( p(\pi) \geq 0 \). Write \( \gamma = \gamma_1 \pi \gamma_2 \), then \( \lambda = \gamma_1 \pi \gamma_2 \) is a feasible lasso from \( s \), and \( \text{MinCr}(\lambda) \leq \text{MinCr}(\gamma) \).

Assume now that there is no cycle \( \pi \) in \( \gamma \) with \( p(\pi) \geq 0 \). We can write \( \gamma = \gamma_1 \gamma_2 \), where \( \gamma_2 \) only takes transitions that appear infinitely often along \( \gamma \). As there is no cycle with non-negative weight in \( \gamma \), the weights of prefixes of \( \gamma_2 \) decrease to \( -\infty \). Hence there is a prefix \( \gamma_2' \) of \( \gamma_2 \) such that \( p(\gamma_2') < -p(\gamma_1) \). But then the accumulated weight of \( \gamma_1 \gamma_2' \), which is a prefix of \( \gamma \), is negative, which contradicts the feasibility of \( \gamma \).

Thus, determining \( L(s,c) \) corresponds to determining whether there is a feasible lasso \( \lambda \) out of \( s \) with \( \text{MinCr}(\lambda) \leq c \). This may be done in polynomial time in the size of the weighted automaton using a slightly modified Bellman-Ford algorithm. Hence, this allows us to solve the existential lower-bound and lower-weak-upper-bound problems in polynomial time. The corresponding universal problems can be solved using very similar techniques that we do not detail here (see the full version). Hence we get the following result.

**Theorem 7.** For finite weighted automata, the existential and universal lower-bound and lower-weak-upper-bound problems are decidable in P. Also, \( \text{MinCr}(s_0) \) is computable in polynomial time.

### 4.2 One-Clock Weighted Timed Automata

Let \( A \) be a one-clock weighted timed automaton. Without loss of generality we shall assume that for any location, the value of the (single) clock \( x \) is bounded by some constant \( M \) (see [3]). Let \( 0 \leq a_1 < a_2 < \cdots < a_n < a_{n+1} = M \) where \( \{a_1, \ldots, a_n\} \) are the constants occurring in \( A \). Then the one-dimensional regions of \( A \) are all elementary open intervals \( (a_i, a_{i+1}) \) and singletons \( \{a_i\} \) (see [19]). In particular, two states \( (q,v) \) and \( (q,v') \) of \( A \) are time-abstract bisimilar whenever \( v \) and \( v' \) belong to the same region.

A corner-point region is a pair \( (\rho, e) \), where \( \rho \) is a region and \( e \) an end-point of \( \rho \). We say that \( (\rho', e') \) is the successor of \( (\rho, e) \) if either \( \rho' \) is the successor region of \( \rho \) and \( e = e' \) or \( \rho = \rho' = (a,b) \), \( e = a \), and \( e' = b \). The corner-point
Theorem 7. For the universal problem, note that Proposition 8 can be modified to a weighted timed automaton version.

Proof (sketch). Let us introduce the lower-bound and lower-weak-upper-bound infimum problem to be the problem as to whether for a given initial credit $c$, $L(s_0, c + \varepsilon)$ ($W_b(s_0, c + \varepsilon)$) holds for any $\varepsilon > 0$. Based on the above propositions we have:

Theorem 9. For one-clock weighted timed automata, the existential and universal lower-bound and lower-weak-upper-bound infimum problems are decidable in $P$. Also, MinCr($s_0$) may be computed in polynomial time.

Proof (sketch). For the existential problem, note that for a given one-clock weighted timed automaton $A$, the size of cpa($A$) is polynomial in the size of $A$, as the regions are constructed from the constants appearing in $A$. Then apply Theorem 7. For the universal problem, note that Proposition 8 can be modified to show that for any run in $A$ there exist a run in cpa($A$) which has always less remaining credit (within any region we simply choose to do the delay at the location with the smallest rate rather than the largest one). \hfill $\square$

4.3 Finite Weighted Games

Recall that a strategy $\sigma$ is said to be memoryless if $\sigma(\gamma)$ only depends on $\text{last}(\gamma)$ for any finite path $\gamma$. The proof of the following lemma can be found in the full version.

Lemma 10. Let $(S_1, S_2, s_0, T)$ be a finite weighted game. Let $b \in \mathbb{N} \cup \{\infty\}$ and $c \in \mathbb{N}$. If there exists a strategy $\sigma$ for Player 1 which ensures that $p_{<15}(\gamma') \geq 0$ for any finite prefix $\gamma'$ of any infinite outcome $\gamma$ of $\sigma$, then there is also a memoryless strategy with the same property. Symmetrically, if Player 2 has a strategy to ensure that for any outcome $\gamma$ of $\sigma$, there is a finite prefix $\gamma'$ of $\gamma$ such that $p_{<15}(\gamma') < 0$, then she has a memoryless strategy with the same property.
Proposition 11. For finite weighted games, the lower-bound and lower-weak-upper-bound problems are decidable in \( \text{NP} \cap \text{coNP} \).

Proof. The \( \text{NP} \) algorithm consists in nondeterministically guessing a memoryless strategy, pruning the transitions that are not selected by that strategy in \( G \), and checking whether for any finite prefix \( \gamma \) of any infinite execution, we have \( p_{\leq \gamma}(\gamma) \geq 0 \), which is polynomial (Theorem 7). If Player 1 has a winning strategy, then this algorithm will answer positively.

The \( \text{coNP} \) algorithm follows from the determinacy of the game: if Player 1 has no winning strategy, then Player 2 has one, which can be chosen memoryless. It can then be guessed and checked in nondeterministic polynomial time. \( \square \)

For the lower-bound problem, we can do better, and prove its equivalence with the mean-payoff problem.

Proposition 12. The mean-payoff game problem is log-space equivalent to the lower-bound problem, and is log-space reducible to the lower-weak-upper-bound problem.

Proof (sketch). Mean-payoff games are defined as follows [17]: given a weighted game \( G = (S_1, S_2, s_0, T) \) and an integer \( m \), is there a strategy for Player 1 s.t. for any infinite outcome \( s_0 \xrightarrow{p_0} s_1 \xrightarrow{p_1} \cdots \), we have \( \lim \inf_{n \to \infty} \sum_{j \leq n} p_j / n \geq m \). By shifting all weights by \( -m \), we can simplify the problem by assuming \( m = 0 \). An important feature of mean-payoff games is that they admit memoryless winning strategies in case they admit winning strategies [15].

Transforming a mean-payoff problem into a lower-bound (or lower-weak-upper-bound) problem is easy: the mean weight of an infinite path is negative iff its accumulated weight is \(-\infty\). Hence a winning strategy for Player 1 for the mean-payoff problem is also winning in the lower-bound (or lower-weak-upper-bound) problem, provided that the initial credit is sufficient (at least the opposite of the sum of all negative weights).

Conversely, given an instance \( G = (S_1, S_2, s_0, T) \) of the lower-bound problem (assuming the initial credit is zero, by possibly adding an extra initial transition setting the initial credit), we construct an equivalent instance of the mean-payoff problem as follows: each transition \( t = s \xrightarrow{p} s' \) is replaced by a new uncontrollable state \( s_t \) and three transitions \( s \xrightarrow{0} s_t, s_t \xrightarrow{0} s' \) and \( s_t \xrightarrow{0} s_0 \). In other words, at each transition of the original game, Player 2 is given the opportunity to go back to the initial state. If the accumulated weight goes below zero at some point, Player 2 will decide to go back to the initial state. Since strategies can be assumed to be memoryless, there will be a periodic outcome with negative mean weight. \( \square \)

The exact complexity of the mean-payoff problem is not known, but it is \( \text{P-hard} \) and in \( \text{UP} \cap \text{coUP} \) [17]. Hence we get the following theorem.

Theorem 13. For finite weighted games, the lower-bound problem is \( \text{P-hard} \) and in \( \text{UP} \cap \text{coUP} \) (thus in \( \text{NP} \cap \text{coNP} \)). The lower-weak-upper-bound problem is \( \text{P-hard} \) and in \( \text{NP} \cap \text{coNP} \).
5 Interval-Bound Problems

We shall now study the interval-bound problems for the finite and timed cases.

5.1 Finite Weighted Automata

The following theorem summarizes all our results on the various interval-bound problems for finite weighted automata and games.

**Theorem 14.** Under the interval-bound constraint,

1. the universal problem for finite weighted automata is decidable in P,
2. the existential problem for finite weighted automata is decidable in PSPACE and NP-hard, and
3. the problem for finite weighted games is EXPTIME-complete.

The complete proofs of all the results mentioned in this theorem can be found in the full version. We only mention some ideas, and then give the complete NP-hardness proof, as we think this can be rather instructive.

1. Using Bellman-Ford-like algorithms, we detect whether there is a negative (resp. positive) cycle or a shortest (resp. longest) path whose cost becomes negative (resp. goes above the maximal bound of the interval). This can be done in polynomial time.

2. From a finite weighted automaton, we can construct a finite graph whose set of states is a pair \((s, c)\) with \(s\) a state of the automaton, and \(c\) a cost value (in the interval \([0, b]\)), and in which \((s, c) \rightarrow (s', c')\) is a transition whenever \(s \xrightarrow{c-c'} s'\) is a transition of the weighted automaton. Guessing an infinite path in that graph gives a solution to the existential problem. This can be done in polynomial space (due to the encoding of \(b\) in binary). The NP-hardness proof will be detailed below.

3. Similarly as for the PSPACE upper bound, we construct the graph described above, and we add alternation: a state belonging to Player 1 (resp. 2) becomes existential (resp. universal). The interval-bound problem for finite weighted games reduces to an alternating graph accessibility problem (see [14]) which can be solved in P, hence the EXPTIME upper bound (the exponential blow-up is still due to the encoding of \(b\) in binary). EXPTIME-hardness is by reduction from the problem of countdown games which was very recently shown to be EXPTIME-complete [18].

**Proof (of the NP-hardness).** We reduce the NP-complete problem SUBSET-SUM (see e.g. [22]) into our existential interval-bound problem. An instance of SUBSET-SUM is a pair \((A, t)\) where \(A \subseteq \mathbb{N}\) is a finite set and \(t \in \mathbb{N}\). The question is whether there is a subset of \(A\) which adds exactly to \(t\). Assume a given instance of SUBSET-SUM \((A, t)\) where \(A = \{t_1, t_2, \ldots, t_n\}\). We construct a weighted automaton \((S, s_0, T)\) where \(S = \{s_0, s_1, \ldots, s_n\}\) and where \(T = \{s_i \xrightarrow{t_{i+1}} s_{i+1} | \)
0 ≤ i < n} ∪ \{s_i \rightarrow s_{i+1} | 0 ≤ i < n\} ∪ \{s_n \rightarrow s_0\}. The construction is depicted below.

Now consider the existential interval-bound problem with upper bound \(t\), and let the initial credit be 0. It is clear that there is an infinite path with the accumulated weight staying between 0 and \(t\) if and only if the SUBSET-SUM problem has a solution.

5.2 Timed Games

We prove in this section that the interval-bound problem for timed games is undecidable, even for games involving only one clock.

**Theorem 15.** The interval-bound problem is undecidable for one-clock weighted timed games.

**Proof (sketch).** This is achieved by simulating a two-counter machine. If \(c_1\) and \(c_2\) are the values of the counters, the accumulated weight along runs of the simulating weighted timed game will be \(E(c_1, c_2) = 5 - 1/(2^{c_1}3^{c_2})\), and we will work with the upper bound of 5. We start with accumulated weight 4 (encoding that the two counters are initialized to 0).

We will describe modules that increment respectively decrement a counter, and then informally describe how we can test that the value of a counter is zero. We assume that there is a global invariant \(x ≤ 1\) in all the modules. Moreover, for the purpose of this reduction, a configuration is a triple \((\ell, x, W)\) where \(\ell\) is a discrete location, \(x\) is the value of the clock, and \(W\) is the accumulated weight.

We first assume we have a module \(ok\) that is winning for Player 1 as soon as this module is entered with an accumulated weight in \([0, 5]\) (it is nothing more than a location, with weight-rate 0, and a controllable self-loop that checks whether \(x = 1\) and resets clock \(x\)).

We now consider the module \(Mod_n\) that is described in Fig. 3 (\(n\) is an integer that will be fixed later in the set \(\{2, 3, 12, 18\}\)). A strategy in \(Mod_n\) for Player 1 which has the property that any outcome either reaches a module \(ok\) or exits the module while always satisfying the lower- and upper-bound conditions will be said to be locally safe in that module.

Let \(e \in [0, 1]\) with \(0 ≤ ne ≤ 30\). Consider the following strategy \(\sigma_n\), depicted on Fig. 4:

- from location \((m, 0, 5 - e)\), it delays during \((5 - e)/6\) time units in \(m\), then leaves to \(m_1\) (thus to configuration \((m_1, (5 - e)/6, 0)\));
- from \(m_1\), it directly goes to \(m_2\);
Infinite Runs in Weighted Timed Automata with Energy Constraints

− from \((m_2,(5 - e)/6,0)\), it waits for 1/6 time units and then leaves to \(m_3\) (hence to configuration \((m_3,1 - e/6,5)\));
− from \((m_3,1 - e/6,5)\), it directly goes to \(m'\);
− in \(m'\), it delays until \(x = 1\), and fires the last transition to the next module.

In \(\text{Mod}_n\), from locations \(m_1\) (resp. \(m_3\)), Player 2 can decide to bring the game to the first (resp. second) vertical branch, and the only way to stay safe is to arrive in \(m_1\) (resp. \(m_3\)) with accumulated cost 0 (resp. 5) and to leave in 0-delay. This is precisely what does strategy \(\sigma_n\), which is then the unique locally safe strategy in module \(\text{Mod}_n\).

**Proposition 16.** Starting from configuration \((m,0,5 - e)\) in \(\text{Mod}_n\), with \(0 \leq ne \leq 30\), Player 1 has a unique locally safe strategy. Under that strategy, the only outcome that visits \(m'\) exits the module with accumulated weight \(5 - ne/6\).

In the sequel, we use the following modules: \(\text{Inc}(c_1) = \text{Mod}_3\), \(\text{Inc}(c_2) = \text{Mod}_2\), \(\text{Dec}(c_1) = \text{Mod}_{12}\), and \(\text{Dec}(c_2) = \text{Mod}_{18}\). When starting with accumulated weight \(5 - e\) in module \(\text{Inc}(c_1)\), the accumulated weight when leaving this module is \(5 - e/2\), hence \(\text{Inc}(c_1)\) increments the counter \(c_1\). Similarly, \(\text{Inc}(c_2)\) increments \(c_2\), and \(\text{Dec}(c_1)\) and \(\text{Dec}(c_2)\) decrement the respective counters. In the last two modules, no check is performed whether the counter to be decremented actually has a positive value.

We briefly describe how such a test can be done. First we construct a module that has two output branches, one where the two counters are equal to 0, and
one where one of the counters is positive. This is to Player 1 to decide in which
branch we go, but Player 2 can easily check that this is correct (checking that the
two counters are equal to zero reduces to checking that the accumulated weight is
4, whereas checking that one of the counters is positive reduces to checking that
the accumulated cost be in the open interval $(4, 5)$). Then, using that module,
we can for instance implement a test that counter $c_2$ is equal to zero as follows:
either both counters are equal to zero, or under the assumption that one of the
counters is positive, and by decrementing the first counter, we will eventually
reach a point where both counters are equal to zero. For lack of space, we do not
detail the construction of those modules, which use ideas quite similar to $\text{Mod}_n$.

Hence, fixing $\mathcal{M}$ to be a two-counter machine, we can construct a one-clock
weighted timed game $G_\mathcal{M}$ which is a positive instance of the interval-bound
problem if and only if $\mathcal{M}$ has an infinite computation. \qed

6 Conclusion

A summary of the results proved in this paper is provided in the following table.
The fields in gray remain open. Matching the complexity lower and upper bounds
for some of the problems is left open: the lower-bound problems for finite games
are strongly related to the well known open problem of complexity of mean-payoff
games; closing the gap between $\text{NP}$-hardness and containment in $\text{PSPACE}$
for the existential interval-bound problem seems intricate and it is a part of our future
work.

<table>
<thead>
<tr>
<th>games</th>
<th>existential problem</th>
<th>universal problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>finite</td>
<td>1-clock</td>
<td>finite</td>
</tr>
<tr>
<td>$L$</td>
<td>$\in \text{UP} \cap \text{coUP}$</td>
<td>$\in \text{P}$ (Th. 7)</td>
</tr>
<tr>
<td></td>
<td>P-h (Th. 13)</td>
<td></td>
</tr>
<tr>
<td>$L+W$</td>
<td>$\in \text{NP} \cap \text{coNP}$</td>
<td>$\in \text{P}$ (Th. 7)</td>
</tr>
<tr>
<td></td>
<td>P-h (Th. 13)</td>
<td></td>
</tr>
<tr>
<td>$L+U$</td>
<td>EXPTIME-c</td>
<td>Undec.</td>
</tr>
<tr>
<td></td>
<td>(Th. 14)</td>
<td>(Th. 15)</td>
</tr>
</tbody>
</table>

Note that the results related to Theorem 9 hold for the initial credits arbitrarily
close (for any given $\varepsilon > 0$) to the given ones.

References

1. R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability in weighted
   In Proc. 4th Int. Workshop Hybrid Systems: Computation and Control (HSCC’01),
3. 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 Int. Workshop on Hybrid Systems: Computation and Control (HSCC’01),