Energy and Mean-payoff Timed Games*

Romain Brenguier
Département d'informatique,
Université Libre de Bruxelles
(U.L.B), Belgium
brenguier@ulb.ac.be

Franck Cassez
NICTA,†
Sydney, Australia
franck.cassez@nicta.com.au

Jean-François Raskin
Département d'informatique,
Université Libre de Bruxelles
(U.L.B), Belgium
jraskin@ulb.ac.be

ABSTRACT

In this paper, we study energy and mean-payoff timed games. The decision problems that consist in determining the existence of winning strategies in those games are undecidable, and we thus provide semi-algorithms for solving these strategy synthesis problems. We then identify a large class of timed games for which our semi-algorithms terminate and are thus complete. We also study in detail the relation between mean-payoff and energy timed games. Finally, we provide a symbolic algorithm to solve energy timed games and demonstrate its use on small examples using HyTech.

1. INTRODUCTION

Timed automata [1], respectively timed games [33, 19], are fundamental models to verify, respectively to synthesize controllers for, timed systems which have to enforce hard real-time constraints. Those models were introduced in the nineties and the underlying theory has since then been successfully implemented in efficient analysis tools such as Kronos [15] and UppAal [32] for verification, and UppAal-Tiga [4] for synthesis. The latter has been used to solve industrial case studies, e.g. [28, 20].

Recently, there has been an important research effort to lift verification and synthesis techniques from the Boolean case to the quantitative case, see [26] and references therein. More specifically, lots of progress has been made recently on zero-sum two-player games played on weighted graphs, in which edges are decorated with *costs* or *rewards*, see for example [13, 22, 23, 18], with the objective of setting up a framework for the synthesis of optimal controllers (see also [34] for applications in linear control systems). Important examples of such games are *mean-payoff* and *energy* games [13, 18, 23]. In those games, two players move a token along the edges of a weighted graph whose vertices are partitioned into vertices that belong to player 1, and player 2 respectively. In each round of the game, the player that owns the vertex with the token chooses an outgoing edge and target vertex to move the token to. By playing in such a way, the two players form an infinite path through the graph. Player 1 wins the *mean-payoff objective* if the long-run average of the edge-weights along this path is non-negative, and he wins the *energy objective*, if there exists a bound $c \in \mathbb{Z}$ such that the running sum of weights of the traversed edges along the infinite path never goes below c (this can model for example that the system never runs out of energy). As the games we consider are zero-sum, player 2 wins when he can enforce the complementary objectives. In the finite state case, the mean-payoff and energy objectives are *inter-reducible*, and this fact was used recently to provide algorithmic improvements to solve mean-payoff games [18].

Extensions of timed automata with costs and rewards have also been studied. In [3, 5], timed automata are extended with continuous variables that are used as *observers*, and allow for modeling accumulation of costs or rewards along executions. The main motivation for studying those extensions is to offer an extra modeling power while avoiding severe intractability of richer models like hybrid automata. Indeed, it has been shown that the reachability problem for weighted/priced timed automata remains decidable [3, 5], and more precisely PSPACE-c [8], while the reachability problem is undecidable already for the class of stopwatch automata [21] (a simple class of hybrid automata). Also the existence of executions in a weighted automaton that ensure a bound on the mean-payoff can also be decided in PSPACE [10]. In this paper, we consider timed extensions of the important classes of mean-payoff and energy games.

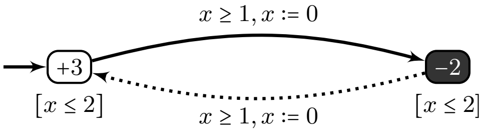

Figure 1: Turn-Based Energy Timed Game $\mathcal{A}$.

Example 1. Fig. 1 gives an example of an energy (turn-based) timed game. Eve (player 1) owns the left location and decides when to take the transition from left to right, while Adam (player 2) owns the right location and decides when to take the transition from right to left; x is a dense-time clock. Each transition resets the clock x, and when time elapses the energy level grows with derivative 3 in Eve's lo-

[†] NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.
[*] Work supported by ERC Starting Grant inVEST (279499).

HSCC'14, April 15–17, 2014, Berlin, Germany.
Copyright 2014 ACM 978-1-4503-2732-9/14/04 ...$15.00.
http://dx.doi.org/10.1145/2562059.2562116.

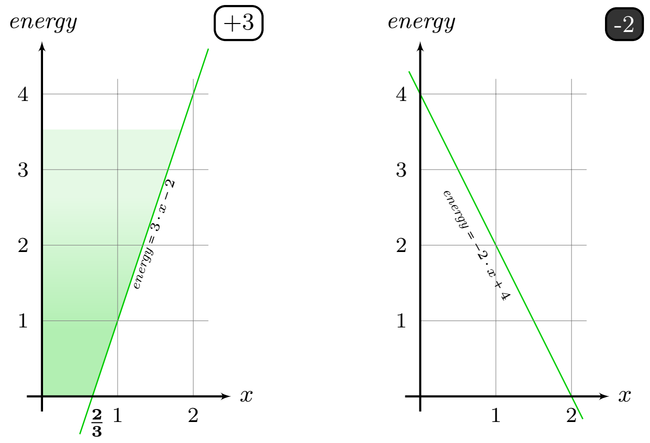

Figure 2: Winning Zones for Timed Game $\mathcal{A}$.

cation and decreases with derivative 2 in Adam's location. In the remaining of this paper, we use the following conventions: plain (resp. dashed) arrows are Eve's (resp. Adam's) transitions; a location *invariant* is enclosed in brackets (e.g. $[x \leq 2]$) and must be satisfied by the valuation of the clocks when the control is in that location; when an edge weight is non zero, we attach it to the edge. Fig. 2, left, depicts the initial energy levels that are sufficient for Eve in order to have a winning strategy against any strategy of Adam, i.e. to ensure an infinite execution in which the energy level never falls below 0. For instance, if the infinite game starts in Eve's location, then if the value of clock x is less than $\frac{2}{3}$, an initial energy level 0 is sufficient; for $x > \frac{2}{3}$, the initial energy level should be larger than or equal to $3 \cdot x - 2$. Similarly, the winning zone in Adam's location is depicted on the right hand side of Fig. 2. The main purpose of this article is to propose algorithmic methods to compute such information.

Unfortunately, for weighted extensions of timed games, even the cost bounded reachability problem is undecidable [17], and we show here that both mean-payoff and energy games are undecidable. This is unfortunate as such cost extensions of timed games are very natural and well-suited to model optimality problems in embedded control [20]. Nevertheless, we believe that the undecidability result should not be an end to the story and we study in this paper *semi-algorithms* (completeness and/or termination is not guaranteed) to solve those two synthesis problems. We also identify a large class of timed games where our semi-algorithms are complete. To the best of our knowledge, there are the first positive results for those objectives on timed automata. There are related works in the literature but they apply to orthogonal classes of games, or to other objectives. Indeed, in [9], it is shown that mean-payoff games are decidable for O-minimal hybrid automata, this class is different from the one identified here as timed automata are not O-minimal hybrid automata. In [30], the authors study the average time per transition problem for turn-based timed games; their results do not apply to mean-payoff, nor to energy objectives.

Contributions. Our contribution is threefold. First, we study the relation between mean-payoff and energy timed games. As we already mentioned, in the finite state case, the mean-payoff and energy objectives are inter-reducible [18]: given a weighted game G, Eve wins the mean-payoff objective if, and only if, she wins the energy objective. We show here that the relationship between the two types of games is more complex in the timed case. We identify conditions under which it is possible to transfer winning strategies for one objective into winning strategies for the other objective, and we show that those conditions are also necessary. Those results are formalized by Thm. 1 and Thm. 2.

Second, Thm. 3 establishes the undecidability of the decision problems associated with energy and mean-payoff timed games. This result is unfortunate but not surprising (it was already conjectured in [7], see page 89). This negative result motivates the main contribution of this paper: we propose two semi-algorithms for synthesizing winning strategies. We first consider a *cycle forming game* (in the spirit of [6]) on the region graph associated with the underlying weighted timed game: the two players move a token on the region graph and the game is stopped as soon as a cycle is formed. In Sect. 3.4, we partition the set of simple cycles of the region graph into those that are *good* for Eve, those that are *good* for Adam, and those that are neither good for Eve nor for Adam. If the formed cycle belongs to the first set then Eve is declared winner of the cycle forming game, if the cycle belongs to the second then Adam is the winner, otherwise it is a draw. Thm. 4 establishes that if Eve wins the cycle forming game then she has also a winning strategy in associated energy games, and Thm. 5 proves a similar result for Adam. Then, we identify a class of weighted timed games, that we call *robust*, for which this reduction to the cycle forming game on the region graph is complete: in this case the good cycles for Eve and the good cycles for Adam partition the set of simple cycles of the region graph. This class covers the class of timed games where costs appear on edges only. Thm. 10 establishes the decidability of the membership problem for the class of robust weighted timed games.

Finally, as the cycle forming game is defined on the region graph, it does not lead to a practical algorithmic solution. This is why we propose in addition a symbolic semi-algorithm to solve energy timed games. In Thm. 13, we show that our symbolic algorithm is also complete on the class of *robust* weighted timed games. In order to show the feasibility of our approach, we have implemented this algorithm as a script for HyTech [27] and ran it on small examples.

Our main theorems and their relation with the different classes of games we consider are depicted in Fig. 3 and 4.

Structure of the paper.

In Sect. 2, we define the mean-payoff and energy timed games. In Sect. 3, we develop semi-algorithms based on reductions to cycle games played on the region graph. In Sect. 4, we identify a class *robust games*, for which the reduction to cycle games is complete. In Sect. 5, we propose a symbolic semi-algorithm which is also complete for robust games.

Due to space constraints, most proofs have been omitted from this paper; they can be found in the full version [16].

2. PRELIMINARIES

In this section, we first recall the definition of concurrent games. Then we review a useful result from [29] that defines a canonical decomposition of infinite paths in a graph into *simple cycles*. Next, we introduce *weighted timed games*, the semantics of which is given in term of infinite concurrent games. Starting from the notion of *weight* (or cost/reward),

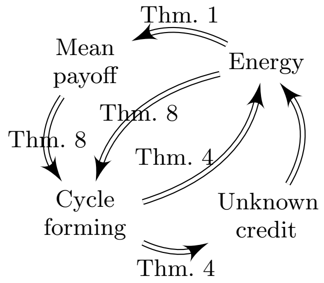

Figure 3: Winning strategies for `Eve`

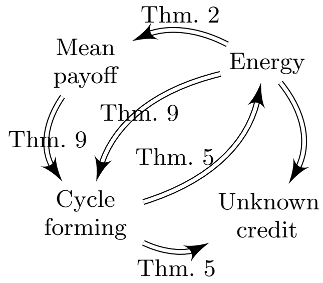

Figure 4: Winning strategies for `Adam`

we define *mean payoff* and *energy* objectives. We close the section with a study of the relationships that exist between mean-payoff and energy objectives in timed games.

We let $\mathbb{N}$ be the set of natural numbers, $\mathbb{Z}$ the set of integers, $\mathbb{R}$ the set of reals and $\mathbb{R}_+$ the set of non-negative reals.

2.1 Concurrent Games

Definition 1. ([2]) A *concurrent game* between two players $\texttt{Eve}$ and $\texttt{Adam}$ is a tuple $\mathcal{C} = \langle \texttt{St}, \iota, \texttt{Act}, \texttt{Mov}, \texttt{Tab}, \Omega \rangle$, where:

- $\texttt{St}$ is the set of *states*;
- $\iota \in \texttt{St}$ is the *initial* state;
- $\texttt{Act}$ is the set of *actions*;
- $\texttt{Mov} : \texttt{St} \times \{\texttt{Eve}, \texttt{Adam}\} \mapsto 2^{\texttt{Act}} \setminus \{\varnothing\}$ gives for a state and a player the set of *allowed actions*, we let $\texttt{Mov}(s) = \{(a, b) \mid a \in \texttt{Mov}(s, \texttt{Eve}), b \in \texttt{Mov}(s, \texttt{Adam})\}$;
- $\texttt{Tab} : \texttt{St} \times \texttt{Act} \times \texttt{Act} \mapsto \texttt{St}$ is the *transition function*;
- $\Omega \subseteq (\texttt{St} \cdot (\texttt{Act} \times \texttt{Act}))^{\omega}$ is the *objective* for $\texttt{Eve}$.

A concurrent game is *finite* if $\texttt{St}$ and $\texttt{Act}$ are finite. It is *turn-based* if in each state s, one of the players has only one allowed action.

A round of the game consists in $\texttt{Eve}$ and $\texttt{Adam}$ to choose independently and simultaneously some actions, say $a_\exists$ and $a_\forall$ respectively, such that $a_\exists \in \texttt{Mov}(s, \texttt{Eve})$ and $a_\forall \in \texttt{Mov}(s, \texttt{Adam})$. The pair $m = (a_\exists, a_\forall)$ is an allowed *move* i.e. $m \in \texttt{Mov}(s)$. By playing finitely (resp. infinitely) many rounds from state $s \in \texttt{St}$, the players build a finite (resp. infinite) *path*. Formally, a *path* is a finite or infinite sequence $s_0 \cdot m_0 \cdot s_1 \cdot m_1 \cdots s_k \cdot m_k \cdots$ of alternating states and moves, such that $\forall i \geq 0, m_i \in \texttt{Mov}(s_i)$ and $s_{i+1} = \texttt{Tab}(s_i, m_i)$. We also write a path as $s_0 \xrightarrow{m_0} s_1 \xrightarrow{m_1} \cdots$. The length, $|\rho|$, of an infinite path ρ is ∞ and the length of a finite path ρ with n moves (ending in s_n) is n and $\texttt{last}(\rho) = s_n$. We write $\rho_n, n \leq |\rho|$, for s_n, the $n+1$-th state in ρ, and $\texttt{first}(\rho) = \rho_0 = s_0$. Given a path $\rho = s_0 \xrightarrow{m_0} s_1 \xrightarrow{m_1} \cdots$, we write $\rho_{\leq n}, n \leq |\rho|$ for the prefix of ρ up to s_n that is the finite path $s_0 \xrightarrow{m_0} \cdots \xrightarrow{m_{n-1}} s_n$.

A *play* is an infinite path in $(\texttt{St} \cdot (\texttt{Act} \times \texttt{Act}))^{\omega}$, and a *history* is a finite path in $(\texttt{St} \cdot (\texttt{Act} \times \texttt{Act}))^{*} \cdot \texttt{St}$. The set of plays from s is $\texttt{Play}(\mathcal{C}, s)$ and $\texttt{Play}(\mathcal{C}) = \texttt{Play}(\mathcal{C}, \iota)$. A play is *winning* for $\texttt{Eve}$ if it belongs to the objective Ω.

Definition 2. (Strategies) A *strategy* for $\texttt{Eve}$ (resp. $\texttt{Adam}$) is a function which associates with a history h an $\texttt{Eve}$-action in $\texttt{Mov}(\texttt{last}(h), \texttt{Eve})$ (resp. an $\texttt{Adam}$-action). A pair of strategies $(\sigma_\exists, \sigma_\forall)$ forms a *strategy profile.* Given a strategy profile, its *outcome from state s*, written $\texttt{Out}_s(\sigma_\exists, \sigma_\forall)$, is the unique play ρ such that: $\rho_0 = s$ and $\forall n \geq 0, \rho_{n+1} = \texttt{Tab}(\rho_n, \sigma_\exists(\rho_{\leq n}), \sigma_\forall(\rho_{\leq n}))$. Given a strategy $\sigma_\exists$ of $\texttt{Eve}$, its *outcomes from state s*, written $\texttt{Out}_s(\sigma_\exists)$, are the set of plays ρ for which there is a strategy $\sigma_\forall$ of $\texttt{Adam}$, such that $\rho = \texttt{Out}_s(\sigma_\exists, \sigma_\forall)$. A strategy $\sigma_\exists$ of $\texttt{Eve}$ is *winning*, if for all strategies $\sigma_\forall$ of $\texttt{Adam}$, $\texttt{Out}_\iota(\sigma_\exists, \sigma_\forall) \in \Omega$; strategy $\sigma_\forall$ of $\texttt{Adam}$ is *winning*, if for all strategies $\sigma_\exists$ of $\texttt{Eve}$, $\texttt{Out}_\iota(\sigma_\exists, \sigma_\forall) \notin \Omega$.

2.2 Decomposition in Simple Cycles

In the sequel we will reduce energy and mean payoff games to games played on the region with *cycles* objectives. In this paragraph we recall the key results [29] related to the decomposition of a play into *simple cycles.* A history $h = s_0 \cdot m_0 \cdot s_1 \cdot m_1 \cdots s_n$ is a *cycle* if $s_0 = s_n, n \geq 1$. A *simple cycle* is a cycle such that for all i and j, $0 \leq i < j < n$, $s_i \neq s_j$. We write $\mathbb{C}(\mathcal{C})$ ($\mathbb{C}$ when $\mathcal{C}$ is clear from the context) for the set of simple cycles in the concurrent game $\mathcal{C}$.

Every history h of a finite game can be uniquely decomposed into a sequence of simple cycles, except for a finite part. The decomposition process maintains a *stack*, $\texttt{st}(h)$, of distinct states and moves. We write the stack content $s_1 \cdot m_1 \cdot s_2 \cdot \cdots \cdot m_{n-1} \cdot s_n$ where s_1 is at the bottom of the stack and s_n the top. We use the notation $s \in \texttt{st}(h)$ for $s \in \{s_1, s_2, \cdots, s_n\}$. The *decomposition*, $\texttt{dec}(h)$, is a set of simple cycles. We define $\texttt{dec}(h)$ and $\texttt{st}(h)$ inductively as follows:

- for the single state history s, $\texttt{dec}(s) = \varnothing$ and $\texttt{st}(s) = s$.
- let $h' = h \cdot m \cdot s$, m a move, $s \in \texttt{St}$, be a history.
  - If $s \in \texttt{st}(h)$, and $\texttt{st}(h) = \alpha \cdot s \cdot \beta$, then $\texttt{st}(h') = pop(\texttt{st}(h), |\beta|)$ and $\texttt{dec}(h') = \texttt{dec}(h) \cup \{s \cdot \beta \cdot m \cdot s\}$.
  - else $\texttt{dec}(h') = \texttt{dec}(h)$, $\texttt{st}(h') = push(\texttt{st}(h), m \cdot s)$.

Note that the stack always contains distinct elements, therefore only simple cycles are added to the decomposition. The elements in the stack from the bottom to the top, form a history $s_0 \cdot m_0 \cdot s_1 \cdot m_1 \cdots s_n$, where $n+1$ is the height of the stack. The decomposition of a play is the union of the decompositions of the finite prefixes of the play.

2.3 Weighted Timed Games

Let X be a finite set of variables called *clocks.* A *clock valuation* is a mapping $v : X \to \mathbb{R}_+$. We let $\mathbb{R}_+^X$ be the set of clock valuations over X. We let $\mathbf{0}_X$ be the *zero* valuation where all the clocks in X are set to 0 (we use $\mathbf{0}$ when X is clear from the context). Given $\delta \in \mathbb{R}_+$, $v + \delta$ denotes the valuation defined by $(v+\delta)(x) = v(x) + \delta$. We let $\mathfrak{C}(X)$ be the set of *convex constraints* on X which is the set of conjunctions of constraints of the form $x \bowtie c$ with $c \in \mathbb{N}$ and $\bowtie \in \{\leq, <, =, >, \geq\}$. Given a constraint $g \in \mathfrak{C}(X)$ and a valuation v, we write $v \models g$ if g is satisfied by v. Given $Y \subseteq X$ and a valuation v, $[Y \leftarrow 0]v$ is the valuation defined by $([Y \leftarrow 0]v)(x) = v(x)$ if $x \notin Y$ and $([Y \leftarrow 0]v)(x) = 0$ otherwise.

Definition 3. A *weighted timed game* [31] (WTG for short) is a tuple $\mathcal{T} = \langle \texttt{L}, \ell_\iota, X, T_\exists, T_\forall, \texttt{Inv}, \texttt{w} \rangle$, where:

- $\texttt{L}$ is the (finite) set of locations and ℓ_ι is the initial location;

- X is a finite set of clocks;
- $T_\exists, T_\forall \subseteq \mathtt{L} \times \mathfrak{C}(X) \times 2^X \times \mathtt{L}$ are the set of transitions belonging to $\mathtt{Eve}$ and $\mathtt{Adam}$ respectively, and we let $T = T_\exists \cup T_\forall$; An element of $T_\exists$ (resp. $T_\forall$) is an $\mathtt{Eve}$-transition (resp. $\mathtt{Adam}$-transition).
- $\mathtt{Inv}{:}\,\mathtt{L} \to \mathfrak{C}(X)$ defines the invariants of each location;
- $\mathtt{w}{:}\,\mathtt{L} \cup T \to \mathbb{Z}$ is a weight function assigning integer weights to locations and discrete transitions.

If, from each location, all the outgoing transitions belong to the same player, $\mathcal{T}$ is said *turn-based*.

Informally, a WTG is played as follows: a state of the game is a pair (ℓ, v) where ℓ is a location and v is a clock valuation such that $v \models \mathtt{Inv}(\ell)$. The game starts from the initial state $(\ell_\iota, \mathbf{0})$. From a state (ℓ, v), each player $p \in \{\mathtt{Eve}, \mathtt{Adam}\}$ chooses (independently) a *timed action* $a_p = (d_p, e_p)$ where $d_p \in \mathbb{R}_+$ and $e_p = (\ell, g, Y, \ell')$ is a p-transition. The intended meaning is that p wants to delay for d_p time units and then fire transition e_p. There are some restrictions on the possible choices of timed actions (d_p, e_p): 1) d_p must be compatible with the current state (ℓ, v) and location invariant, i.e. for all $0 \le d' \le d_p$, $v + d' \models \mathtt{Inv}(\ell)$; 2) e_p must be enabled after d_p time units, i.e. $v + d_p \models g$; 3) the target location's invariant must be satisfied when entering this location, i.e. $[Y \leftarrow 0](v + d_p) \models \mathtt{Inv}(\ell')$.

A timed action satisfying these restrictions is said *legal*. If from a given state, one player has no legal timed action to play (i.e. no discrete action is enabled in the future for this player), it plays a special action $\perp$. At each round of the game, players propose some actions, $a_\exists$ for $\mathtt{Eve}$, and $a_\forall$ for $\mathtt{Adam}$. Either $a_\exists$ is a legal action for $\mathtt{Eve}$; or there are no legal actions for $\mathtt{Eve}$ and $a_\exists = \perp$. Similarly for $a_\forall$. We assume that from any reachable state of the game, at least one player has a legal action, hence the pair $(\perp, \perp)$ is never proposed.

To determine the effect of a joint action, we select the player p that chooses the shortest delay d_p. In case both players choose the same delay, the convention is that $\mathtt{Adam}$ is selected (this is without loss of generality and other policies can be accommodated for). These informal game rules are formalized in the next section.

2.4 Semantics of Timed Games

Given a timed action $(d, e) \in \mathbb{R}_+ \times T$ with $e = (\ell, g, Y, \ell')$, a state (ℓ, v), the successor state in the WTG is (ℓ', v') if: 1) $\forall 0 \le \delta \le d, v + \delta \models \mathtt{Inv}(\ell)$; 2) and $v + \delta \models g$; 3) and $[Y \leftarrow 0](v + d) \models \mathtt{Inv}(\ell')$. We denote this transition $(\ell, v) \xrightarrow{(d,e)} (\ell', v')$ which accounts for a combined delay transition of d time units followed by the discrete step firing edge e. The *duration* of this transition is $\mathtt{d}((\ell, v) \xrightarrow{(d,e)} (\ell', v')) = d$. Its *reward* (or *weight*) is $\mathtt{w}((\ell, v) \xrightarrow{(d,e)} (\ell', v')) = d \cdot \mathtt{w}(\ell) + \mathtt{w}(e)$.

Given an objective $\Omega \subseteq \Big((\mathtt{L} \times \mathbb{R}_+^X) \cdot ((\mathbb{R}_+ \times T_\exists) \times (\mathbb{R}_+ \times T_\forall))\Big)^\omega$, the semantics of the WTG $\mathcal{T}$ is the (infinite) concurrent game $\mathcal{C}(\mathcal{T}, \Omega) = (\mathtt{St}, \iota, \mathtt{Act}, \mathtt{Mov}, \mathtt{Tab}, \Omega)$ defined by:

- the set of states is $\mathtt{St} = \mathtt{L} \times \mathbb{R}_+^X$ and the initial state is $\iota = (\ell_\iota, \mathbf{0})$;
- the set of actions is $\mathtt{Act} = \mathtt{Act}_\exists \cup \mathtt{Act}_\forall$, where $\mathtt{Act}_\exists = \mathbb{R}_+ \times T_\exists$ are the actions for $\mathtt{Eve}$ and $\mathtt{Act}_\forall = \mathbb{R}_+ \times T_\forall$ are the actions for $\mathtt{Adam}$;
- $\mathtt{Mov}(s, \mathtt{Eve}) \in \big(2^{\mathtt{Act}_\exists} \smallsetminus \{\varnothing\}\big) \cup \{\perp\}$ is the set of legal actions for $\mathtt{Eve}$ in s if there is at least one, or $\{\perp\}$ otherwise; and $\mathtt{Mov}(s, \mathtt{Adam})$ is defined similarly. Given $(a_\exists, a_\forall) \in \mathtt{Mov}(s, \mathtt{Eve}) \times \mathtt{Mov}(s, \mathtt{Adam})$, we define $\mathtt{Mov}(a_\exists, a_\forall)$ as follows:
  - if $a_\exists = \perp$ (resp. $a_\forall = \perp$) then $\mathtt{Adam}$ (resp. $\mathtt{Eve}$) is selected and $\mathtt{Mov}(a_\exists, a_\forall) = a_\forall$ (resp. $\mathtt{Mov}(a_\exists, a_\forall) = a_\exists$);
  - otherwise $a_\exists = (d_\exists, e_\exists)$ and $a_\forall = (d_\forall, e_\forall)$ and: 1. if $d_\exists < d_\forall$, $\mathtt{Mov}(a_\exists, a_\forall) = a_\exists$; 2. if $d_\forall \le d_\exists$ then $\mathtt{Mov}(a_\exists, a_\forall) = a_\forall$;
- Given two actions $a_\exists$ and $a_\forall$, $\mathtt{Tab}((\ell, v), a_\exists, a_\forall) = (\ell', v')$ if $(\ell, v) \xrightarrow{\mathtt{Mov}(a_\exists, a_\forall)} (\ell', v')$.

Let $h = s_0 \xrightarrow{a_1^\exists, a_1^\forall} s_1 \ \cdots\ s_{n-1} \xrightarrow{a_{n-1}^\exists, a_{n-1}^\forall} s_n \cdots$ be a finite or infinite path in $\mathcal{C}(\mathcal{T}, \Omega)$. The *duration* and *reward* of h are respectively:

$$\mathtt{d}(h) = \sum_{k=0}^{|h|-1} \mathtt{d}\left(s_k \xrightarrow{\mathtt{Mov}(a_k^\exists, a_k^\forall)} s_{k+1}\right)$$

$$\mathtt{w}(h) = \sum_{k=0}^{|h|-1} \mathtt{w}\left(s_k \xrightarrow{\mathtt{Mov}(a_k^\exists, a_k^\forall)} s_{k+1}\right)$$

A play ρ is said *non-Zeno* if $(d(\rho_{\le n}))_{n \in \mathbb{N}}$ is unbounded. A strategy σ is *immune from Zenoness* if all its outcomes are non-Zeno. A game is said to have *bounded transitions* if there is a bound D, such that for all states (ℓ, v), actions $a_\exists$, $a_\forall$: $\mathtt{d}\Big((\ell, v) \xrightarrow{\mathtt{Mov}(a_\exists, a_\forall)} \mathtt{Tab}((\ell, v), a_\exists, a_\forall)\Big) \le D$.

2.5 Mean-payoff and Energy Objectives

The *mean payoff* (per time unit) of a play is defined as the long-run average of reward per time unit. Formally, the mean payoff of a play ρ is:

$$\mathtt{MP}(\rho) = \liminf_{n \to \infty} \frac{\mathtt{w}(\rho_{\le n})}{\mathtt{d}(\rho_{\le n})}.$$

Definition 4. We consider the following types of games:

- The *mean payoff game* $\mathcal{T}_{\mathtt{MP}}$ associated with a WTG $\mathcal{T}$, is the game played on $\mathcal{T}$ where the objective (for $\mathtt{Eve}$) is to obtain a non-negative mean payoff: i.e. $\mathcal{T}_{\mathtt{MP}} = \mathcal{C}(\mathcal{T}, \Omega_{\mathtt{MP}})$ where $\Omega_{\mathtt{MP}} = \{\rho \in \mathtt{Play}(\mathcal{C}) \mid \mathtt{MP}(\rho) \ge 0\}$.
- Given an initial credit $c \ge 0$, the *c-energy game* $\mathcal{T}_{E(c)}$ associated to a WTG $\mathcal{T}$, is the game played on $\mathcal{T}$ where the objective $\Omega(c)$ is to maintain the reward of every prefix of every play above $-c$: i.e. $\mathcal{T}_{E(c)} = \mathcal{C}(\mathcal{T}, \Omega(c))$ where $\Omega(c) = \{\rho \in \mathtt{Play}(\mathcal{C}) \mid \forall n \in \mathbb{N}.\ c + \mathtt{w}(\rho_{\le n}) \ge 0\}$.
- The *energy game* associated with a WTG $\mathcal{T}$, is the game $\mathcal{C}(\mathcal{T}, \Omega_E)$ where the objective is $\Omega_E = \cup_{c \ge 0} \Omega(c)$.

Decision problems. For each type of games, we define the associated decision problem:

- *Mean-payoff*: Given a mean payoff game $\mathcal{T}_{\mathtt{MP}}$, is there a winning strategy for $\mathtt{Eve}$ in $\mathcal{T}_{\mathtt{MP}}$?
- *c-energy*: Given a c-energy game $\mathcal{T}_{E(c)}$, is there a winning strategy for $\mathtt{Eve}$ in $\mathcal{T}_{E(c)}$?

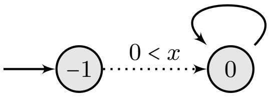

Figure 5: WTG $\mathcal{B}$.

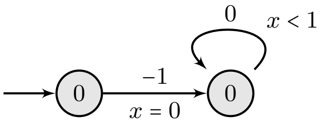

Figure 6: WTG $\mathcal{D}$.

- *Energy*: Given an energy game $\mathcal{T}_E$, is there a winning strategy for `Eve` in $\mathcal{T}_E$?

We also consider the following related problem:

- *Unknown initial credit*: Given a WTG $\mathcal{T}$, is there a credit c such that `Eve` has a winning strategy in $\mathcal{T}_{E(c)}$?

We also consider these problems from `Adam`'s point of view.

To conclude this section we study the relations between mean payoff and energy games and state that all decision problems we have defined are undecidable for WTG.

2.6 Relations Between Mean-payoff and Energy Objectives

Obviously, if for some c the c-energy game is won by `Eve`, then the energy game is also won. In the other direction, if `Adam` has a winning strategy for the energy game then it is also winning for any c-energy game. In the finite state case the problem of energy and unknown initial credit are equivalent [18]: if `Eve` has a winning strategy for the energy game she has a memoryless one, and there is a bound on the maximum energy consumed by the outcomes of that strategy. This is not the case in general for WTG as demonstrated by the WTG $\mathcal{B}$ of Fig. 5, for which `Eve` wins the mean-payoff game and the energy game, but no c-energy game.

While energy and mean-payoff objectives are inter-reducible in the finite state case [18], the relationships between the two classes of objectives, formalized in the next two theorems, is more subtle for weighted timed games.

THEOREM 1. *Let $\mathcal{T}$ be a WTG. If `Eve` has a winning strategy $\sigma_\exists$ in the energy game $\mathcal{T}_E$ and $\sigma_\exists$ is immune from Zenoness, then $\sigma_\exists$ is a winning strategy in the mean payoff game $\mathcal{T}_{\mathtt{MP}}$.*

Example 2. The following example shows that if we do not have immunity from Zenoness, the property no longer holds. In the game of Fig. 6, any play is winning for `Eve` in the c-energy game if $c > 1$. However, the total delay of a play is always smaller or equal to 1, hence the mean-payoff is smaller than -1, which means that `Eve` is losing.

We let $\mathcal{T}^{+\delta}$ be the game $\mathcal{T}$ in which we increase the weights of all locations by $\delta \in \mathbb{R}$. Formally $\mathcal{T}^{+\delta}$ is the WTG $\langle \mathtt{L}, \ell_\iota, X, T_\exists, T_\forall, \mathtt{Inv}, \mathtt{w}_{+\delta}\rangle$, where: 1) $\mathtt{w}_{+\delta}(\ell) = \mathtt{w}(\ell) + \delta$ if $\ell \in \mathtt{L}$; 2) $\mathtt{w}_{+\delta}(t) = \mathtt{w}(t)$ if $t \in T$.

THEOREM 2. *Let $\mathcal{T}$ be a WTG. If there exists $\delta > 0$, such that `Adam` has a winning strategy $\sigma_\forall$ in the energy game $\mathcal{T}_E^{+\delta}$ which is immune from Zenoness, then $\sigma_\forall$ is a winning strategy in the mean payoff game $\mathcal{T}_{\mathtt{MP}}$.*

Example 3. The following example shows that if we do not add this δ to the weight of locations, the property no longer holds. In the game of Fig. 7, for any initial credit c, `Adam` wins the c-energy game $\mathcal{T}_{E(c)}$. However `Eve` has a winning strategy in the mean-payoff game $\mathcal{T}_{\mathtt{MP}}$. She has to choose a delay which increases fast enough so that the weight of the play is small compared to its duration. For instance, if at the n-th step of the game, she chooses to delay for n^2 time units, the average weight of the play will be greater than $-\frac{1}{n}$. Hence it converges towards 0 and the mean-payoff is 0. Notice that if we add a small positive δ to the weight of each location, then following the same strategy, `Eve` also wins the c-energy game $\mathcal{T}_{E(c)}$ for c greater than $\frac{1}{\delta}$.

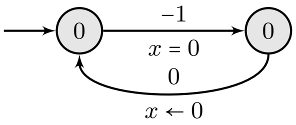

Figure 7: A WTG $\mathcal{T}$.

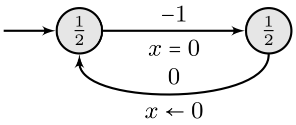

Figure 8: WTG $\mathcal{T}^{+\frac{1}{2}}$.

As already announced in the introduction, all the decisions problems that we have defined on weighted timed games are undecidable. The following theorem can be established (details are given in the full version of this paper) using variants of techniques used in [17, 14]:

THEOREM 3. *The mean-payoff, c-energy, energy and unknown initial credit problems are undecidable for WTG.*

3. A SEMI-PROCEDURE USING REGIONS

In this section, starting from the classical notion of regions [1], we define a finite concurrent game that exploits the relationship between timed paths and their projections in the region graph. We identify simple cycles in the region graph that are *good* for `Eve` (they roughly correspond to fragments of timed paths with *positive* reward), and others that are *good* for `Adam` (they roughly correspond to fragments of timed paths with *negative* reward). Thm. 4 tells us that if `Eve` can force to visit only her good cycles in the region graph then she has a winning strategy in the original energy timed game, and Thm. 5 is a symmetric result for `Adam`. To formalize those results, we need the notion of *quasi-path*: when we decompose a timed path according to simple cycles of the region graph, we introduce *jumps* inside regions because we remove a fragment of a timed path that starts inside a region and ends up in a possibly different state of the same region. Finally, we show how to solve the cycle forming game. This reduction is not complete: there are games which have winning strategies for `Eve` (or `Adam`) that our procedure will not find. However, we identify in Section 4 a class of games for which this reduction is complete.

3.1 Regions

We first recall the classical notion of *regions* [1]. If $k \in \mathbb{N}$, we write $\mathfrak{C}_k(X)$ for the set of constraints in $\mathfrak{C}(X)$ in which constants are integers within the interval $[\![0;k]\!]$. Let $\mathcal{T}$ be a WTG, and let $M = \max\{c \mid x \sim c \text{ is a constraint in } \mathcal{T}\}$. For $\delta \in \mathbb{R}_+$, we write $\lfloor\delta\rfloor$ the integral part of δ and $\mathrm{fr}(\delta)$ its fractional part. The equivalence relation $\equiv_{X,M}$ over $\mathbb{R}_+^X \times \mathbb{R}_+^X$ by $v \equiv_{X,M} v'$ if, and only if: 1) for all clocks $x \in X$, either $\lfloor v(x)\rfloor$ and $\lfloor v'(x)\rfloor$ are the same, or both $v(x)$ and $v(x)$ exceed M; 2) for all clocks $x, y \in X$ with $v(x) \le M$ and $v(y) \le M$, $\mathrm{fr}(v(x)) \le \mathrm{fr}(v(y))$ if, and only if, $\mathrm{fr}(v'(x)) \le \mathrm{fr}(v'(y))$; 3) for all clocks $x \in X$ with $v(x) \le M$, $\mathrm{fr}(v(x)) = 0$ if, and only if, $\mathrm{fr}(v'(x)) = 0$;

This equivalence relation naturally induces a partition $\mathcal{R}_{X,M}$ of $\mathbb{R}_+^X$. We write $[v]_{X,M}$ ($[v]$ when X and M are fixed) for the equivalence class of $v \in \mathbb{R}_+^X$. An equivalence class is called a *region.* It is well known that this partition has the following properties: 1) it is compatible with the constraints in $\mathfrak{C}_M(X)$, i.e. for every $r \in \mathcal{R}_{X,M}$, and constraint $g \in \mathfrak{C}_M(X)$ either all valuations in r satisfy the clock constraint g, or no valuation in r satisfies it; 2) it is compatible with time elapsing, i.e. if there is $v \in r$ and $t \in \mathbb{R}_+$ such that $v + t \in r'$, then for all $v' \in r$ there is t' such that $v' + t' \in r'$; 3) it is compatible with resets, i.e. if $Y \subseteq X$ then if $[Y \leftarrow 0]r \cap r' \neq \varnothing$ then $[Y \leftarrow 0]r \subseteq r'$.

A region r is said to be *time-elapsing*, if for any $v \in r$ there is $t > 0$ such that $v + t \in r$. We write $\mathrm{Succ}(r)$ the *successors* of r by time elapsing, it is defined by $r' \in \mathrm{Succ}(r)$ if there is $v \in r$ and $t \geq 0$ such that $v + t \in r'$.

3.2 Region Game

Given an objective $\Omega \subseteq ((\mathtt{L} \times \mathcal{R}_{X,M}) \cdot \mathtt{Act})^\omega$, the *region game* associated with a WTG $\mathcal{T} = \langle \mathtt{L}, \ell_0, X, T_\exists, T_\forall, \mathtt{Inv}, \mathtt{w} \rangle$ is the concurrent game $\mathcal{R}(\mathcal{T}, \Omega) = \langle \mathtt{St}, \iota, \mathtt{Act}, \mathtt{Mov}, \mathtt{Tab}, \Omega \rangle$ where:

- $\mathtt{St} = \mathtt{L} \times \mathcal{R}_{X,M}$ and $\iota = (\ell_\iota, \mathbf{0})$ is the initial state;
- $\mathtt{Act}$ is the set of actions. They are either $\bot$ or of the form (r, e, a) where $r \in \mathcal{R}_{X,M}$, $e \in T_\exists \cup T_\forall$ is a transition, and $a \in \{\mathtt{head}; \mathtt{tail}\}$; intuitively, an action is a target region (abstract delay) and a discrete transition. The extra component in $\{\mathtt{head}; \mathtt{tail}\}$ is needed to determine who plays first when the two players choose the same abstract delay (target region).
- Let $s = (\ell, r)$. Action (r', e, a) belongs to $\mathtt{Mov}(s, \mathtt{Eve})$ (resp. $\mathtt{Mov}(s, \mathtt{Adam})$), if: 1) $\exists (\ell, g, Y, \ell') \in T_\exists$ (resp. $T_\forall$); 2) $r' \in \mathrm{Succ}(r)$; 3) $r' \subseteq \mathtt{Inv}(\ell)$; 4) $r' \cap g \neq \varnothing$; 5) and $[Y \leftarrow 0]r' \subseteq \mathtt{Inv}(\ell')$. If there are no such action then only $\bot$ is allowed and this is the only situation in which $\bot$ is allowed.
- Let $s = (\ell, r)$ and $(r_\exists, e_\exists, a_\exists) \in \mathtt{Mov}(s, \mathtt{Eve})$, $(r_\forall, e_\forall, a_\forall) \in \mathtt{Mov}(s, \mathtt{Adam})$. $s' = \mathtt{Tab}(s, (r_\exists, e_\exists, a_\exists), (r_\forall, e_\forall, a_\forall))$ is defined as follows:
  - if $r_\exists \neq r_\forall$, one region is a strict (time abstract) predecessor of the other (as they are both successors of r). If $r_\exists$ is a strict predecessor of $r_\forall$, $\mathtt{Eve}$'s action $(r_\exists, e_\exists, a_\exists)$ is selected and otherwise $\mathtt{Adam}$'s action $(r_\forall, e_\forall, a_\forall)$ is selected.
  - otherwise $r_\exists = r_\forall$ and two cases arise: 1. either $r_\exists$ is not a time-elapsing region: in this case $\mathtt{Adam}$'s move is selected; 2. or $r_\exists$ is a time-elapsing region; which move is selected then depends on the extra components $a_\exists, a_\forall \in \{\mathtt{head}; \mathtt{tail}\}$: if $a_\exists = a_\forall$ then $\mathtt{Eve}$'s move is selected and otherwise $\mathtt{Adam}$'s move is selected.

  Once an action (r', e, a) with $e = (\ell, g, Y, \ell')$ is selected, the resulting state is $s' = (\ell', r'')$ with $r'' = [Y \leftarrow 0]r'$.

Remark 1. *In case $\mathcal{T}$ is turn-based, then in each state of $\mathcal{R}(\mathcal{T}, \Omega)$ only one player has a choice. The region game can then be seen as a (classical) turn-based finite game.*

Example 4. We want to reduce the problem of finding winning strategies in a WTG to an equivalent problem in the region game. To illustrate the need for the extra component in the actions (i.e. $a \in \{\mathtt{head}; \mathtt{tail}\}$) consider the example of Fig. 9. In the WTG (left), $\mathtt{Eve}$ has no winning strategy to win the mean payoff game: $\mathtt{Adam}$ can always choose a delay shorter than her from ℓ_0 to enforce location ℓ_2. For the same reason $\mathtt{Adam}$ has also no winning strategy. In the region game (right), we abstract away from the actual delays $\mathtt{Eve}$ and $\mathtt{Adam}$ can propose: they have only one possible choice which is to propose to delay up-to region $0 < x < 1$. To reproduce the possibility that either $\mathtt{Eve}$ or $\mathtt{Adam}$ are able to propose the smallest delay, we use the choices of both players in $\{\mathtt{head}, \mathtt{tail}\}$.

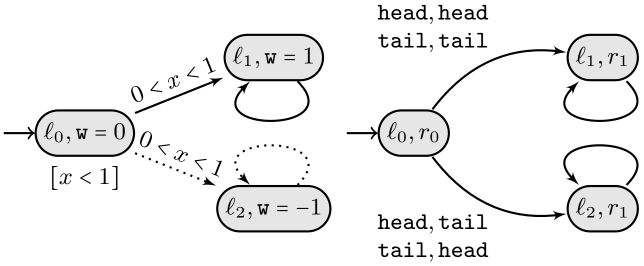

Figure 9: A WTG $\mathcal{T}$ and its associated region game.

3.3 Quasi Paths

A *quasi path* in a WTG is a sequence of states and transitions $\rho = (\ell_0, v_0)\tau_0(\ell_1, v_1)\tau_1 \cdots \tau_{n-1}(\ell_n, v_n)$ such that for all $0 \leq i \leq n-1$ either: 1) τ_i is a move $(a_\exists, a_\forall)$ and $(\ell_i, v_i) \xrightarrow{\mathtt{Mov}(a_\exists, a_\forall)} (\ell_{i+1}, v_{i+1})$; 2) or $\ell_i = \ell_{i+1}$ and $[v_i] = [v_{i+1}]$. In that case τ_i is called a *jump.* We will denote jumps by $\curvearrowright$. A *quasi cycle* is a quasi path such that $(\ell_n, [v_n]) = (\ell_0, [v_0])$.

In order to extend the reward to quasi paths, we need a *weight function* $\delta : \mathtt{L} \times \mathcal{R}_{X,M} \mapsto \mathbb{R}$ which attributes a weight to jumps according to the region in which they happen. We define $\mathtt{w}_\delta$ the reward for each transition τ_i, depending on its type: 1) if τ_i is a move $(a_\exists, a_\forall)$ and $(\ell_i, v_i) \xrightarrow{\mathtt{Mov}(a_\exists, a_\forall)} (\ell_{i+1}, v_{i+1})$, then $\mathtt{w}_\delta((\ell_i, v_i)\tau_i(\ell_{i+1}, v_{i+1})) = d \cdot \mathtt{w}(\ell_i) + \mathtt{w}(e)$ where $(d, e) = \mathtt{Mov}(a_\exists, a_\forall)$; 2) otherwise $\tau_i = \curvearrowright$, then $\mathtt{w}_\delta((\ell_i, v_i) \curvearrowright (\ell_{i+1}, v_{i+1})) = \delta([\ell_i, v_i])$. The reward for the quasi path ρ is then $\mathtt{w}_\delta(\rho) = \sum_{i<|\rho|} \mathtt{w}_\delta((\ell_i, v_i)\tau_i(\ell_{i+1}, v_{i+1}))$.

We define a projection from quasi paths to paths in the region game by forgetting jumps and projecting each state to its associated region. Formally, the projection π is defined inductively: 1. $\pi((\ell, v)) = (\ell, [v])$; 2. $\pi(h \curvearrowright (\ell, v)) = \pi(h)$; 3. $\pi\left(h \cdot (\ell, v) \xrightarrow{a_\exists, a_\forall} (\ell, v')\right) = \pi(h \cdot (\ell, v)) \xrightarrow{b_\exists, b_\forall} (\ell', [v'])$ where $a_p = (d_p, e_p)$ for $p \in \{\mathtt{Eve}, \mathtt{Adam}\}$, $b_\exists = ([v + d_\exists], e_\exists, \mathtt{head})$, and $b_\forall = ([v + d_\forall], e_\forall, \mathtt{head})$ if $d_\exists < d_\forall$ and $b_\forall = ([v + d_\forall], e_\forall, \mathtt{tail})$ otherwise. It is naturally extended from histories to plays: ρ' is the projection of ρ if for all finite prefixes h of ρ, $\pi(h)$ is a prefix of ρ'. If h is a path in the region game, the path ρ is said *compatible* with h if $\pi(\rho) = h$, and we write $\gamma(h)$ for the set of path compatible with h.

3.4 Reduction to the Region Game

Given a weight function $\delta : \mathtt{L} \times \mathcal{R}_{X,M} \mapsto \mathbb{R}$, we will write $\mathbb{C}_\delta^+$ for the set of simple cycles in the region game that only correspond to quasi cycles rewarding more than δ if δ is positive and more than 0 otherwise. Formally[1]:

$$\mathbb{C}^{+}_{\delta} = \{c \in \mathbb{C} \mid \forall \rho \in \gamma(c).\ \mathtt{w}_{\delta}(\rho) \geq \max\{\delta(\mathtt{first}(c)), 0\}\}.$$

Given a real number $\varepsilon > 0$, we write $\mathbb{C}^{-\varepsilon}_{\delta}$ for the simple cycles of the region game that correspond to quasi cycles with weight lower than δ and lower than $-\varepsilon$. Formally[2]:

$$\mathbb{C}^{-\varepsilon}_{\delta} = \{c \in \mathbb{C} \mid \forall \rho \in \gamma(c).\ \mathtt{w}_{\delta}(\rho) \leq \min\{\delta(\mathtt{first}(c)), -\varepsilon\}\}.$$

The winning condition of the region game will be given by *cycle objectives*. The intuition behind the definition of these objectives, is that if Eve can force the play to see only cycles with positive reward (i.e. in $\mathbb{C}^{+}_{\delta}$), the accumulated weight will be positive, except for a finite part. Which means she is winning the c-energy game, if c is big enough to cover the loss of energy in this finite part.

In the region game, we will consider the cases where the objective of Eve is given by $\Omega^{+}_{\delta} = \{\rho \mid \mathtt{dec}(\rho) \subseteq \mathbb{C}^{+}_{\delta}\}$. That is, she wins for plays whose decomposition in simple cycles only contains positive cycles.

Given a WTG $\mathcal{T}$, let $W_T = \min_{t \in T}\{\mathtt{w}(t)\} \cup \{0\}$ and $W_{\mathtt{L}} = \min_{\ell \in \mathtt{L}}\{\mathtt{w}(\ell)\} \cup \{0\}$.

Theorem 4. *Let $\mathcal{T}$ be a WTG, if Eve has a winning strategy in $\mathcal{R}(\mathcal{T}, \Omega^{+}_{\delta})$ then: 1) she has a winning strategy $\tau_\exists$ in the energy game $\mathcal{T}_E$; 2) if $\mathcal{T}$ has bounded transitions, $\tau_\exists$ is a winning strategy in the energy game $\mathcal{T}_{E(c)}$ for the initial credit $c = |L \times \mathcal{R}_{X,M}| \cdot (W_L \cdot D + W_T)$; 3) if $\tau_\exists$ is immune from Zenoness, then it is winning in the mean payoff game $\mathcal{T}_{\mathtt{MP}}$.*

Remark 2. *We made the hypothesis that there exists a bound on the duration of transitions in order to get the result for the unknown initial credit. Consider the example of Fig. 5. In this game, Eve is winning in the region game for Ω^{+}_{δ}, and therefore by Thm. 4 she also wins in the energy game $\mathcal{T}_E$, by Thm. 1 she also wins the mean-payoff game $\mathcal{T}_{\mathtt{MP}}$ if we consider a strategy that is immune from Zenoness. However, the transition going out of ℓ_0 can be taken by Adam at any moment, its duration is not bounded. Indeed, whatever the initial credit is, Adam can force a play which costs more than this credit, by delaying the transition for long enough. Therefore Eve has no winning strategy for any fixed initial credit and the answer to the unknown initial credit problem is negative.*

We now consider the objective for Eve: $\Omega^{-\varepsilon}_{\delta} = \{\rho \mid \mathtt{dec}(\rho) \nsubseteq \mathbb{C}^{-\varepsilon}_{\delta}\}$. That is, she wins if the decomposition in simple cycle contains at least on simple cycle that is not below $-\varepsilon$.

Theorem 5. *Let $\mathcal{T}$ be a WTG, if Adam has a winning strategy in $\mathcal{R}(\mathcal{T}, \Omega^{-\varepsilon}_{\delta})$ then 1) he has a winning strategy in the energy game $\mathcal{T}_E$; 2) if $\mathcal{T}$ has bounded transitions, then Adam has a winning strategy in the mean payoff game $\mathcal{T}_{\mathtt{MP}}$.*

3.5 Solving the Region Game

We now show how to solve a finite game with objective of the form $\Omega = \{\rho \mid \forall c \in \mathtt{dec}(\rho).c \in \mathbb{C}^W\}$ or $\Omega = \{\rho \mid \exists c \in \mathtt{dec}(\rho).c \in \mathbb{C}^W\}$. We can then apply this technique to solve the region game. To do so we unravel the game, and stop as soon as a cycle is formed. The play is then winning if the cycle formed belong to $\mathbb{C}^W$. This technique is adapted from [6].

Definition 5. Let $\mathcal{G} = \langle \mathtt{St}, \iota, \mathtt{Act}, \mathtt{Mov}, \mathtt{Tab}, \Omega \rangle$ be a concurrent game with $\Omega = \{\rho \mid \forall c \in \mathtt{dec}(\rho).c \in \mathbb{C}^W\}$ or $\Omega = \{\rho \mid \exists c \in \mathtt{dec}(\rho).c \in \mathbb{C}^W\}$. The unraveling of $\mathcal{G}$, written $\mathcal{U}(\mathcal{G})$, is the tuple $\langle \mathtt{St}', \iota, \mathtt{Act}, \mathtt{Mov}', \mathtt{Tab}', \Omega' \rangle$:

- the set of states is $\mathtt{St}' = \{h \in (\mathtt{St} \cdot (\mathtt{Act} \times \mathtt{Act}))^* \cdot \mathtt{St} \mid \forall i, j \neq i.\ h_i \neq h_j\} \cup \{\uparrow, \downarrow\}$, the set of histories of the original game where all states appear at most once; with the addition of a winning state $\uparrow$ and a loosing state $\downarrow$ for Eve;
- $\mathtt{Mov}'(h, p) = \mathtt{Mov}(\mathtt{last}(h), p)$;
- for an history h and a move $(a_\exists, a_\forall)$, let $s = \mathtt{Tab}(\mathtt{last}(h), a_\exists, a_\forall)$: 1) if s does not appear in h then $\mathtt{Tab}'(h, a_\exists, a_\forall) = h \xrightarrow{a_\exists, a_\forall} s$; 2) otherwise, let i be such that $h_i = s$, and $c = h_{\geq i} \xrightarrow{a_\exists, a_\forall} s$ (notice that such a i is unique): if c belongs to $\mathbb{C}^W$ then $\mathtt{Tab}'(h, a_\exists, a_\forall) = \uparrow$ and otherwise $\mathtt{Tab}'(h, a_\exists, a_\forall) = \downarrow$. Then, from $\uparrow$ and $\downarrow$, there are only self loops, thus $\mathtt{Tab}'(x, a_\exists, a_\forall) = x$ for $x \in \{\uparrow, \downarrow\}$.
- the objective is to reach $\uparrow$, i.e. $\Omega' = (\mathtt{St}' \cdot (\mathtt{Act} \times \mathtt{Act}))^* \cdot (\uparrow \cdot (\mathtt{Act} \times \mathtt{Act}))^\omega$.

Theorem 6. *Let $\mathcal{G}$ be a concurrent game and $\mathcal{U}(\mathcal{G})$ its unraveling. Then Eve has a winning strategy in the unraveled game $\mathcal{U}(\mathcal{G})$ if, and only if, she has a winning strategy in $\mathcal{G}$.*

Theorem 7. *Given a finite concurrent game $\mathcal{G}$ with objective $\Omega = \{\rho \mid \forall c \in \mathtt{dec}(\rho).c \in \mathbb{C}^W\}$ or $\Omega = \{\rho \mid \exists c \in \mathtt{dec}(\rho).c \in \mathbb{C}^W\}$ where $\mathbb{C}^W$ is given by an automaton, deciding if Eve has a winning strategy is* PSPACE*-complete.*

4. ROBUST GAMES

The reduction to the cycle forming game in the region graph is complete when there exists a weight function δ, that partitions the set of simple cycles of the region into good ones for Eve and good ones for Adam.

Definition 6. (Robust game) A WTG is said *δ-robust* if $\mathbb{C} = \mathbb{C}^{+}_{\delta} \cup \mathbb{C}^{-\varepsilon}_{\delta}$ for some ε. We simply call a WTG *robust* when there exists $\delta : \mathcal{R}_{X,M} \mapsto \mathbb{R}$ such that it is δ-robust.

Remark 3. *Note that a WTG $\mathcal{G}$ where all the costs are discrete (i.e. $\forall \ell \in \mathtt{L}.\ w(\ell) = 0$) is robust for $\delta = 0$ and $\varepsilon < 1$. The results of this section implies decidability of the energy problem and mean payoff problem for this class.*

Remark 4. *If $\mathcal{T}$ is robust, then the leafs of $\mathcal{U}(\mathcal{R}(\mathcal{T}))$ are partitioned between winning for Eve and winning for Adam. If, in addition, this game is turn-based, then it is determined. By Thm. 4 and Thm. 9, we can conclude that if $\mathcal{T}$ is robust and turn-based then the energy game $\mathcal{T}_E$ is determined.*

Now, we establish that in a robust WTG, we can decide if Eve can win the energy game. This is a consequence of the following theorem which complements Thm. 4. A symmetric result also holds for Adam.

[1] Note, that this definition is inductive: as a jump in the region (ℓ, r) gives a reward of $\delta(\ell, r)$, we make sure that a (quasi)-cycle on that region always provides a reward larger than or equal to this value.

[2] Note, that the definition for the *good* cycles of Adam is symmetric but slightly stronger as we require that the weight of (quasi)-cycles to be ε-bounded away from zero.

THEOREM 8. *Let $\mathcal{T}$ be a δ-robust WTG: 1) if* `Eve` *has a winning strategy in the energy game $\mathcal{T}_E$ then she has a winning strategy in $\mathcal{U}(\mathcal{R}(\mathcal{T},\Omega_\delta^+))$; 2) if $\mathcal{T}$ has bounded transitions and* `Eve` *has a winning strategy in the mean payoff game $\mathcal{T}_{\mathtt{MP}}$ then she has a winning strategy in $\mathcal{U}(\mathcal{R}(\mathcal{T},\Omega_\delta^+))$.*

THEOREM 9. *Let $\mathcal{T}$ be a δ-robust WTG: 1) if* `Adam` *has a winning strategy in the energy game $\mathcal{T}_E$, then he has a winning strategy in $\mathcal{U}(\mathcal{R}(\mathcal{T},\Omega_\delta^{-\varepsilon}))$; 2) if $\mathcal{T}$ has bounded transitions and* `Adam` *has a winning strategy in the mean payoff game $\mathcal{T}_{\mathtt{MP}}$, then he has a winning strategy in $\mathcal{U}(\mathcal{R}(\mathcal{T},\Omega_\delta^{-\varepsilon}))$.*

Given a weighted timed game, it is decidable whether this game is robust or not. Additionally, we present several complexity results about this problem in the full version of this paper.

THEOREM 10. *The membership problem for the class of robust game is decidable.*

Finally, we can characterize the complexity of deciding the energy problem for robust weighted timed games:

THEOREM 11. *The energy problem for robust games is in* EXPSPACE *and is* EXP*-hard.*

PROOF SKETCH. The algorithm proceeds by constructing the region game and then solving it using the algorithm of Thm. 7. This is correct because of Thm. 4, 6 and 8. □

5. FIXPOINT ALGORITHM

While the reduction to the cycle forming game in the region graph is elegant and allows us to identify a large and natural class of weighted timed games with decidable properties, this reduction does not lead directly to a practical semi-algorithm. In this section, we design a valuation iteration algorithm that can be implemented symbolically using polyhedra, which can be executed on any weighted timed game, and find winning strategies for `Eve` when it terminates successfully. We also show that termination is guaranteed on robust weighted timed games.

5.1 Value Iteration Algorithm

Our value iteration algorithm is an adaptation of the solution for the finite state case described in [18] to the setting of WTG. It computes successive approximations of the minimal energy level that `Eve` needs to win the energy game.

Essentially, the semi-algorithm is based on the iteration of an operator that computes successive approximations of the energy level/credit that is necessary for `Eve` to maintain the energy level positive for k rounds in the energy timed game, where k increases along with the iterations. Most importantly, our algorithm is parameterized by a value $c \in \mathbb{N}$, that represents a maximal energy level that we want to track: if the energy level necessary to stay alive from a given state (l, v) for k rounds is larger than c then (ℓ, v) is considered as loosing. This is a sound approximation when looking for winning strategies. This parameter is important to enforce termination of the analysis. If a fixed point is reached, then it contains enough information to identify winning states (those that are not mapped to $+\infty$ by the operator) and construct winning strategies. If the analysis is negative (no winning strategy found) then this value can be increased. Furthermore, we show that for robust weighted timed games, there is a finite value c which is computable and sufficient to detect winning strategies for `Eve`.

In this section, we assume the WTG is fixed and transitions, weight functions, etc ...refer to this game. Given $c \in \mathbb{N}$, we write $a \ominus_c b$ for $\max(0, a-b)$ if $a - b \le c$ and $+\infty$ otherwise. In the sequel we consider mappings in $\mathcal{S} = [\mathtt{St} \mapsto \mathbb{R}_+ \cup \{+\infty\}]$ that associate with each state an element in $\mathbb{R}_+ \cup \{+\infty\}$.

Given $c \in \mathbb{N}$, the operator $\mathtt{lift}_c : \mathcal{S} \mapsto \mathcal{S}$ is defined by:

$$\forall s \in \mathtt{St}, \mathtt{lift}_c(f)(s) = \inf_{a_\exists} \sup_{a_\forall} \left\{ f(s') \ominus_c \mathtt{w}(s \xrightarrow{a_\exists, a_\forall} s') \right\}$$

We let $f_0^c : \mathtt{St} \mapsto \mathbb{R}_+ \cup \{+\infty\}$ be the mapping defined by $\forall s \in \mathtt{St}, f_0^c(s) = 0$. We then inductively define f_{k+1}^c, for $k \ge 0$ to be $\mathtt{lift}_c(f_k^c)$. Thus, $f_n^c(s)$ represents the initial energy level that is needed by `Eve` to keep the energy level positive for n steps from s. If more than c is needed then $f_n^c(s)$ is set to be $+\infty$ (c being the maximal energy level that we want to track). This is formalized in Lem. 1.

LEMMA 1. *For all state s, index n, credit c, $\varepsilon > 0$:*

$$f_n(s) \ge -\sup_{\sigma_\exists} \inf_{\sigma_\forall} \left\{ w_{/c}(\mathit{Out}_s(\sigma_\exists, \sigma_\forall)_{\le n}) \right\}.$$

where for a history $h = h_0 \xrightarrow{m_0} h_1 \cdots \xrightarrow{m_n} h_n$, $w_{/c}(h) = -\infty$ if $\exists i.\ w(h_i \xrightarrow{m_i} h_{i+1}) < -c - \varepsilon$ and $w(h)$ otherwise.

THEOREM 12 (CORRECTNESS). *If there exists $n \ge 0$ such that $f_{n+1} = f_n$, and $f_n^c(\iota) \ne +\infty$, then for any $\varepsilon > 0$,* `Eve` *has a winning strategy for the c'-energy game $\mathcal{T}_{E(c')}$ with initial credit $c' = f_n^c(\iota) + \varepsilon$.*

In the case of robust game, we show that we can stop the algorithm after a finite number of iterations.

THEOREM 13 (TERMINATION). *Let $\mathcal{T}$ be a WTG and c a fixed credit. If $\mathcal{T}$ is δ, ε-robust and $f_{n_0}^c(\ell_\iota, \mathbf{0}) \ne +\infty$ for $n_0 \ge \left(\frac{c \cdot (|L| \cdot |\mathcal{R}_{X,M}| + 1)}{\varepsilon} + 1\right) \cdot |L| \cdot |\mathcal{R}_{X,M}|$ then* `Eve` *has a winning strategy in the energy game.*

If moreover the game has bounded transition, the algorithm is complete.

THEOREM 14 (COMPLETENESS FOR ROBUST GAMES). *Let $\mathcal{T}$ be a robust WTG, which has bounded transitions, and $c = |L \times \mathcal{R}_{X,M}| \cdot (W_L \times D + W_T) + 1$.* `Eve` *has a winning strategy in the energy game $\mathcal{T}_{E(c)}$ if, and only if, $f_{n_0}^c \ne +\infty$.*

5.2 Symbolic Algorithm

We have implemented the previous value iteration algorithm in HYTECH [27]. The implementation is based on the symbolic *controllable timed predecessors* operator defined in [11] and first implemented in HYTECH for cost optimal reachability games [12]. The choice of HYTECH compared to state-of-the-art hybrid systems' analyzers like PHAVer [24] or SpaceEx [25] is motivated by the fact that HYTECH has a built-in script language in which we can define the symbolic *controllable* predecessors operator easily. The symbolic algorithm/program in HYTECH for the example of Fig. 1 is given in the full version of this paper. The result of the computation for the value iteration algorithm with $c = 4$ is depicted on Fig. 2 and show the winning zones for `Eve` in locations `Eve` and `Adam`.

The value iteration algorithm is implemented as the iterative computation of the fixpoint of a safety *hybrid* game. The hybrid game has a special variable E, the energy variable which is the only variable that is not a clock. Each location ℓ of the original WTG has a counterpart location $\ell_{\mathcal{H}}$ in the hybrid game. If $w(\ell) = k \in \mathbb{Z}$ then the derivative of E in $\ell_{\mathcal{H}}$ is given by $\frac{dE}{dt} = k$; each discrete transition (ℓ, g, z, ℓ') of the WTG also has a counterpart transition $(\ell, g, z \wedge E := E + k, \ell')$ if $w(\ell, g, z, \ell') = k$.

A state of the hybrid game is thus defined by $((\ell, v), E)$ where (ℓ, v) is a state of the original WTG. The existence of a winning strategy for the c-energy game $\mathcal{T}$ is reduced to the existence of a winning strategy in the associated hybrid game for the safety objective $E \geq 0$ (in each location.) Let $\mathit{Safe} = \{((\ell, v), E) \mid 0 \leq E \leq c\}$, where the upper bound c is the one used in the `lift` function from subsection 5.1. We define the winning states of the safety hybrid game as the greatest fixpoint of:

$$X = \mathit{Safe} \cap \mathtt{Pred}_t(\mathtt{Up}(\mathtt{cPred}(X)), \mathtt{uPred}(\overline{X}))$$

where `cPred` (controllable predecessor), `uPred` (uncontrollable predecessor), $\mathtt{Pred}_t$ (temporal predecessor) are defined as in [11, 12] and

$$\mathtt{Up}(Y) = \{((\ell, v), e') \mid \exists ((\ell, v), e) \in Y \wedge e' \geq e\}.$$

The `Up` operator captures in our symbolic implementation the role of the bound c in the $\mathtt{lift}_c$ operator: indeed, while the set X_i contains only triples $((\ell, v), e)$ where $e \leq c$, it is clear that we must include in X_{i+1} triples $((\ell', v'), e')$ from which `Eve` can force in one round the upward closure (for the energy level) of safe states in i steps. This is because if `Eve` can win from (ℓ, v) with a given energy level then she can win from that state with any greater energy level.

Example 5. In Fig. 10, plain (resp. dashed) arrows are controllable (resp. uncontrollable) edges. In location ℓ_0, no task is scheduled and the (battery) energy is recharging at rate +3. There is a background task B to be run at least every 2 t.u. if the other task has not arrived (and actually running in location ℓ_1) and a sporadic task S (interrupt) that can happen any time. The task B can be scheduled from location ℓ_0 (this is controllable) and we can stop to run it after at least 1 t.u. (measured by clock x.) The background task B has less priority than S and if S happens it is scheduled and B preempted. If we schedule the background task B, we are rewarded by +2 energy units. In locations ℓ_0, ℓ_1, the sporadic task S can occur (uncontrollable) and in this case it must be scheduled (going to ℓ_2) which consumes energy at rate $-\alpha$. The execution time of S is at most 1 t.u. (measured by clock x) and successive occurrences of S must be separated by at least 2 t.u. (measured by clock y.)

On this example, our symbolic algorithm terminates. If $\alpha = 3$, the HyTech program (in the full version of this paper) gives a minimal initial energy level of 3 to be able to win the game (notice that we start with $y = 2$ and thus the sporadic task can arrive at the initial instant.) The optimal strategy from the point of view of `Adam` is to trigger the sporadic task S as often as possible. While a winning strategy for `Eve` (scheduler) is to wait in location ℓ_0 as long as possible. If the sporadic task arrives again, it is not before 1 t.u. and thus we are rewarded by at least 3 energy credits. If the sporadic does not occur before $x = 2$, we get 6 energy credits, and we can switch to ℓ_1 which increases energy by 2. This ensures winning the energy game, see Fig. 11 for a graphical representation of the winning region. The set of winning state computed by the HyTech program can be used to determine the minimal initial credit for each possible initial state: for example $\mathit{energy} \geq 3$ is necessary for the initial condition $x = 0, y = 2$.)

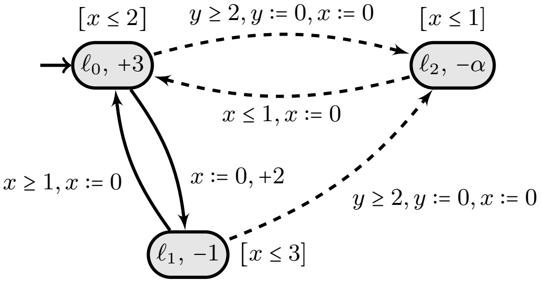

Figure 10: Scheduling Example

Now assume $\alpha = 4$. The previous strategy is not winning any more. However, the result of the HyTech program is now: $\mathit{energy} > 4$ (for $x = 0, y = 2$ as initial state.) In this case, while the minimal (infimum) initial credit is 4, there is no strategy realizing this value; the game cannot be won with an initial credit of 4 but rather with any value strictly above 4. Note that this information is collected by our symbolic algorithm but not by the operator $\mathtt{lift}_c$ as this operator is defined using inf, sup operators.

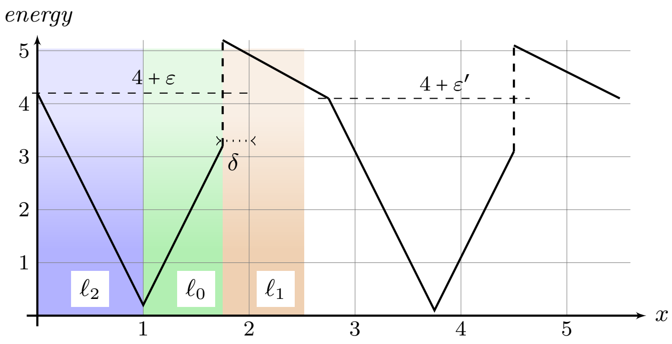

Figure 11: Winning Strategy for `Eve`, $\alpha = 4$.

6. REFERENCES

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

[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, Sept. 2002.

[3] R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In *HSCC*, volume 2034 of *LNCS*. Springer, 2001.

[4] G. Behrmann, A. Cougnard, A. David, E. Fleury, K. G. Larsen, and D. Lime. Uppaal-tiga: Time for playing games! In *CAV*, volume 4590 of *LNCS*. Springer, 2007.

[5] G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. W. Vaandrager.

Minimum-cost reachability for priced timed automata. In *HSCC*, volume 2034 of *LNCS*. Springer, 2001.

[6] H. Björklund, S. Sandberg, and S. G. Vorobyov. Memoryless determinacy of parity and mean payoff games: a simple proof. *Theoretical Computer Science*, 310(1-3):365–378, 2004.

[7] P. Bouyer. From qualitative to quantitative analysis of timed systems – Mémoire d'habilitation à diriger des recherches, 2009.

[8] P. Bouyer, T. Brihaye, V. Bruyère, and J.-F. Raskin. On the optimal reachability problem of weighted timed automata. *Formal Methods in System Design*, 31(2), 2007.

[9] P. Bouyer, T. Brihaye, M. Jurdzinski, R. Lazic, and M. Rutkowski. Average-price and reachability-price games on hybrid automata with strong resets. In *FORMATS*, volume 5215 of *LNCS*. Springer, 2008.

[10] P. Bouyer, E. Brinksma, and K. G. Larsen. Staying alive as cheaply as possible. In *HSCC*, volume 2993 of *LNCS*. Springer, 2004.

[11] P. Bouyer, F. Cassez, E. Fleury, and K. G. Larsen. Optimal Strategies in Priced Timed Game Automata. In *FSTTCS 2004*, pages 148–160, 2004.

[12] P. Bouyer, F. Cassez, E. Fleury, and K. G. Larsen. Synthesis of Optimal Strategies Using HyTech. *Electronic Notes in Theoretical Computer Science*, 119(1):11–31, 2005.

[13] P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba. Infinite runs in weighted timed automata with energy constraints. In *FORMATS*, volume 5215 of *LNCS*. Springer, 2008.

[14] P. Bouyer, K. G. Larsen, and N. Markey. Lower-bound constrained runs in weighted timed automata. In *QEST'12*, pages 128–137, London, UK, Sept. 2012. IEEE Computer Society Press.

[15] M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine. Kronos: A model-checking tool for real-time systems. In *CAV*, volume 1427 of *LNCS*. Springer, 1998.

[16] R. Brenguier, F. Cassez, and J.-F. Raskin. Energy and mean-payoff timed games. 2014. http://hal.archives-ouvertes.fr/docs/00/94/30/15/PDF/mainFull.pdf.

[17] T. Brihaye, V. Bruyère, and J.-F. Raskin. On optimal timed strategies. In *FORMATS*, volume 3829 of *LNCS*. Springer, 2005.

[18] L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, and J.-F. Raskin. Faster algorithms for mean-payoff games. *Formal methods in system design*, 38(2):97–118, 2011.

[19] F. Cassez, A. David, E. Fleury, K. G. Larsen, and D. Lime. Efficient on-the-fly algorithms for the analysis of timed games. In *CONCUR*, volume 3653 of *LNCS*. Springer, 2005.

[20] F. Cassez, J. J. Jessen, K. G. Larsen, J.-F. Raskin, and P.-A. Reynier. Automatic synthesis of robust and optimal controllers - an industrial case study. In *HSCC*, volume 5469 of *LNCS*. Springer, 2009.

[21] F. Cassez and K. G. Larsen. The impressive power of stopwatches. In *CONCUR*, volume 1877 of *LNCS*. Springer, 2000.

[22] K. Chatterjee, L. Doyen, and T. A. Henzinger. A survey of stochastic games with limsup and liminf objectives. In *ICALP*, volume 5556 of *LNCS, Part II*. Springer, 2009.

[23] K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin. Generalized mean-payoff and energy games. In *FSTTCS*, volume 8 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

[24] G. Frehse. PHAVer: algorithmic verification of hybrid systems past HyTech. *STTT*, 10(3):263–279, 2008.

[25] G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable Verification of Hybrid Systems. In *CAV*, volume 6806 of *LNCS*, pages 379–395. Springer, 2011.

[26] T. Henzinger. Quantitative reactive modeling and verification. *Computer Science Research and Development*, page 14, 2013.

[27] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. Hytech: A model checker for hybrid systems. *STTT*, 1(1-2):110–122, 1997.

[28] J. J. Jessen, J. I. Rasmussen, K. G. Larsen, and A. David. Guided controller synthesis for climate controller using Uppaal Tiga. In *FORMATS*, volume 4763 of *LNCS*. Springer, 2007.

[29] M. Jurdziński. Deciding the winner in parity games is in UP and in co-UP. *Information Processing Letters*, 68(3):119–124, 1998.

[30] M. Jurdzinski and A. Trivedi. Average-time games. In *FSTTCS*, volume 2 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2008.

[31] S. La Torre, S. Mukhopadhyay, and A. Murano. Optimal-reachability and control for acyclic weighted timed automata. In *IFIP TCS*, volume 2, pages 485–497, 2002.

[32] K. G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a nutshell. *STTT*, 1(1-2):134–152, 1997.

[33] O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems (an extended abstract). In *STACS*, pages 229–242, 1995.

[34] P. Tabuada. Controller synthesis for bisimulation equivalence. *Systems & Control Letters*, 57(6):443–452, 2008.