

Decidability of One-Clock Weighted Timed Games with Arbitrary Weights

Benjamin Monmege ✉ 

Aix Marseille Univ, CNRS, LIS, Marseille, France

Julie Parreaux ✉

Aix Marseille Univ, CNRS, LIS, Marseille, France

Pierre-Alain Reynier ✉

Aix Marseille Univ, CNRS, LIS, Marseille, France

Abstract

Weighted Timed Games (WTG for short) are the most widely used model to describe controller synthesis problems involving real-time issues. Unfortunately, they are notoriously difficult, and undecidable in general. As a consequence, one-clock WTG has attracted a lot of attention, especially because they are known to be decidable when only non-negative weights are allowed. However, when arbitrary weights are considered, despite several recent works, their decidability status was still unknown. In this paper, we solve this problem positively and show that the value function can be computed in exponential time (if weights are encoded in unary).

2012 ACM Subject Classification Software and its engineering → Formal software verification; Theory of computation → Algorithmic game theory

Keywords and phrases Weighted timed games, Algorithmic game theory, Timed automata

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.15

Related Version *Full Version*: <https://arxiv.org/abs/2207.01608> [21]

Funding This work was partially funded by ANR project Ticktac (ANR-18-CE40-0015) and ANR project DeLTA (ANR-16-CE40-0007).

1 Introduction

The task of designing programs is becoming more and more involved. Developing formal methods to ensure their correctness is thus an important challenge. Programs sensitive to real-time allow one to measure time elapsing in order to take decisions. The design of such programs is a notoriously difficult problem because timing issues may be intricate, and a posteriori debugging such issues is hard. The model of timed automata [2] has been widely adopted as a natural and convenient setting to describe real-time systems. This model extends finite-state automata with finitely many real-valued variables, called clocks, and transitions can check clocks against lower/upper bounds and reset some clocks.

Model-checking aims at verifying whether a real-time system modelled as a timed automaton satisfies some desirable property. Instead of verifying a system, one can try to synthesise one automatically. A successful approach, widely studied during the last decade, is one of the two-player games. In this context, a player represents the *controller*, and an antagonistic player represents the *environment*. Being able to identify a winning strategy of the controller, i.e. a recipe on how to react to uncontrollable actions of the environment, consists in the synthesis of a system that is guaranteed to be correct by construction.

In the realm of real-time systems, timed automata have been extended to timed games [3] by partitioning locations between the two players. In a turn-based fashion, the player that must play proposes a delay and a transition. The controller aims at satisfying some ω -regular



© Benjamin Monmege, Julie Parreaux, and Pierre-Alain Reynier; licensed under Creative Commons License CC-BY 4.0

33rd International Conference on Concurrency Theory (CONCUR 2022).

Editors: Bartek Klin, Slawomir Lasota, and Anca Muscholl; Article No. 15; pp. 15:1–15:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

objective however the environment player behaves. Deciding the winner in such turn-based timed games has been shown to be EXPTIME-complete [18], and a symbolic algorithm allowing tool development has been proposed [4].

In numerous application domains, in addition to real-time, other quantitative aspects have to be taken into account. For instance, one could aim at minimising the energy used by the system. To address this quantitative generalisation, weighted (aka priced) timed games (WTG for short) have been introduced [8, 5]. Locations and transitions are equipped with integer weights, allowing one to define the accumulated weight associated with a play. In this context, one focuses on a simple, yet natural, reachability objective: given some target location, the controller, that we now call *Min*, aims at ensuring that it will be reached while minimising the accumulated weight. The environment, that we now call *Max*, has the opposite objective: avoid the target location or, if not possible, maximise the accumulated weight. This allows one to define the value of the game as the minimal weight *Min* can guarantee. The associated decision problem asks whether this value is less than or equal to some given threshold.

In the earliest studies of this problem, [1, 8] proposed semi-decision procedures to approximate this value for WTG with non-negative weights. In addition, [8] identifies the subclass of strictly non-Zeno cost WTG for which their algorithm terminates. This approximation is motivated by the undecidability of the problem, first shown in [11]. This restriction has recently been lifted to WTG with arbitrary weights in [15].

An orthogonal research direction to recover decidability is to reduce the number of clocks and more precisely to focus on *one-clock WTG*. Though restricted, a single clock is often sufficient for modelling purposes. When only non-negative weights are considered, decidability has been proven in [10] and later improved in [22, 17] to obtain exponential time algorithms. Despite several recent works, the decidability status of one-clock WTG with arbitrary weights is still open. In the present paper, we show the decidability of the value problem for this class. More precisely, we prove that the value function can be computed in exponential time (if weights are encoded in unary and not in binary).

Before exposing our approach, let us briefly recap the existing results. Positive results obtained for one-clock WTG with non-negative weights are based on a reduction to so-called *simple WTG*, where the underlying timed automata contain no guard, no reset, and the clock value along with the execution exactly spans the $[0, 1]$ interval. In simple WTG, it is possible to compute (in exponential time) the whole value function starting at time 1 and going back in time until 0 [10, 22]. Another technique, that we will not explore further in the present work, consists in using the paradigm of strategy iteration [17], leading to an exponential-time algorithm too. A PSPACE lower-bound is also known for related decision problems [16].

More recent works extend the positive results of simple WTG to arbitrary weights [12, 13], yielding decidability of *reset-acyclic* one-clock WTG with arbitrary weights, with a *pseudo-polynomial time* complexity (that is polynomial if weights are encoded in unary). It is also explained how to extend the result to all WTG where no cyclic play containing a reset may have a negative weight arbitrarily close to 0. Moreover, it is shown that *Min* needs memory to play (almost-)optimally, in a very structured way: *Min* uses *switching strategies*, that are composed of two memoryless strategies, the second one being triggered after a given (pseudo-polynomial) number κ of steps.

The crucial ingredient to obtain decidability for non-negative weights or reset-acyclic weighted timed games is to limit the number of reset transitions taken along a play. This is no longer possible in presence of cycles of negative weights containing a reset. There, *Min*

may need to iterate cycles for a number κ of times that depends on the desired precision ε on the value (to play ε -optimally, Min needs to cycle $O(1/\varepsilon)$ times, see Example 3). To rule out these annoying behaviours, we rely on three main ingredients:

- As there is a single clock, a cyclic path ending with a reset corresponds to a cycle of configurations. We define the *value* of such a cycle, that allows us to identify which player may benefit from iterating it.
- Using the classical region graph construction, we prove stronger properties on the value function (it is continuous on the closure of region intervals). This allows us to prove that Max has an optimal memoryless strategy that avoids cycles whose value is negative.
- We introduce a partial unfolding of the game, so as to obtain an acyclic WTG, for which decidability is known. To do so, we rely on the existence of (almost-)optimal switching strategies for Min, allowing us to limit the depth of exploration. Also we keep track of cycles encountered and handle them according to their value. Using the previous result on the existence of a “smart” optimal strategy for Max, we show that this unfolding has the same value as the original WTG.

The paper is organized as follows: weighted timed games are presented in Section 2. We then focus on cycles in Section 3. Our unfolding is presented in Section 4, with a sketch of the main proof. Some of the technical proofs can be found in Appendix, and a long version is available with all the proofs [21].

2 Weighted timed games

2.1 Definitions

We will consider weighted timed games with a single clock, denoted by x . The valuation of this clock is a non-negative real number ν . On such a clock, transitions of the timed games will be able to check some interval constraints on the clock, i.e. intervals I of real values with closed or open bounds that are natural numbers (or $+\infty$). For every interval $I = (a, b)$ we denote by $\bar{I} = [a, b]$ its closure.

- **Definition 1.** A weighted timed game is a tuple $\langle Q_{\text{Min}}, Q_{\text{Max}}, Q_t, Q_u, \Delta, \text{wt}, \text{wt}_t \rangle$ with
- $Q = Q_{\text{Min}} \uplus Q_{\text{Max}} \uplus Q_t$ a finite set of locations split between players Min and Max (in drawings, locations belonging to Min are depicted by circles and the ones belonging to Max by squares) and a set of target locations;
 - $Q_u \subseteq Q_{\text{Min}} \uplus Q_{\text{Max}}$ a set of urgent locations where time cannot be delayed;
 - Δ a finite set of transitions each of the form $q \xrightarrow{I, R, w} q'$, with q and q' two locations (with $q \notin Q_t$), I an interval, $w \in \mathbb{Z}$ being the weight of the transition, and R being either $\{x\}$ when the clock must be reset (depicted by $x := 0$), or \emptyset when it does not;
 - $\text{wt}: Q \rightarrow \mathbb{Z}$ a weight function associating an integer weight with each location: for uniformisation of the notations, we extend this weight function to also associate with each transition the weight it contains, i.e. $\text{wt}(q \xrightarrow{I, R, w} q') = w$;
 - and $\text{wt}_t: Q_t \times \mathbb{R}_{\geq 0} \rightarrow \bar{\mathbb{R}}$ a function mapping each target configuration to a final weight, where $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$.

The addition of final weights in weighted timed games (WTG) is not standard, but we use it in the process of solving those games: in any case, it is possible to simply map a given target location to the weight 0, allowing us to recover the standard definitions of the literature. The presence of urgent locations is also unusual: in a timed automaton with several clocks, urgency can be modelled with an additional clock u that is reset just before

15:4 Decidability of One-Clock Weighted Timed Games with Arbitrary Weights

entering the urgent location and with constraints $u \in [0, 0]$ on outgoing transitions. However, when limiting the number of clocks to one, we regain modelling capabilities by allowing for such urgent locations. The weight of an urgent location is never used and will thus not be given in drawings: instead, urgent locations will be displayed with a u inside.

The semantics of a WTG \mathcal{G} is defined in terms of an infinite transition system $\llbracket \mathcal{G} \rrbracket$ whose vertices are configurations $(q, \nu) \in Q \times \mathbb{R}_{\geq 0}$. Configurations are split into players according to the location q , and a configuration (q, ν) is a target if $q \in Q_t$. Edges linking vertices will be labelled by elements of $\mathbb{R}_{\geq 0} \times \Delta$, to encode the delay that a player wants to spend in the current location, before firing a certain transition. For every delay $t \in \mathbb{R}_{\geq 0}$, transition $\delta = q \xrightarrow{I, R, w} q' \in \Delta$ and valuation ν , we add a labelled edge $(q, \nu) \xrightarrow{t, \delta} (q', \nu')$ if

- $\nu + t \in I$;
- $\nu' = 0$ if $R = \{x\}$, and $\nu' = \nu + t$ otherwise;
- and $t = 0$ if $q \in Q_u$.

This edge is given a weight $t \times \text{wt}(q) + \text{wt}(\delta)$ taking into account discrete and continuous weights.

As usual in related work [1, 8, 9], we will assume that the valuation of the clock x is *bounded* by the greatest constant M to appear in guards, and we, therefore, restrict ourselves to configurations of the form $(q, \nu) \in Q \times [0, M]$. We also suppose the absence of deadlocks except on target locations, i.e. for each location $q \in Q \setminus Q_t$ and valuation $\nu \in [0, M]$, there exist $t \in \mathbb{R}_{\geq 0}$ and $\delta = q \xrightarrow{I, R, w} q' \in \Delta$ such that $(q, \nu) \xrightarrow{t, \delta} (q', \nu')$, and no transitions start from Q_t . This second restriction is without loss of generality by applying classical techniques [6, Lemma 5].

We also assume that the final weight functions satisfy a sufficient property ensuring that they can be encoded in finite space. First, we call *regions*¹ of \mathcal{G} the set

$$\text{Reg}_{\mathcal{G}} = \{(M_i, M_{i+1}) \mid 0 \leq i \leq k-1\} \cup \{\{M_i\} \mid 0 \leq i \leq k\}$$

where $M_0 = 0 < M_1 < \dots < M_k$ are all the endpoints of the intervals appearing in the guards of \mathcal{G} (to which we add 0 if needed). Then, we require final weight functions to be piecewise affine with a finite number of pieces and continuous on each region. More precisely, we assume that cutpoints and coefficients are rational and given in binary.

We let W_{loc} , W_{tr} and W_{fin} be the maximum absolute value of weights of locations, transitions and final functions, i.e.

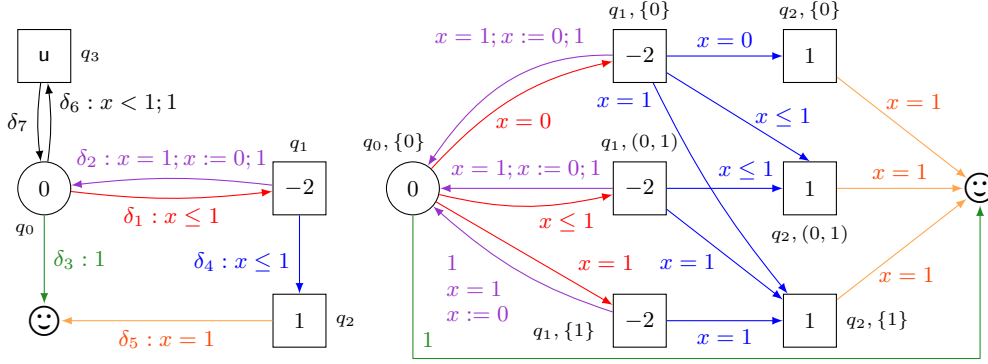
$$W_{\text{loc}} = \max_{q \in Q} |\text{wt}(q)| \quad W_{\text{tr}} = \max_{\delta \in \Delta} |\text{wt}(\delta)| \quad W_{\text{fin}} = \sup_{q \in Q_t \text{ s.t. } \text{wt}_t(q, \cdot) \notin \{+\infty, -\infty\}} \sup_{\nu \in I} |\text{wt}_t(q, \nu)|$$

We also let W be the maximum of W_{loc} , W_{tr} , and W_{fin} .

We call *path* a finite or infinite sequence of consecutive transitions $\delta_0, \delta_1, \dots$ of Δ , that we sometimes denote by $q_0 \xrightarrow{\delta_0} q_1 \xrightarrow{\delta_1} q_2 \dots$. We let FPaths be the set of all finite paths. We let $|\pi|$ be the number of transitions in the finite path π , that we call its *length*. For a given transition δ , we let $|\pi|_{\delta}$ denote the number of occurrences of δ in π .

We call *play* a finite or infinite sequence of edges in the semantics of the game that we denote by $(q_0, \nu_0) \xrightarrow{t_0, \delta_0} (q_1, \nu_1) \xrightarrow{t_1, \delta_1} (q_2, \nu_2) \dots$. A play is said to *follow* a path if both use the same sequence of transitions. We let $|\rho|$ be the *length* of play, defined as the length of the path it follows. We let $|\rho|_{\delta}$ be the number of occurrences of the transition δ in the finite play ρ . More generally, for all sets of transitions A , we let $|\rho|_A$ be the number of

¹ This is inspired by a construction by Laroussinie, Markey, and Schnoebelen [19], which allows one to reduce the number of regions with respect to the more usual one of [2] in the case of a single clock.



■ **Figure 1** On the left, a WTG with a cyclic path of weight $[-1, 1]$ containing a reset. Missing weights are 0. The target location is \ominus , whose final weight function is zero. Location q_3 is urgent. On the right, its closure restricted to locations q_0, q_1, q_2 and \ominus .

occurrences of all transitions from A in the finite play ρ , i.e. $|\rho|_A = \sum_{\delta \in A} |\rho|_{\delta}$. We let FPlays be the set of finite plays. For a finite path π or a finite play ρ , we let $\text{last}(\pi)$ and $\text{last}(\rho)$ be the last location or configuration. We let $\text{FPaths}_{\text{Max}}$ (respectively, $\text{FPaths}_{\text{Min}}$) and $\text{FPlays}_{\text{Max}}$ (respectively, $\text{FPlays}_{\text{Min}}$) be the subset of finite paths or plays whose last element belong to player Max (respectively, Min).

A finite play ρ can be associated with a weight that consists in accumulating the weight of the edges it traverses: if $\rho = (q_0, \nu_0) \xrightarrow{t_0, \delta_0} (q_1, \nu_1) \cdots (q_k, \nu_k)$, we let

$$\text{wt}_{\Sigma}(\rho) = \sum_{i=0}^{k-1} (\text{wt}(\ell_i) \times t_i + \text{wt}(\delta_i)).$$

A maximal play ρ (either infinite or trapped in a deadlock that is necessarily a target configuration) is associated with a payoff $\text{P}(\rho)$ as follows: the payoff of an infinite play (meaning that it never visits a target location) is $+\infty$, while the payoff of a finite play, thus ending in a target configuration (q, ν) , is $\text{wt}_{\Sigma}(\rho) + \text{wt}_t(q, \nu)$. The weight of a finite path π consists of the set of the cumulated weight of all the finite plays that follow π : $\text{wt}_{\Sigma}(\pi) = \{\text{wt}_{\Sigma}(\rho) \mid \rho \text{ following } \pi\}$. By [5], the weight of a path is known to be an interval of values. Moreover, when all the guards along the path are closed intervals, the weight of the path is also a closed interval.

A *cyclic path* is a finite path that starts and ends in the same location. A *cyclic play* is a finite play that starts and ends in the same configuration: it necessarily follows a cyclic path, but the reverse might not be true since some non-cyclic plays can follow a cyclic path (if they do not end in the same clock valuation as the one in which they start).

► **Example 2.** The cyclic path $\pi = q_0 \xrightarrow{\delta_1} q_1 \xrightarrow{\delta_2} q_0$ depicted on the left in Figure 1 has a weight between -1 (with the play $(q_0, 0) \xrightarrow{0, \delta_1} (q_1, 0) \xrightarrow{1, \delta_2} (q_0, 0)$) and 1 (with the play $(q_0, 0) \xrightarrow{1, \delta_1} (q_1, 1) \xrightarrow{0, \delta_2} (q_0, 0)$), so $\text{wt}_{\Sigma}(\pi) = [-1, 1]$. Another cyclic path is $\pi' = q_0 \xrightarrow{\delta_6} q_3 \xrightarrow{\delta_7} q_0$ which goes via an urgent location. All plays that follow this path are of the form $(q_0, \nu) \xrightarrow{t, \delta_6} (q_3, \nu + t) \xrightarrow{0, \delta_7} (q_0, \nu + t)$ with ν and $\nu + t$ less than 1, that all have a weight 1. Thus $\text{wt}_{\Sigma}(\pi') = \{1\}$.

A *strategy* aims at giving the recipe of each player. A strategy of Min is a function $\sigma: \text{FPlays}_{\text{Min}} \rightarrow \mathbb{R}_{\geq 0} \times \Delta$ mapping each finite play ρ whose last configuration belongs to Min to a pair (t, δ) of delay and transition, such that the play ρ can be extended by an edge

labelled with (t, δ) . A similar definition holds for strategies τ of Max. We let $\text{Strat}_{\text{Min}, \mathcal{G}}$ (respectively, $\text{Strat}_{\text{Max}, \mathcal{G}}$) be the set of strategies of Min (respectively, Max) in the game \mathcal{G} , or simply $\text{Strat}_{\text{Min}}$ and $\text{Strat}_{\text{Max}}$ if the game is clear from the context: we will always use letters σ and τ to differentiate from strategies of Min and Max.

A strategy is said to be *memoryless* if it only depends on the last configuration of the plays. More formally, Max's strategy τ is memoryless if for all plays ρ and ρ' such that $\text{last}(\rho) = \text{last}(\rho')$, we have $\tau(\rho) = \tau(\rho')$.

A play ρ is said to be *conforming* to a strategy σ (respectively, τ) if the choice made in ρ at each location of Min (respectively, Max) is the one prescribed by σ (respectively, τ). Moreover, a finite path π is said to be conforming to a strategy σ (respectively, τ) if there exists a finite play following π that is conforming to σ (respectively, τ).

After both players have chosen their strategies σ and τ , each initial configuration (q, ν) gives rise to a unique maximal play that we denote by $\text{Play}((q, \nu), \sigma, \tau)$. The *value* of the configuration (q, ν) is then obtained by letting players choose their strategies as they want, first Min and then Max, or vice versa since WTG is known to be determined [12]:

$$\text{Val}_{\mathcal{G}}(q, \nu) = \sup_{\tau} \inf_{\sigma} \text{P}(\text{Play}((q, \nu), \sigma, \tau)) = \inf_{\sigma} \sup_{\tau} \text{P}(\text{Play}((q, \nu), \sigma, \tau)).$$

The value of a strategy σ of Min (symmetric definitions can be given for strategies τ of Max) is defined as $\text{Val}_{\mathcal{G}}^{\sigma}(q, \nu) = \sup_{\tau} \text{P}(\text{Play}((q, \nu), \sigma, \tau))$. Then, a strategy σ^* of Min is *optimal* if, for all initial configurations (q, ν) , $\text{Val}_{\mathcal{G}}^{\sigma^*}(q, \nu) \leq \text{Val}_{\mathcal{G}}(q, \nu)$. Because of the infinite nature of the timed games, optimal strategies may not exist: for example, a player may want to let time elapse as much as possible, but with a delay $t < 1$ because of a strict guard, preventing them to obtain the optimal value. We will see in Example 12 that this situation can even happen when all guards contain only *closed* comparisons. We naturally extend the definition to *almost-optimal strategies*, taking into account small possible errors: we say that a strategy σ^* of Min is ε -*optimal* if, for all initial configurations (q, ν) , $\text{Val}_{\mathcal{G}}^{\sigma^*}(q, \nu) \leq \text{Val}_{\mathcal{G}}(q, \nu) + \varepsilon$.

► **Example 3.** We have seen that in q_0 (on the left in Figure 1), Min has no interest in following the cycle $q_0 \xrightarrow{\delta_6} q_3 \xrightarrow{\delta_7} q_0$ since it has weight $\{1\}$. Jumping directly to the target location via δ_3 leads to a weight of 1. But Min can do better: from valuation 0, by jumping to q_1 after a delay of $t \leq 1$, it leaves a choice to Max to either jump to q_2 and the target leading to a total weight of $1 - t$, or to loop back in q_0 thus closing a cyclic play of weight $-2(1 - t) + 1 = 2t - 1$. If t is chosen too close to 1, the value of the cycle is greater than 1, and Max will benefit from it by increasing the total weight. If t is chosen as smaller than $1/2$, the weight of the cycle is negative, and Max will prefer to go to the target to obtain a weight $1 - t$ close to 1, not very beneficial to Min. Thus, Min prefers to play just above $1/2$, let say at $1/2 + \varepsilon$. In this case, Max will choose to go to the target with a total weight of $1/2 + \varepsilon$. The value of the game, in configuration $(q_0, 0)$ is thus $\text{Val}_{\mathcal{G}}(q_0, 0) = 1/2$. Not only Min does not have an optimal strategy (but only ε -optimal ones, for every $\varepsilon > 0$), but needs memory to play ε -optimally, since Min cannot play *ad libitum* transition δ_2 with a delay $1/2 - \varepsilon$: in this case, Max would prefer staying in the cycle, thus avoiding the target. Thus, Min will play the transition δ_1 at least $1/4\varepsilon$ times so that the cumulated weight of all the cycles is below $-1/2$, in which case Min can safely use transition δ_1 still earning $1/2$ in total.

2.2 Closure

We first recall more in details the method used to solve WTG in [12], starting with the (slightly updated presentation of the) construction that consists in enhancing the locations with regions and closing all guards while preserving the value of the game.

► **Definition 4.** The closure of a WTG \mathcal{G} is the WTG $\overline{\mathcal{G}} = \langle L_{\text{Min}}, L_{\text{Max}}, L_t, L_u, \overline{\Delta}, \overline{\text{wt}}, \overline{\text{wt}_t} \rangle$ where:

- $L = L_{\text{Min}} \uplus L_{\text{Max}} \uplus L_t$ with $L_{\text{Min}} = Q_{\text{Min}} \times \text{Reg}_{\mathcal{G}}$, $L_{\text{Max}} = Q_{\text{Max}} \times \text{Reg}_{\mathcal{G}}$, $L_t = Q_t \times \text{Reg}_{\mathcal{G}}$, $L_u = Q_u \times \text{Reg}_{\mathcal{G}}$;
- for all $(q, I) \in L$, $(q, I) \xrightarrow{I_g \cap I'', R, w} (q', I') \in \overline{\Delta}$ if and only if there exist a transition $q \xrightarrow{I_g, R, w} q' \in \Delta$, and a region I'' such that $I_g \cap I'' \neq \emptyset$, the lower bound of I'' is at least the one of I (to model time elapsing), and I' is equal to I'' if $R = \emptyset$ and to $\{0\}$ otherwise: $\overline{I_g \cap I''}$ stands for the topological closure of the non-empty interval $I_g \cap I''$;
- for all (q, I) , we have $\overline{\text{wt}}(q, I) = \text{wt}(q)$;
- for all $(q, I) \in L_t$, for $\nu \in I$, $\overline{\text{wt}_t}((q, I), \nu) = \text{wt}_t(q, \nu)$ and extend $\nu \mapsto \overline{\text{wt}_t}((q, I), \nu)$ by continuity on \overline{I} , the closure of the interval I . We may also let $\overline{\text{wt}_t}((q, I), \nu) = +\infty$ for all $\nu \notin \overline{I}$, even though we will never use this in the following.

The following set of configurations is an invariant of the closure (i.e. starting from such configuration fulfilling the invariant, we can only reach configurations fulfilling the invariant):

- configurations $((q, \{M_k\}), M_k)$;
- and configurations $((q, (M_k, M_{k+1})), \nu)$ with $\nu \in [M_k, M_{k+1}]$ (and not only in (M_k, M_{k+1}) as one might expect).

► **Example 5.** Figure 1 depicts the closure (left) of the WTG (right) restricted to locations q_0, q_1, q_2 , and \ominus (we have seen that q_3 is anyway useless).

The closure of the guards allows players to mimic a move in \mathcal{G} “arbitrarily close” to M_{k+1} in (M_k, M_{k+1}) to be simulated by jumping on M_{k+1} still in the region (M_k, M_{k+1}) .

► **Lemma 6** ([12]). For all WTG \mathcal{G} , $(q, I) \in Q \times \text{Reg}_{\mathcal{G}}$ and $\nu \in I$, $\text{Val}_{\mathcal{G}}(q, \nu) = \text{Val}_{\overline{\mathcal{G}}}((q, I), \nu)$.

It is also shown in [12] that we can transform an ε -optimal strategy of $\overline{\mathcal{G}}$ into an ε' -optimal strategy of \mathcal{G} with $\varepsilon' < 2\varepsilon$ and vice-versa. Not only the closure construction adds the capability for a player to play “arbitrarily close” to the border of a region as a new move, but it also makes the value function more manageable for our purpose. Indeed, as shown in [12], the mapping $\nu \mapsto \text{Val}_{\mathcal{G}}(\ell, \nu)$ is continuous over all regions, but there might be discontinuities at the borders of the regions. The closure construction clears this issue by softening the borders of each region independently:

► **Lemma 7.** For all WTG \mathcal{G} and $(q, I) \in Q \times \text{Reg}_{\mathcal{G}}$, the mapping $\nu \mapsto \text{Val}_{\overline{\mathcal{G}}}((q, I), \nu)$ is continuous over \overline{I} .

In [12], it is also shown that the mapping $\nu \mapsto \text{Val}_{\mathcal{G}}(\ell, \nu)$ is piecewise affine on each region where it is not infinite, that the total number of pieces (and thus of *cutpoints*, in-between two such affine pieces) is exponential, and that all cutpoints and the value associated to such a cutpoint are rational numbers. In more recent developments in [13], authors improve the exponential complexity into *pseudo-polynomial* (i.e. polynomial in the number of locations and in the biggest weight W), which we will use in the sequel. Thus, they obtain:

► **Theorem 8** ([13]). If \mathcal{G} is an acyclic WTG (i.e. that does not contain cyclic path), then for all locations q , the piecewise affine mapping $\nu \mapsto \text{Val}_{\mathcal{G}}(q, \nu)$ is computable in time polynomial in $|Q|$ and W .

In [13], this result is slightly extended to take into account cyclic paths containing resets when their weight is either non-negative, or not arbitrarily close to 0.

► **Example 9.** Notice that the game on the left in Figure 1 does not fulfil this hypothesis: indeed the play $(q_0, 0) \xrightarrow{1/2-\varepsilon, \delta_1} (q_1, 1/2-\varepsilon) \xrightarrow{1/2+\varepsilon, \delta_2} (q_0, 0)$ is a cyclic play of weight -2ε negative and arbitrarily close to 0.

2.3 Contribution

In this work, we use a different technique to push the decidability frontier, and prove that the value function is computable for all WTG (in particular the one of Figure 1):

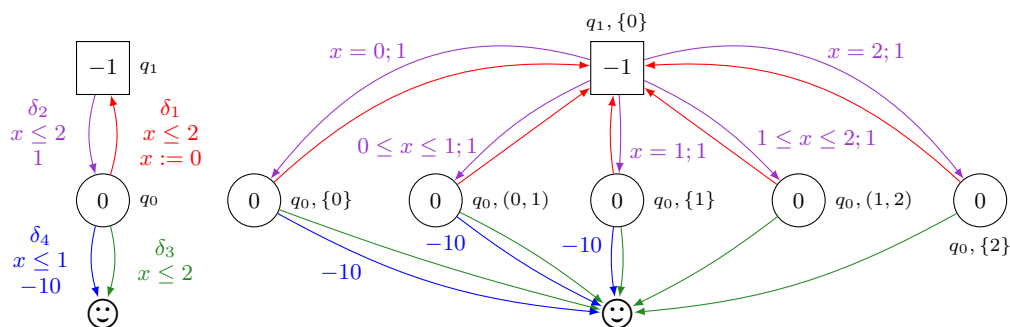
► **Theorem 10.** *For all WTG \mathcal{G} and all locations q_i , the mapping $\nu \mapsto \text{Val}_{\mathcal{G}}(q_i, \nu)$ is computable in time exponential in $|Q|$ and W_{tr} , and polynomial in W_{loc} and W_{fin} .*

► **Remark 11.** The complexities of Theorems 8 and 10 would be more traditionally considered as exponential and doubly-exponential if weights of the WTG were encoded in binary as usual. In this work, we thus count the complexities as if all weights were encoded in unary and thus consider W to be the bound of interest. For Theorem 8, the obtained bound is classically called *pseudo-polynomial* in the literature.

The rest of this article gives the proof of Theorem 10. We fix a WTG \mathcal{G} and an initial location q_i . We let $\bar{\mathcal{G}} = \langle L_{\text{Min}}, L_{\text{Max}}, L_t, L_u, \bar{\Delta}, \bar{\text{wt}}, \bar{\text{wt}}_t \rangle$ be its closure. We first use Lemma 6 which allows us to deduce the result by computing the value functions $\nu \mapsto \text{Val}_{\bar{\mathcal{G}}}((q_i, I), \nu)$, for all regions I . Regions I over which $\nu \mapsto \text{Val}_{\bar{\mathcal{G}}}((q_i, I), \nu)$ is constantly equal to $+\infty$ or $-\infty$ are computable in polynomial-time, as explained in [12]. We, therefore, remove them from $\bar{\mathcal{G}}$ from now on. We now fix an initial region I_i and let $\ell_i = (q_i, I_i)$.

As in the non-negative case [10], the objective is to limit the number of transitions with a reset taken into the plays while not modifying the value of the game. When all weights are non-negative, this is fairly easy to achieve since, intuitively speaking, Min has no interest in using any cycles containing such a transition (since it has non-negative weight and is thus non-beneficial for Min). The game can thus be transformed so that each transition with a reset is taken at most once. To obtain a smaller game, it is even possible to simply count the number of transitions with a reset taken so far in the play and stop the game (with a final weight $+\infty$) in case the counter goes above the number of such transitions in the game. The transformed game has a polynomial number of locations with respect to the original game, and is reset-acyclic, which allows one to solve it with a pseudo-polynomial time complexity (instead of the exponential-time complexity originally achieved in [10, 22]).

The situation is much more intricate in the presence of negative weights since negative cycles containing a transition with a reset can be beneficial for Min, as we have seen in Example 3. Notice that this is still true in the closure of the game, as can be checked on the right in Figure 1. Moreover, some cyclic paths may have an interval of possible weights with both positive and negative values, making it difficult to determine whether it is beneficial to Min or not. To overcome this situation, we will consider the point of view of Max, making a profit from the determinacy of the WTG. We will show that, in a closed game $\bar{\mathcal{G}}$, Max can play *optimally* with *memoryless* strategies while *avoiding negative cyclic plays*. This will simplify our further study since, by following this strategy, Max ensures that only non-negative cyclic plays will be encountered, which is not beneficial to Min. Therefore, as in [10], we will limit the firing of transitions with a reset to at most once. However, we are not able to do it without blowing up exponentially the number of locations of the games. Instead, along the unfolding of the game, we need to record enough information in order to know, in case a cyclic path ending with a reset is closed, whether this cyclic path has a potential negative weight (in which case Max will indeed not follow it) or non-negative weight (in which case it is not beneficial for Min to close the cycle). Determining in which case we are will be made possible by introducing the notion of value of a cyclic path in Section 3. Then, Max has even an optimal strategy to avoid closing cyclic paths with negative value (which is stronger than only avoiding creating negative cyclic plays). The unfolding, denoted \mathcal{U} , will be defined in Section 4. In order to prove that it is a game equivalent to $\bar{\mathcal{G}}$, we will prove that Max can do as well as in \mathcal{U} from $\bar{\mathcal{G}}$ and vice-versa.



■ **Figure 2** On the left, a WTG where Max needs memory to play ε -optimally. On the right, its closure where we merged several transitions by removing unnecessary guards.

3 Controlling negative cycles

One of the main arguments of our proof is that, in the closed game $\bar{\mathcal{G}}$, Max can play *optimally* with *memoryless* strategies while *avoiding negative cyclic plays*. As already noticed in [12], this is not always true in non-closed games: Max may need memory to play ε -optimally without the possibility to avoid some negative cyclic plays.

► **Example 12.** In the WTG \mathcal{G} depicted on the left in Figure 2, we can see that $\text{Val}(q_1, 0) = 0$, but Max does not have an optimal strategy, needs memory to play ε -optimally, and cannot avoid negative cyclic plays. Indeed, if at some point the strategy of Max chooses a delay less than or equal to 1, then Min can always choose δ_4 , and the value of this strategy is -10 . Thus, an optimal strategy for Max always chooses a delay greater than 1. However, Max must choose a delay closer and closer to 1. Otherwise, if there exists $\beta > 0$ such that all delays chosen by the strategy are greater than $1 + \beta$, Min has a family of strategies with a value that will tend to $-\infty$ by staying longer and longer in the cycle with a weight at most $-\beta$. Thus, Max does not have an optimal strategy, and the ε -optimal strategy requires infinite memory to play with delays closer and closer to 1 (for instance, after the n th round in the cycle, Max delays $\varepsilon/2^n$ time units, to sum up, all weights to a value at most $-\varepsilon$).

Such convergence phenomena needed by Max do not exist in $\bar{\mathcal{G}}$ since all guards are closed (this is not sufficient alone though) and by the regularity of Val given by Lemma 7.

► **Example 13.** We consider the closed game depicted on the right in Figure 2. The ε -optimal strategy (with memory) of Max in \mathcal{G} translates into an optimal memoryless strategy in $\bar{\mathcal{G}}$: in $(q_1, \{0\})$, Max can delay 1 time unit and jump into the location $(q_0, (1, 2))$. Then cyclic plays that Min can create have a zero weight and are thus not profitable for either player.

To generalise this explanation, we start by defining the value of cyclic paths ending with a reset in $\bar{\mathcal{G}}$. Intuitively, the value of this cyclic path is the weight that Min (or Max) can guarantee regardless of the delays chosen by Max (or Min) during this one.

► **Definition 14.** We define by induction the value $\text{Val}_{\bar{\mathcal{G}}}^{\nu}(\pi)$ of a finite path π in $\bar{\mathcal{G}}$ from an initial valuation ν of the clock: if π has length 0, we let $\text{Val}_{\bar{\mathcal{G}}}^{\nu}(\pi) = 0$, otherwise, π can be written $\ell_0 \xrightarrow{\delta_0} \pi'$ (with π' starting in location ℓ_1), and we let

$$\text{Val}_{\bar{\mathcal{G}}}^{\nu}(\pi) = \begin{cases} \inf_{t_0} (t_0 \text{wt}(\ell_0) + \text{wt}(\delta_0) + \text{Val}_{\bar{\mathcal{G}}}^{\nu'}(\pi')) & \text{if } \ell_0 \in L_{\text{Min}} \\ \sup_{t_0} (t_0 \text{wt}(\ell_0) + \text{wt}(\delta_0) + \text{Val}_{\bar{\mathcal{G}}}^{\nu'}(\pi')) & \text{if } \ell_0 \in L_{\text{Max}} \end{cases}$$

where t_0 and ν' are such that $(\ell_0, \nu) \xrightarrow{t_0, \delta_0} (\ell_1, \nu')$ is an edge of $\llbracket \bar{\mathcal{G}} \rrbracket$. Then, for a cyclic path π of $\bar{\mathcal{G}}$ ending by a transition with a reset, we let $\text{Val}_{\bar{\mathcal{G}}}(\pi) = \text{Val}_{\bar{\mathcal{G}}}^0(\pi)$.

15:10 Decidability of One-Clock Weighted Timed Games with Arbitrary Weights

The value of a cyclic path belongs to the interval $\text{wt}_\Sigma(\pi)$ and corresponds to the weight of a cyclic play that follows this path.

► **Example 15.** Let $\pi = (q_0, \{0\}) \xrightarrow{\delta_1} (q_1, (0, 1)) \xrightarrow{\delta_2} (q_0, \{0\})$ be the cyclic path of the game $\bar{\mathcal{G}}$ depicted on the right in Figure 1, for which $\text{wt}_\Sigma(\pi) = [-1, 1]$. To evaluate the value of π , Min only needs to choose a delay $t_1 \in [0, 1]$ when firing δ_1 , while Max has no choice but to play a delay $1 - t_1$ when firing δ_2 , generating a finite play ρ of weight $\text{wt}_\Sigma(\rho) = 2t_1 - 1$. We deduce that $\text{Val}_{\bar{\mathcal{G}}}(\pi) = \inf_{t_1} (2t_1 - 1) = -1$ (when Min chooses $t_1 = 0$).

A cyclic path with a negative value ensures that Min can guarantee a cyclic play with a negative weight that follows it, but there may exist other cyclic plays with a non-negative weight that follows it. It is exactly those cycles that are problematic for Max since Min can benefit from them. We now show our key lemma: in the closed game, Max can play optimally and avoid cyclic paths of negative value.

► **Lemma 16.** *In $\bar{\mathcal{G}}$ (where regions with infinite value had been remote), Max has a memoryless optimal strategy τ^* such that*

1. *all cyclic plays conforming to τ^* have a non-negative weight;*
2. *all cyclic paths ending by a reset conforming to τ^* have a non-negative value.*

Sketch of proof. We build upon the fact [8, 7] that the value function $\text{Val}_{\bar{\mathcal{G}}}: L \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ is a fixed point (even the greatest one) of the operator \mathcal{F} defined as follows: for all configurations (ℓ, ν) and all mappings $X: L \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, we let

$$\mathcal{F}(X)(\ell, \nu) = \begin{cases} \overline{\text{wt}}_t(\ell, \nu) & \text{if } \ell \in L_t \\ \inf_{\ell, \nu \xrightarrow{t, \delta} \ell', \nu'} (\overline{\text{wt}}(\delta) + t \overline{\text{wt}}(\ell) + X(\ell', \nu')) & \text{if } \ell \in L_{\text{Min}} \\ \sup_{\ell, \nu \xrightarrow{t, \delta} \ell', \nu'} (\overline{\text{wt}}(\delta) + t \overline{\text{wt}}(\ell) + X(\ell', \nu')) & \text{if } \ell \in L_{\text{Max}} \end{cases}$$

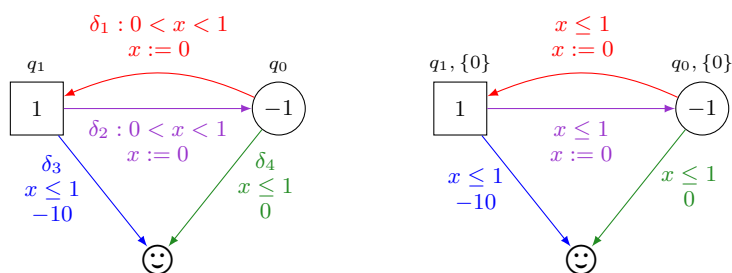
We use this fact to define the memoryless strategy τ^* . Indeed, the identity $\text{Val}_{\bar{\mathcal{G}}} = \mathcal{F}(\text{Val}_{\bar{\mathcal{G}}})$, applied over configurations belonging to Max suggests a choice of transition and delay to play almost optimally. As \mathcal{F} computes a supremum on the set of possible (transitions and) delays, this does not directly lead to a specific choice: in general, this would give rise to ε -optimal strategies and not an optimal one. This is where we rely on the continuity of $\text{Val}_{\bar{\mathcal{G}}}$ (Lemma 7) on each closure of region to deduce that this supremum is indeed a maximum. More precisely, for $\ell \in L_{\text{Max}}$, we can write $\mathcal{F}(\text{Val}_{\bar{\mathcal{G}}})(\ell, \nu)$ as

$$\max_{\delta \in \bar{\Delta}} \sup_{t \text{ s.t. } \ell, \nu \xrightarrow{t, \delta} \ell', \nu'} (\overline{\text{wt}}(\delta) + t \overline{\text{wt}}(\ell) + \text{Val}_{\bar{\mathcal{G}}}(\ell', \nu')).$$

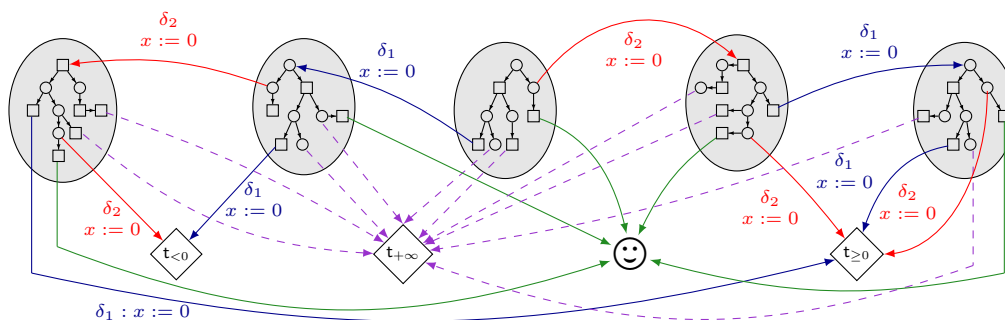
The guard of transition δ is the closure \bar{I} of a region $I \in \text{Reg}_{\mathcal{G}}$, therefore, t is in a closed interval J of values such that $\nu + t$ falls in \bar{I} . Notice that ν' is either 0 if δ contains a reset or is $\nu + t$: in both cases, this is a continuous function of t . Relying on the continuity of $\text{Val}_{\bar{\mathcal{G}}}$, the mapping $t \in J \mapsto \overline{\text{wt}}(\delta) + t \overline{\text{wt}}(\ell) + \text{Val}_{\bar{\mathcal{G}}}(\ell', \nu')$ is thus continuous over a compact set so that its supremum is indeed a maximum. We thus let the memoryless strategy τ^* be such that, for all configurations (ℓ, ν) , $\tau^*(\ell, \nu)$ is chosen arbitrarily in

$$\arg\max_{\delta \in \bar{\Delta}} \arg\max_{t \text{ s.t. } \ell, \nu \xrightarrow{t, \delta} \ell', \nu'} (\overline{\text{wt}}(\delta) + t \overline{\text{wt}}(\ell) + \text{Val}_{\bar{\mathcal{G}}}(\ell', \nu')) \quad (1)$$

The strategy τ^* is then extended to finite plays by considering only the last configuration of the play. We can show that τ^* is an optimal strategy that satisfies the two properties of the lemma. ◀



■ **Figure 3** On the left, a WTG such that its closure on the right contains a cyclic path with a weight $[-1, 1]$ and a value 0. Moreover Max uses the cyclic path to play optimally.



■ **Figure 4** Scheme of the unfolding of a closed game.

Lemma 16 does not allow us to conclude on the decidability of the value problem since we use the unknown value $\text{Val}_{\overline{\mathcal{G}}}$ to define the optimal strategy. However, it will help us in the final step of the proof (see Appendix A).

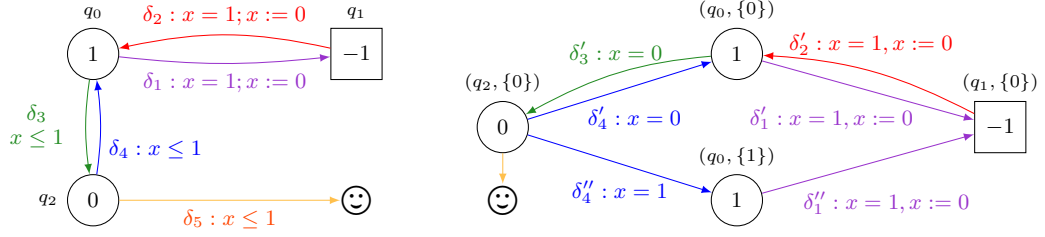
As a side note, it is tempting to strengthen Lemma 16-2 so as to ensure that all cyclic paths ending by a reset conforming to τ^* have a non-negative weight (and not only the value), i.e. an interval of weights entirely included in $[0, +\infty)$. Unfortunately, this does not hold, as shown in the following example:

► **Example 17.** We consider the closed game depicted on the right in Figure 3. Let $\pi = (q_0, \{0\}) \xrightarrow{\delta_1} (q_1, \{0\}) \xrightarrow{\delta_2} (q_0, \{0\})$ be the cyclic path for which $\text{wt}_{\Sigma}(\pi) = [-1, 1]$. To evaluate the value of π , Min and Max need to choose delays $t_1, t_2 \in [0, 1]$ when firing δ_1 and δ_2 . We obtain a set of finite plays ρ parametrised by t_1 and t_2 of weight $\text{wt}_{\Sigma}(\rho) = -t_1 + t_2$. We deduce that $\text{Val}_{\mathcal{G}}(\pi) = \inf_{t_1} \sup_{t_2} (t_2 - t_1) = 0$ (when Min and Max choose $t_1 = t_2 = 1$). The optimal strategy of Max does not use the transition δ_3 and is thus forced to play in the previous cyclic path (with a non-negative value but with a negative weight): from the configuration $((q_0, \{0\}), 0)$, Min has thus no other choice than playing transition δ_4 after a delay of 1 unit of time, leading to a value of -1 .

4 Unfolding

We now define the partial *unfolding* of the game $\overline{\mathcal{G}}$ that we need in order to compute $\text{Val}_{\overline{\mathcal{G}}}$, stopping the unfolding when too many transitions with a reset have been taken or when the play is too long since the last reset. About the transitions with a reset, when such a transition is taken for the first time, we go into anew copy of the game, from which, if this

15:12 Decidability of One-Clock Weighted Timed Games with Arbitrary Weights



■ **Figure 5** A WTG (left), and a portion of its closure (right) where δ'_2 belongs to a cyclic path of non-negative value and another cyclic path of negative value.

transition happens to be chosen one more time, we stop the game by jumping into a new target location. The final weight of this target location is determined by the value of the cyclic path (ending with a reset) that would have just been closed. If the cyclic path has a negative value, then we go in a leaf $t_{<0}$ of final weight $-\infty$ since this is a desirable cycle for Min. Otherwise, we go in a leaf $t_{\geq 0}$ of final weight big enough $|L|(W_{\text{tr}} + M W_{\text{loc}}) + W_{\text{fin}}$ (for technical reasons that will become clear later, we cannot simply put a final weight $+\infty$) so as it remains an undesirable behaviour for Min.

A single transition with a reset can be part of two distinct cyclic paths, one of negative value and the other of non-negative value, as demonstrated in the following example:

► **Example 18.** In Figure 5, we have depicted a WTG (left) and a portion of its closure (right), where δ'_2 is contained in a cyclic path of negative value

$$(q_0, \{0\}) \xrightarrow{\delta'_3} (q_2, \{0\}) \xrightarrow{\delta''_4} (q_0, \{1\}) \xrightarrow{\delta''_1} (q_1, \{0\}) \xrightarrow{\delta'_2} (q_0, \{0\})$$

and another cyclic path of non-negative (zero) value

$$(q_0, \{0\}) \xrightarrow{\delta'_1} (q_1, \{0\}) \xrightarrow{\delta'_2} (q_0, \{0\}).$$

Thus, knowing the last transition of the cycle is not enough to compute the value of the cyclic path. Instead, we need to record the whole cyclic path: copying the game (as done in the non-negative setting [10]) is not enough, our unfolding needs to remember the path followed so far. Locations of the unfolding are thus finite paths of $\bar{\mathcal{G}}$.

In order to obtain an acyclic unfolding, we will rely on a property of *reset-acyclic* WTG, i.e that do not contain cyclic paths with a transition with a reset. For such WTG, [13] shows the existence of an ε -optimal strategy for Min with a particular shape. This strategy is also called as *switching strategy* [14], as defined by the following:

► **Definition 19.** A switching strategy σ is described by two memoryless strategies σ^1 and σ^2 , as well as a switching threshold κ' . The strategy σ then consists in playing strategy σ^1 until either we reach a target location or the finite play has a length of at least κ' , in which case we switch to strategy σ^2 .

Intuitively, σ^1 aims at reaching a cyclic play with negative weight, while σ^2 is an attractor to the target. As a consequence, we can estimate the maximal number of steps needed by σ^2 to reach the target. Combining this with the switching threshold κ' we can deduce a threshold κ that upper bounds the number of steps under the switching strategy σ to reach the target. We obtain the following result with an explicit bound κ given by the previous work of [13]. From a combination of their Lemma 5 and Theorem 25, we know that the switching threshold κ' is in

$$O(|L| \times [W_{\text{loc}} + W_{\text{tr}}^4 |L|^9 \times |L| W_{\text{tr}} + W_{\text{tr}}^4 |L|^9]) = O(|L|^{11} (W_{\text{loc}} + W_{\text{tr}}^4))$$

■ **Algorithm 1** Function NEXT that maps pairs $(\pi, \delta) \in \text{FPaths}_{\overline{\mathcal{G}}} \times \overline{\Delta}$ to pairs (π', δ') composed of a finite path π' of $\overline{\mathcal{G}}$ (or $t_{\geq 0}$, or $t_{< 0}$, or $t_{+\infty}$) and a new transition δ' of the unfolding \mathcal{U} .

```

1: function NEXT( $\pi, \delta = \ell_1 \xrightarrow{I,R,w} \ell_2$ ): ▷ last( $\pi$ ) =  $\ell_1$ 
2:   if  $\ell_2 \in L_t$  then  $\pi' := \ell_2$ 
3:   else if  $R = \{x\}$  then
4:     if  $|\pi|_\delta = 0$  then  $\pi' := \pi \cdot \delta$ 
5:     else { let  $\pi = \pi_1 \cdot \delta \cdot \pi_2$ 
6:       if  $\text{Val}_{\overline{\mathcal{G}}}(\pi_2 \cdot \delta) \geq 0$  then  $\pi' := t_{\geq 0}$  else  $\pi' := t_{< 0}$  }
7:   else { let  $\pi = \pi_1 \cdot \pi_2$  where  $\pi_2$  contains no reset and  $|\pi_2|$  is maximal
8:     if  $|\pi_2| = \kappa$  then  $\pi' := t_{+\infty}$  else  $\pi' := \pi \cdot \delta$  }
9:    $\delta' := \pi \xrightarrow{I,R,w} \pi'$  ▷  $\Delta\text{proj}(\delta') := \delta$ 
10:  return  $(\pi', \delta')$ 

```

Then, we let κ'' be the number of turns taken by σ^2 to reach the target location, which is polynomial in the number of locations of the underlying region automaton, thus polynomial in the number of locations of the game (since there is only one-clock). Overall, this gives a definition for κ as

$$\kappa = \kappa' + \kappa'' = O(|L|^{12}(W_{\text{loc}} + W_{\text{tr}}^4))$$

that is polynomial in $|Q|$ (as $|L|$ is polynomial in $|Q|$) and in W .

► **Lemma 20** ([13]). *Let \mathcal{G} be a reset-acyclic WTG. Min has an ε -optimal switching strategy σ such that all plays conforming to σ reach the target within κ steps. Moreover, κ is polynomial in $|Q|$ and W .*

As a consequence, assuming that Min plays almost optimally using a switching strategy, we can bound the number of steps between two transitions with a reset by κ . This property allows us to avoid incorporating cycles in the unfolding: we cut the unfolding when the play becomes longer than κ since the last seen transition with a reset. In this case, we will jump into a new target location $t_{+\infty}$ whose final weight is equal to $+\infty$ since it is an undesirable behaviour for Min.

The scheme of the unfolding is depicted in Figure 4 when the closed game $\overline{\mathcal{G}}$ contains two transitions with a reset, δ_1 and δ_2 , each belonging to several cycles of different values (negative and non-negative). Inside each grey component, transitions with no reset are unfolded for κ steps by only keeping in the current location the path followed so far. In-between the components are transitions with a reset. The second time they are visited, the value of the cycle it closes is computed, and we jump in $t_{< 0}$ or $t_{\geq 0}$ depending on the sign of the value.

► **Definition 21.** *The unfolding of $\overline{\mathcal{G}}$ from the initial location ℓ_i is the (a priori infinite) WTG $\mathcal{U} = \langle L'_{\text{Min}}, L'_{\text{Max}}, L'_t, L'_u, \Delta', \text{wt}', \text{wt}'_t \rangle$ with $L'_{\text{Min}} \subseteq \text{FPaths}_{\text{Min}}$, $L'_{\text{Max}} \subseteq \text{FPaths}_{\text{Max}}$, $L'_t \subseteq L_t \cup \{t_{\geq 0}, t_{< 0}, t_{+\infty}\}$ such that*

- $L' = L'_{\text{Min}} \uplus L'_{\text{Max}} \uplus L'_t$ and Δ' are the smallest sets such that $\ell_i \in L'$ and for all $\pi \in L'_{\text{Min}} \uplus L'_{\text{Max}}$ and $\delta \in \Delta$, if $\text{NEXT}(\pi, \delta) = (\pi', \delta')$ then $\pi' \in L'$ and $\delta' \in \Delta'$ (where NEXT is defined in Algorithm 1);
- $L'_u = \{\pi \in L' \mid \text{last}(\pi) \in L_u\}$;
- for all $\pi \notin L'_t$, $\text{wt}'(\pi) = \overline{\text{wt}}(\text{last}(\pi))$;
- for all $\pi \in L'_t$, for all ν ,

$$\begin{aligned} \text{wt}'_t(\pi, \nu) &= \overline{\text{wt}}_t(\pi, \nu) & \text{if } \pi \in L_t & & \text{wt}'_t(t_{\geq 0}, \nu) &= |L|(W_{\text{tr}} + M W_{\text{loc}}) + W_{\text{fin}} \\ \text{wt}'_t(t_{< 0}, \nu) &= -\infty & & & \text{wt}'_t(t_{+\infty}, \nu) &= +\infty. \end{aligned}$$

15:14 Decidability of One-Clock Weighted Timed Games with Arbitrary Weights

A target location is reached when the length between two resets is too long or when a transition with a reset would appear two times. Moreover, the length of a path in the location that is not a target, given by the application of NEXT, strictly increases. This allows us to show that \mathcal{U} is a *finite* and acyclic WTG as expected.

► **Lemma 22.** *\mathcal{U} is an acyclic WTG with a finite set of locations of cardinality exponential in $|Q|$ and W_{tr} .*

Furthermore, in \mathcal{U} , as we showed in $\bar{\mathcal{G}}$, in Lemma 16, Max can play optimally with a memoryless strategy. Note that, unlike in $\bar{\mathcal{G}}$, there exist no cyclic paths in \mathcal{U} : however, we can check the positivity of the “cyclic plays” in-between two occurrences of the same transition containing a reset when we jump in $\mathbf{t}_{\geq 0}$.

► **Lemma 23.** *In \mathcal{U} , Max has a memoryless optimal strategy τ^* such that if $\rho = \rho_1 \xrightarrow{t_1, \delta'_1} \rho_2 \xrightarrow{t_2, \delta'_2} (\mathbf{t}_{\geq 0}, 0)$ is conforming to τ^* with $\Delta \text{proj}(\delta'_1) = \Delta \text{proj}(\delta'_2)$ a transition with a reset of x , then $\text{wt}_{\Sigma}(\rho_2 \xrightarrow{t_2, \delta'_2} (\mathbf{t}_{\geq 0}, 0)) \geq 0$.*

The property on the weight of plays that reach $\mathbf{t}_{\geq 0}$ is guaranteed by the structure of \mathcal{U} . Indeed, as \mathcal{U} is acyclic, we know that the value of the path followed by a play ending in $\mathbf{t}_{\geq 0}$ is non-negative. That would no longer be the case if we would have defined \mathcal{U} with grey components containing cyclic paths without reset, since the value of cyclic path do not compose, as demonstrated by the following example.

► **Example 24.** In the WTG $\bar{\mathcal{G}}$ depicted in Figure 5, we can see that $\text{Val}((q_0, \{0\}) \xrightarrow{\delta_1} (q_1, \{0\}) \xrightarrow{\delta_2} (q_0, \{0\})) = 0$: Min and Max must delay 1 in each location, and $\text{Val}^0((q_0, \{0\}) \xrightarrow{\delta_3} (q_2, \{0\}) \xrightarrow{\delta_4} (q_0, \{1\})) = 0$. However, when we composed these two cyclic path, we obtain that $\text{Val}((q_0, \{0\}) \xrightarrow{\delta_3} (q_2, \{0\}) \xrightarrow{\delta_4} (q_0, \{1\}) \xrightarrow{\delta_1} (q_1, \{0\}) \xrightarrow{\delta_2} (q_0, \{0\})) = -1$.

Now, as in Lemma 16, τ^* is defined with argmax on transitions and delays. Thus, to obtain a play ending in $\mathbf{t}_{\geq 0}$ with a non-negative weight, we constrain Max to play the value of the cycle that reached $\mathbf{t}_{\geq 0}$ by assigning it a finite final weight.

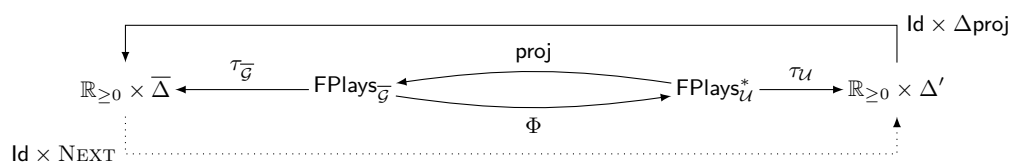
Finally, the most difficult part of the proof is to show that the unfolding preserves the value. Remember that we have fixed an initial location $\ell_i = (q_i, I_i)$ to build \mathcal{U} .

► **Theorem 25.** *For all $\nu \in I_i$, $\text{Val}_{\bar{\mathcal{G}}}(\ell_i, \nu) = \text{Val}_{\mathcal{U}}(\ell_i, \nu)$.*

Before proving Theorem 25, we show how this helps prove our main result.

Proof of Theorem 10. Remember (by Lemma 6) that we only need to explain how to compute $\nu \mapsto \text{Val}_{\bar{\mathcal{G}}}((q_i, I_i), \nu)$ over I_i . By Theorem 25, this is equivalent to computing $\nu \mapsto \text{Val}_{\mathcal{U}}((q_i, I_i), \nu)$ over I_i . We now explain why this is doable.

First, the definition of \mathcal{U} is effective: we can compute it entirely, making use of Lemma 22 showing that it is a finite WTG. The only non-trivial part is the determination of $\text{Val}_{\bar{\mathcal{G}}}(\pi_2 \cdot \delta)$ in Algorithm 1 to determine in which target location we jump. Since $\pi_2 \cdot \delta$ is a finite path, we can apply Theorem 8 to compute the value of the corresponding game, which is exactly the value $\text{Val}_{\bar{\mathcal{G}}}(\pi_2 \cdot \delta)$. The complexity of computing the value of a path is polynomial in the length of this path (that is exponential in $|Q|$ and W_{tr} , by Lemma 22) and polynomial in $|Q|$ and W (notice that weights of $\bar{\mathcal{G}}$ are the same as the ones in \mathcal{G}): this is thus of complexity exponential in $|Q|$ and W_{tr} , and polynomial in W_{loc} and W_{fin} . Since \mathcal{U} has an exponential number of locations with respect to $|Q|$ and W_{tr} , the total time required to compute \mathcal{U} is exponential with respect to $|Q|$ and W_{tr} , and polynomial with respect to W_{loc} and W_{fin} .



■ **Figure 6** Scheme showing the links between the different objects defined for the proof of Theorem 25 where $\text{FPlays}_{\mathcal{U}}^*$ is the set of finite plays of \mathcal{U} avoiding target locations $t_{\geq 0}$ and $t_{< 0}$.

Lemma 22 ensures that \mathcal{U} is acyclic, so we can apply Theorem 8 to compute the value mapping $\nu \mapsto \text{Val}_{\mathcal{U}}((q_i, I_i), \nu)$ as a piecewise affine and continuous function. It requires a complexity polynomial in the number of locations of \mathcal{U} , and in W_{loc} , W_{tr} , and W_{fin} (since weights of \mathcal{U} all come from \mathcal{G}). Knowing the previous bound on the number of locations of \mathcal{U} , this complexity translates into an exponential time complexity with respect to $|Q|$ and W_{tr} , and polynomial with respect to W_{loc} and W_{fin} . ◀

The proof of Theorem 25 splits into two inequalities. We prove in Appendix A that $\text{Val}_{\bar{\mathcal{G}}}(\ell_i, \nu) \leq \text{Val}_{\mathcal{U}}(\ell_i, \nu)$, i.e. that Max can guarantee to always do at least as good in \mathcal{U} as in $\bar{\mathcal{G}}$. We thus show that for an optimal strategy $\tau_{\bar{\mathcal{G}}}$ in $\bar{\mathcal{G}}$ (defined by Lemma 16), there exists a strategy $\tau_{\mathcal{U}}$ in \mathcal{U} such that for all plays ρ conforming to $\tau_{\mathcal{U}}$, there exists a play conforming to $\tau_{\bar{\mathcal{G}}}$ with a weight at most the weight of ρ . As it is depicted in Figure 6, the strategy $\tau_{\mathcal{U}}$ is defined via a *projection* of plays of \mathcal{U} in $\bar{\mathcal{G}}$: we use the mapping NEXT to send back transitions of $\bar{\Delta}$ to Δ' .

We then prove in Appendix B that $\text{Val}_{\bar{\mathcal{G}}}(\ell_i, \nu) \geq \text{Val}_{\mathcal{U}}(\ell_i, \nu)$, i.e. Max can guarantee to always do at least as good in $\bar{\mathcal{G}}$ as in \mathcal{U} . We thus show that for an optimal strategy $\tau_{\mathcal{U}}$ in \mathcal{U} (defined by Lemma 23), there exists a strategy $\tau_{\bar{\mathcal{G}}}$ in $\bar{\mathcal{G}}$ such that for the unique play ρ conforming to $\tau_{\bar{\mathcal{G}}}$ and the *switching* strategy (see Definition 19), there exists a play conforming to $\tau_{\mathcal{U}}$ with a weight at most the weight of ρ . As depicted in Figure 6, the strategy $\tau_{\bar{\mathcal{G}}}$ is defined via a function Φ that puts plays of $\bar{\mathcal{G}}$ in \mathcal{U} . Intuitively, this function removes all cyclic plays ending with a reset from plays in $\bar{\mathcal{G}}$.

5 Conclusion

We solve one-clock WTG with arbitrary weights, an open problem for several years. We strongly rely on the determinacy of the game, taking the point of view of Max, instead of the one of Min as was done in previous work with only non-negative weights. We also use technical ingredients such as the closure of a game, switching strategies for Min, and acyclic unfoldings. Regarding the complexity, our algorithm runs in exponential time (with weights encoded in unary), which does not match the known PSPACE lower bound with weights in binary [16]. Observe that this lower bound only uses non-negative weights. This complexity gap deserves further study. Our work also opens three research directions. First, as we unfold the game into a finite tree, it would be interesting to develop a symbolic approach that shares computation between subtrees in order to obtain a more efficient algorithm. Second, playing stochastically in WTG with shortest path objectives has been recently studied in [20]. One could study an extension of one-clock WTG with stochastic transitions. In this context, Min aims at minimizing the expectation of the accumulated weight. Third, the analysis of cycles that we have done in the setting of one-clock WTG can be an inspiration to identify new decidable classes of WTG with arbitrarily many clocks.

References

- 1 Rajeev Alur, Mikhail Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, volume 3142 of *LNCS*, pages 122–133. Springer, 2004. doi:10.1007/978-3-540-27836-8_13.
- 2 Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994. doi:10.1016/0304-3975(94)90010-8.
- 3 Eugene Asarin and Oded Maler. As soon as possible: Time optimal control for timed automata. In *Hybrid Systems: Computation and Control*, volume 1569 of *LNCS*, pages 19–30. Springer, 1999. doi:10.1007/3-540-48983-5_6.
- 4 Gerd Behrmann, Agnès Cougnard, Alexandre David, Emmanuel Fleury, Kim Guldstrand Larsen, and Didier Lime. Uppaal-tiga: Time for playing games! In Werner Damm and Holger Hermanns, editors, *Proceedings of the 19th International Conference on Computer Aided Verification (CAV 2007)*, volume 4590 of *LNCS*, pages 121–125. Springer, 2007. doi:10.1007/978-3-540-73368-3_14.
- 5 Gerd Behrmann, Ansgar Fehnker, Thomas Hune, Kim Larsen, Paul Pettersson, Judi Romijn, and Frits Vaandrager. Minimum-cost reachability for priced time automata. In *International Workshop on Hybrid Systems: Computation and Control*, pages 147–161. Springer, 2001. doi:10.1007/3-540-45351-2_15.
- 6 Béatrice Bérard, Antoine Petit, Volker Diekert, and Paul Gastin. Characterization of the expressive power of silent transitions in timed automata. *Fundamenta Informaticae*, 36(2-3):145–182, 1998. doi:10.3233/FI-1998-36233.
- 7 Patricia Bouyer. Erratum to the FSTTCS'04 paper “optimal strategies in priced timed game automata”. Personal Communication, 2016.
- 8 Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. Optimal strategies in priced timed game automata. In Kamal Lodaya and Meena Mahajan, editors, *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science*, pages 148–160, Berlin, Heidelberg, 2005. Springer. doi:10.1007/978-3-540-30538-5_13.
- 9 Patricia Bouyer, Samy Jaziri, and Nicolas Markey. On the value problem in weighted timed games. In *Proceedings of the 26th International Conference on Concurrency Theory (CONCUR'15)*, volume 42 of *Leibniz International Proceedings in Informatics*, pages 311–324. Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CONCUR.2015.311.
- 10 Patricia Bouyer, Kim G. Larsen, Nicolas Markey, and Jacob Illum Rasmussen. Almost optimal strategies in one clock priced timed games. In S. Arun-Kumar and Naveen Garg, editors, *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science*, pages 345–356, Berlin, Heidelberg, 2006. Springer. doi:10.1007/11944836_32.
- 11 Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On optimal timed strategies. In Paul Pettersson and Wang Yi, editors, *Formal Modeling and Analysis of Timed Systems*, pages 49–64, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:10.1007/11603009_5.
- 12 Thomas Brihaye, Gilles Geeraerts, Axel Haddad, Engel Lefauchaux, and Benjamin Monmege. Simple priced timed games are not that simple. In *Proceedings of the 35th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'15)*, volume 45 of *LIPIcs*, pages 278–292. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.FSTTCS.2015.278.
- 13 Thomas Brihaye, Gilles Geeraerts, Axel Haddad, Engel Lefauchaux, and Benjamin Monmege. One-clock priced timed games with negative weights. Research Report 2009.03074, arXiv, 2021.
- 14 Thomas Brihaye, Gilles Geeraerts, Axel Haddad, and Benjamin Monmege. Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games. *Acta Informatica*, 54(1):85–125, February 2017. doi:10.1007/s00236-016-0276-z.

- 15 Damien Busatto-Gaston, Benjamin Monmege, and Pierre-Alain Reynier. Optimal reachability in divergent weighted timed games. In *Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2017)*, LNCS, pages 162–178. Springer, 2017. doi:10.1007/978-3-662-54458-7_10.
- 16 John Fearnley, Rasmus Ibsen-Jensen, and Rahul Savani. One-clock priced timed games are PSPACE-hard. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Sciences (LICS'20)*, pages 397–409. ACM, 2020. doi:10.1145/3373718.3394772.
- 17 Thomas Dueholm Hansen, Rasmus Ibsen-Jensen, and Peter Bro Miltersen. A faster algorithm for solving one-clock priced timed games. In Pedro R. D'Argenio and Hernán C. Melgratti, editors, *Proceedings of the 24th International Conference on Concurrency Theory (CONCUR'13)*, volume 8052 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2013. doi:10.1007/978-3-642-40184-8_37.
- 18 Marcin Jurdziński and Ashutosh Trivedi. Reachability-time games on timed automata. In *Proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP'07)*, volume 4596 of *LNCS*, pages 838–849. Springer, 2007.
- 19 François Laroussinie, Nicolas Markey, and Philippe Schnoebelen. Model checking timed automata with one or two clocks. In *Proceedings of CONCUR'04*, pages 387–401, 2004. doi:10.1007/978-3-540-28644-8_25.
- 20 Benjamin Monmege, Julie Parreaux, and Pierre-Alain Reynier. Playing Stochastically in Weighted Timed Games to Emulate Memory. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *LIPICs*, pages 137:1–137:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICs.ICALP.2021.137.
- 21 Benjamin Monmege, Julie Parreaux, and Pierre-Alain Reynier. Decidability of one-clock weighted timed games with arbitrary weights. Research Report 2207.01608, arXiv, 2022. arXiv:2207.01608.
- 22 Michał Rutkowski. Two-player reachability-price games on single-clock timed automata. In *Proceedings of the Ninth Workshop on Quantitative Aspects of Programming Languages (QAPL'11)*, volume 57 of *EPTCS*, pages 31–46, 2011.

A Proof of $\text{Val}_{\bar{\mathcal{G}}}(\ell_i, \nu) \leq \text{Val}_{\mathcal{U}}(\ell_i, \nu)$

We show this first inequality by rewriting it $\text{Val}_{\bar{\mathcal{G}}}(\ell_i, \nu) \leq \sup_{\tau_{\mathcal{U}}} \text{Val}_{\mathcal{U}}^{\tau_{\mathcal{U}}}(\ell_i, \nu)$. Let $\tau_{\bar{\mathcal{G}}}$ be a memoryless optimal strategy of Max in $\bar{\mathcal{G}}$ satisfying the conditions of Lemma 16: in particular, $\text{Val}_{\bar{\mathcal{G}}}(\ell_i, \nu) = \text{Val}_{\bar{\mathcal{G}}}^{\tau_{\bar{\mathcal{G}}}}(\ell_i, \nu)$. To conclude, it is thus sufficient to build from $\tau_{\bar{\mathcal{G}}}$ a strategy $\tau_{\mathcal{U}}$ in \mathcal{U} such that

► **Proposition 26.** $\text{Val}_{\bar{\mathcal{G}}}^{\tau_{\bar{\mathcal{G}}}}(\ell_i, \nu) \leq \text{Val}_{\mathcal{U}}^{\tau_{\mathcal{U}}}(\ell_i, \nu)$

Following Figure 6, we use a projection operator to do so. It projects finite plays of \mathcal{U} starting in ℓ_i (since these are the only ones we need to take care of) to finite plays of $\bar{\mathcal{G}}$. For this reason, from now on, $\text{FPlays}_{\mathcal{U}}$ and $\text{FPlays}_{\bar{\mathcal{G}}}$ denote the subsets of plays that start in location ℓ_i . Moreover, we limit ourselves to projecting plays of \mathcal{U} that do not reach the targets $t_{<0}$ and $t_{\geq 0}$, since otherwise there is no canonical projection in $\bar{\mathcal{G}}$. Formally, we thus let $\text{FPlays}_{\mathcal{U}}^*$ be all such finite plays of $\text{FPlays}_{\mathcal{U}}$ that do not end in $t_{<0}$ or $t_{\geq 0}$. The projection function $\text{proj}: \text{FPlays}_{\mathcal{U}}^* \rightarrow \text{FPlays}_{\bar{\mathcal{G}}}$ is defined inductively on finite plays $\rho \in \text{FPlays}_{\mathcal{U}}^*$ by letting $\text{proj}(\rho)$ be

$$\left\{ \begin{array}{ll} (\ell_i, \nu) & \text{if } \rho = (\ell_i, \nu) \in L' \\ \text{proj}(\rho') \xrightarrow{t, \Delta \text{proj}(\delta')} (\text{last}(\pi), \nu) & \text{if } \rho = \rho' \xrightarrow{t, \delta'} (\pi, \nu) \\ \text{proj}(\rho') \xrightarrow{t, \Delta \text{proj}(\delta')} (\ell', \nu) & \text{if } \rho = \rho' \xrightarrow{t, \delta'} (t_{+\infty}, \nu) \text{ and } \Delta \text{proj}(\delta') = \ell \xrightarrow{I, R} \ell' \end{array} \right.$$

where $\Delta \text{proj}(\delta')$ is defined on line 9 of the NEXT function (see Algorithm 1). It fulfils the following properties:

15:18 Decidability of One-Clock Weighted Timed Games with Arbitrary Weights

► **Lemma 27.** For all plays $\rho \in \text{FPlays}_{\mathcal{U}}^*$ such that ρ does not end in $t_{+\infty}$,

1. if $\text{last}(\rho) = (\pi, \nu)$ then $\text{last}(\text{proj}(\rho)) = (\text{last}(\pi), \nu)$;
2. $\text{wt}_{\Sigma}(\rho) = \text{wt}_{\Sigma}(\text{proj}(\rho))$;
3. if $\text{last}(\rho) = (\pi, \nu)$ with $\pi \notin L_t$, then $\text{proj}(\rho)$ follows π .

Proof.

1. This is direct from a case analysis on the definition of proj .
2. We reason by induction on the length of $\rho \in \text{FPlays}_{\mathcal{U}}^*$. If $\rho = (\ell_i, \nu)$, then we have $\text{proj}(\rho) = \rho$, so $\text{wt}_{\Sigma}(\rho) = 0 = \text{wt}_{\Sigma}(\text{proj}(\rho))$. Now, we suppose that $\rho = \rho' \xrightarrow{t, \delta'} (\pi, \nu)$, with ρ' ending in location π' . Then

$$\text{wt}_{\Sigma}(\rho) = \text{wt}_{\Sigma}(\rho') + t \text{wt}'(\pi') + \text{wt}'(\delta')$$

This is equal to

$$\text{wt}_{\Sigma}(\rho') + t \overline{\text{wt}}(\text{last}(\pi')) + \overline{\text{wt}}(\Delta \text{proj}(\delta'))$$

since $\text{wt}'(\pi') = \overline{\text{wt}}(\text{last}(\pi'))$, and $\text{wt}'(\delta') = \overline{\text{wt}}(\Delta \text{proj}(\delta'))$ by definition of \mathcal{U} . By induction hypothesis, this implies that $\text{wt}_{\Sigma}(\rho)$ is equal to

$$\text{wt}_{\Sigma}(\text{proj}(\rho')) + t \overline{\text{wt}}(\text{last}(\pi')) + \overline{\text{wt}}(\Delta \text{proj}(\delta'))$$

By the first item and by definition of $\text{proj}(\rho)$, we conclude that $\text{wt}_{\Sigma}(\rho) = \text{wt}_{\Sigma}(\text{proj}(\rho))$.

3. We reason by induction on the length of ρ . If $\rho = (\ell_i, \nu)$, the property is trivial. Now, we suppose that $\rho' = \rho \xrightarrow{t, \delta'} (\pi', \nu')$. We have $\text{proj}(\rho') = \text{proj}(\rho) \xrightarrow{t, \delta} (\text{last}(\pi'), \nu')$ with $\delta = \Delta \text{proj}(\delta')$. Since ρ is a prefix of $\rho' \in \text{FPlays}_{\mathcal{U}}^*$, ρ belongs to $\text{FPlays}_{\mathcal{U}}^*$ too and does not end in L_t . Thus, letting $\text{last}(\rho) = (\pi, \nu)$, by induction hypothesis, $\text{proj}(\rho)$ follows π . Moreover, we have $\text{NEXT}(\pi, \delta) = (\pi', \delta')$. By definition of NEXT , the value of π' must be obtained from π on lines 4, or 8, and thus $\pi' = \pi \cdot \delta$. In particular, we can deduce that $\text{proj}(\rho')$ follows π . ◀

Then, for all plays $\rho \in \text{FPlays}_{\mathcal{U}}^*$ (for plays not starting in ℓ_i or plays ending in the target, the decision of ρ is irrelevant) such that $\text{last}(\rho) = (\pi, \nu)$ and $\pi \in L'_{\text{Max}}$, we define

$$\tau_{\mathcal{U}}(\rho) = (t, \delta') \quad \text{if } \tau_{\overline{\mathcal{G}}}(\text{proj}(\rho)) = (t, \delta) \text{ and } \text{NEXT}(\pi, \delta) = (\pi', \delta') \quad (2)$$

This is a valid decision for Max . First, by Lemma 27-1, we have $\text{last}(\text{proj}(\rho)) = (\text{last}(\pi), \nu)$. Moreover, delays chosen in $\tau_{\overline{\mathcal{G}}}$ and $\tau_{\mathcal{U}}$ are the same, and the guards of δ and δ' are identical. Thus, whether or not the location π is urgent (i.e. $\text{last}(\pi)$ is urgent), the decision (t, δ') gives rise to an edge in $\llbracket \mathcal{U} \rrbracket$.

Since the definition of $\tau_{\mathcal{U}}$ relies on the projection, it is of no surprise that:

► **Lemma 28.** Let $\rho \in \text{FPlays}_{\mathcal{U}}^*$ be a play conforming to $\tau_{\mathcal{U}}$. Then $\text{proj}(\rho)$ is conforming to $\tau_{\overline{\mathcal{G}}}$.

Proof. We reason by induction on the length of ρ . If $\rho = (\ell_i, \nu)$, then $\text{proj}(\rho) = (\ell_i, \nu)$ and the property is trivial. Otherwise, let $\rho = \rho^1 \xrightarrow{t, \delta'} (\pi, \nu)$. Then, $\text{proj}(\rho) = \text{proj}(\rho^1) \xrightarrow{t, \delta} (\text{last}(\pi), \nu)$ where $\delta = \Delta \text{proj}(\delta')$. By induction hypothesis, $\text{proj}(\rho^1)$ is conforming to $\tau_{\overline{\mathcal{G}}}$. Let $\text{last}(\text{proj}(\rho^1)) = (\ell^1, \nu^1)$. If $\ell^1 \in L_{\text{Min}}$, we directly conclude that $\text{proj}(\rho)$ is conforming to $\tau_{\overline{\mathcal{G}}}$ too. Otherwise, and since ρ is conforming to $\tau_{\mathcal{U}}$ and the last location of ρ^1 also belongs to Max (by Lemma 27-1), we have $\tau_{\mathcal{U}}(\rho^1) = (t, \delta')$. In particular, by definition of $\tau_{\mathcal{U}}$ (see (2)), $\tau_{\overline{\mathcal{G}}}(\text{proj}(\rho^1)) = (t, \Delta \text{proj}(\delta')) = (t, \delta)$. Thus $\rho_{\overline{\mathcal{G}}}$ is conforming to $\tau_{\overline{\mathcal{G}}}$. ◀

Now, we prove Proposition 26. To do so, we show that for all plays ρ_U from (ℓ, ν) conforming to τ_U , there exists a play $\rho_{\bar{G}}$ from (ℓ, ν) conforming to $\tau_{\bar{G}}$ such that $P(\rho_{\bar{G}}) \leq P(\rho_U)$. We cannot directly use the projection operator, since some plays ρ_U may end up in $t_{<0}$ or $t_{\geq 0}$. We treat the ones ending in $t_{\geq 0}$ by making use of the final weight function we have chosen for $t_{\geq 0}$ (bigger than any acyclic play of \bar{G}). We show that there cannot be such plays ρ_U ending in $t_{<0}$, since they would contradict Lemma 16-2.

Proof of Proposition 26. Let ρ_U be a play conforming to τ_U . If ρ_U does not reach a target location of \mathcal{U} or reaches target $t_{+\infty}$, then $P(\rho_U) = +\infty$, and for all plays $\rho_{\bar{G}}$ conforming to $\tau_{\bar{G}}$, we have $P(\rho_{\bar{G}}) \leq +\infty = P(\rho_U)$. Now, suppose that ρ_U reaches a target location different from $t_{+\infty}$.

- If the target location reached by ρ_U is not in $\{t_{\geq 0}, t_{<0}\}$, then $\rho_U \in \text{FPlays}_{\mathcal{U}}^*$ and we can thus let $\rho_{\bar{G}} = \text{proj}(\rho_U)$. Lemma 28 ensures that $\rho_{\bar{G}}$ is conforming to $\tau_{\bar{G}}$. Moreover, Lemma 27-1 ensures that if $\text{last}(\rho_U) = (\pi, \nu)$ then $\text{last}(\rho_{\bar{G}}) = (\text{last}(\pi), \nu)$ so that $\text{wt}'_t(\pi, \nu) = \overline{\text{wt}}'_t(\text{last}(\pi), \nu)$. Since proj preserves the weight (see Lemma 27-2), we obtain $P(\rho_{\bar{G}}) = P(\rho_U)$.
- If the target location reached by ρ_U is $t_{\geq 0}$, then we decompose ρ_U as $\rho_U = \rho_U^1 \xrightarrow{t, \delta'} (t_{\geq 0}, \nu)$ with $\rho_U^1 \in \text{FPlays}_{\mathcal{U}}^*$. Let $\rho_{\bar{G}} = \rho_{\bar{G}}^1 \rho_{\bar{G}}^2$ be a play such that $\rho_{\bar{G}}^1 = \text{proj}(\rho_U^1) \xrightarrow{t, \delta} (\ell, \nu)$ with $\delta = \Delta \text{proj}(\delta')$, and $\rho_{\bar{G}}^2$ be the play from (ℓ, ν) conforming to $\tau_{\bar{G}}$ and an attractor of Min to L_t . We note that $\rho_{\bar{G}}^2$ exists since the value in \mathcal{G} is supposed finite, thus Min can always guarantee to reach the target, moreover in at most $|L|$ steps (since regions are already encoded in this game). Letting $(\pi', \nu') = \text{last}(\rho_U^1)$, Lemma 27-1 ensures that $(\text{last}(\pi'), \nu') = \text{last}(\text{proj}(\rho_U^1))$. If $\pi' \in L'_{\text{Max}}$, since $\tau_U(\rho_U^1) = (t, \delta')$ and $\text{NEXT}(\pi', \delta) = (t_{<0}, \delta')$, by construction of τ_U , this implies that $\tau_{\bar{G}}(\text{proj}(\rho_U^1)) = (t, \delta)$. The last move of ρ_U^1 is thus conforming to $\tau_{\bar{G}}$. By Lemma 28 and the choice of $\rho_{\bar{G}}^2$, $\rho_{\bar{G}}$ is thus entirely conforming to $\tau_{\bar{G}}$. Moreover, $\overline{\text{wt}}(\text{last}(\pi')) = \text{wt}'(\pi')$ by definition of the unfolding. Thus, also using Lemma 27-2, we obtain

$$\begin{aligned} \text{wt}_{\Sigma}(\rho_{\bar{G}}^1) &= \text{wt}_{\Sigma}(\text{proj}(\rho_U^1)) + t \overline{\text{wt}}(\text{last}(\pi')) + \overline{\text{wt}}(\delta) \\ &= \text{wt}_{\Sigma}(\rho_U^1) + t \text{wt}'(\pi') + \text{wt}'(\delta') \\ &= \text{wt}_{\Sigma}(\rho_U) \end{aligned}$$

Moreover, as $\rho_{\bar{G}}^1$ is conforming to an attractor, its length is bounded by $|L|$. Each of its edges has a weight bounded in absolute values by $W_{\text{tr}} + M W_{\text{loc}}$. By adding its final weight, we obtain

$$P(\rho_{\bar{G}}^2) \leq |L|(W_{\text{tr}} + M W_{\text{loc}}) + W_{\text{fin}}$$

To conclude, we remark that ρ_U reaches $t_{\geq 0}$, and its weight is thus

$$P(\rho_U) = \text{wt}_{\Sigma}(\rho_U) + |L|(W_{\text{tr}} + M W_{\text{loc}}) + W_{\text{fin}}$$

Therefore $P(\rho_{\bar{G}}) = \text{wt}_{\Sigma}(\rho_{\bar{G}}^1) + P(\rho_{\bar{G}}^2) \leq P(\rho_U)$.

- If the target location reached by ρ_U is $t_{<0}$, as before we decompose ρ_U as $\rho_U = \rho_U^1 \xrightarrow{t, \delta'} (t_{\geq 0}, \nu)$ with $\rho_U^1 \in \text{FPlays}_{\mathcal{U}}^*$. Let $\rho_{\bar{G}}^1 = \text{proj}(\rho_U^1) \xrightarrow{t, \delta} (\ell, \nu)$ with $\delta = \Delta \text{proj}(\delta')$. As in the previous case, $\rho_{\bar{G}}^1$ is conforming to $\tau_{\bar{G}}$. By definition of \mathcal{U} , letting π the last location of ρ_U^1 (not in L'_t), we have $\text{NEXT}(\pi, \delta) = (t_{<0}, \delta')$ with $|\pi|_{\delta} > 0$: by letting $\pi = \pi_1 \delta \pi_2$ with $|\pi_2|_{\delta} = 0$, we have $\text{Val}_{\bar{G}}(\pi_2 \delta) < 0$. By Lemma 27-3, we know that $\text{proj}(\rho_U^1)$ follows π . Thus, $\rho_{\bar{G}}^1$ follows $\pi \delta$, and as a consequence, finishes by a play that follows the cyclic path $\pi_2 \delta$ of negative value. Since it is conforming to $\tau_{\bar{G}}$, it contradicts Lemma 16-2.

15:20 Decidability of One-Clock Weighted Timed Games with Arbitrary Weights

To conclude, we have shown that for all plays $\rho_{\mathcal{U}}$ from (ℓ_i, ν) conforming to $\tau_{\mathcal{U}}$, we can build a play $\rho_{\overline{\mathcal{G}}}$ from (ℓ_i, ν) conforming to $\tau_{\overline{\mathcal{G}}}$ such that $P(\rho_{\overline{\mathcal{G}}}) \leq P(\rho_{\mathcal{U}})$. In particular,

$$\text{Val}_{\overline{\mathcal{G}}}^{\tau_{\overline{\mathcal{G}}}}(\ell_i, \nu) = \inf_{\tau_{\overline{\mathcal{G}}} \in \text{Strat}_{\text{Min}, \overline{\mathcal{G}}}} P(\text{Play}((\ell_i, \nu), \sigma_{\overline{\mathcal{G}}}, \tau_{\overline{\mathcal{G}}})) \leq \inf_{\tau_{\mathcal{U}} \in \text{Strat}_{\text{Min}, \mathcal{U}}} P(\text{Play}((\ell_i, \nu), \sigma_{\mathcal{U}}, \tau_{\mathcal{U}})) = \text{Val}_{\mathcal{U}}^{\tau_{\mathcal{U}}}(\ell_i, \nu)$$

◀

B Proof of $\text{Val}_{\overline{\mathcal{G}}}(\ell_i, \nu) \geq \text{Val}_{\mathcal{U}}(\ell_i, \nu)$

We show this second inequality slightly differently. First we rewrite it: $\text{Val}_{\overline{\mathcal{G}}}(\ell_i, \nu) \geq \sup_{\tau_{\mathcal{U}}} \text{Val}_{\mathcal{U}}^{\tau_{\mathcal{U}}}(\ell_i, \nu)$. Considering for $\tau_{\mathcal{U}}$ the memoryless optimal strategy of Max in \mathcal{U} satisfying the conditions of Lemma 23, we therefore show that

► **Proposition 29.** $\text{Val}_{\overline{\mathcal{G}}}(\ell_i, \nu) \geq \text{Val}_{\mathcal{U}}^{\tau_{\mathcal{U}}}(\ell_i, \nu)$

To do so, following Figure 6, we first define the function Φ , mapping plays of $\overline{\mathcal{G}}$ in plays of \mathcal{U} . It needs to take care of the appearance of more than one occurrence of a transition with a reset in plays of $\overline{\mathcal{G}}$. Formally, it is defined by induction on the length of the plays by letting $\Phi(\ell_i, \nu) = (\ell_i, \nu)$, and for all plays $\rho \in \text{FPlays}_{\overline{\mathcal{G}}}$, letting $\rho' = \rho \xrightarrow{t, \delta} (\ell, \nu)$,

1. if $\Phi(\rho)$ ends in $t_{+\infty}$, we let $\Phi(\rho') = \Phi(\rho)$;
2. else if δ contains a reset and $\Phi(\rho) = \rho_1 \xrightarrow{t', \delta'} \rho_2$ with $\Delta \text{proj}(\delta') = \delta$, letting π the first location of ρ_2 , we let $\Phi(\rho') = \rho_1 \xrightarrow{t', \delta'} (\pi, 0)$;
3. otherwise, $\Phi(\rho') = \Phi(\rho) \xrightarrow{t, \delta'} (\pi', \nu)$ if $\text{NEXT}(\pi, \delta) = (\pi', \delta')$ with π the last location of $\Phi(\rho)$.

This function satisfies the following properties:

► **Lemma 30.** For all plays $\rho \in \text{FPlays}_{\overline{\mathcal{G}}}$, if $\text{last}(\Phi(\rho)) = (\pi, \nu)$ with $\pi \notin t_{+\infty}$, then we have $\pi \notin \{t_{<0}, t_{\geq 0}\}$ and

$$\text{last}(\rho) = \begin{cases} (\text{last}(\pi), \nu) & \text{if } \pi \notin L_t \\ (\pi, \nu) & \text{otherwise} \end{cases}$$

Proof. We show the property by induction on the length of ρ . If $\rho = (\ell_i, \nu)$, then $\Phi(\rho) = \rho$ and the property holds. Otherwise, we let $\rho' = \rho \xrightarrow{t, \delta} (\ell, \nu)$, suppose that the property holds for ρ (that does not end in L'_t) and follow the definition of Φ .

1. If $\Phi(\rho)$ ends in $t_{+\infty}$, we have $\Phi(\rho') = \Phi(\rho)$ and this case is thus not possible (since $\Phi(\rho')$ is supposed to not end in $t_{+\infty}$).
2. Else if δ contains a reset and $\Phi(\rho) = \rho_1 \xrightarrow{t', \delta'} \rho_2$ with $\Delta \text{proj}(\delta') = \delta$, letting π' the first location of ρ_2 , we have $\Phi(\rho') = \rho_1 \xrightarrow{t', \delta'} (\pi', 0)$. Letting π_1 the last location of ρ_1 , we have $\text{NEXT}(\pi_1, \delta) = (\pi', \delta')$. If δ goes to location $\ell \in L_t$, then $\pi' = \ell \in L_t$, so that $\text{last}(\rho') = (\ell, 0) = (\text{last}(\Phi(\rho')), 0)$ as expected. Since ρ_1 does not contain a transition δ'_1 such that $\Delta \text{proj}(\delta'_1) = \delta$ (otherwise, in $\Phi(\rho)$, we would have already fired twice the transition δ with a reset, before trying to fire it a third time), we have $\text{last}(\Phi(\rho)) = (\pi, 0)$ with $\pi \notin \{t_{<0}, t_{\geq 0}\}$. Thus $\pi = \pi' \cdot \delta$ (and thus $\pi \notin \{t_{<0}, t_{\geq 0}\}$) so that $\text{last}(\pi) = \ell$, and we conclude.
3. Otherwise $\Phi(\rho') = \Phi(\rho) \xrightarrow{t, \delta'} (\pi', \nu)$ if $\text{NEXT}(\pi, \delta) = (\pi', \delta')$ with π the last location of $\Phi(\rho)$. Once again, we are in a case where $\pi' = \pi \cdot \delta$ which allows us to conclude as before. ◀

Then, we define $\tau_{\bar{\mathcal{G}}}$ such that its behaviour is the same as the one given by $\tau_{\mathcal{U}}$ after the application of Φ on the finite play, i.e. after the removal of all cyclic paths ending by a transition with a reset. Formally, for all plays $\rho \in \text{FPlays}_{\bar{\mathcal{G}}}$, we let $\tau_{\bar{\mathcal{G}}}(\rho)$ be defined as any valid move (t, δ) if $\Phi(\rho)$ ends in $\mathfrak{t}_{+\infty}$, and otherwise

$$\tau_{\bar{\mathcal{G}}}(\rho) = (t, \Delta\text{proj}(\delta')) \quad \text{if } \tau_{\mathcal{U}}(\Phi(\rho)) = (t, \delta') \quad (3)$$

This is a valid decision for **Max**. First, by Lemma 30, $\text{last}(\rho) = (\text{last}(\pi), \nu)$ when $\text{last}(\Phi(\rho)) = (\pi, \nu)$. Moreover, delays chosen in $\tau_{\bar{\mathcal{G}}}$ and $\tau_{\mathcal{U}}$ are the same, and the guards of δ' and $\Delta\text{proj}(\delta')$ are identical. Thus, whether or not the location π is urgent, the decision $(t, \Delta\text{proj}(\delta'))$ gives rise to an edge in $\llbracket \bar{\mathcal{G}} \rrbracket$.

Since the definition of $\tau_{\bar{\mathcal{G}}}$ relies on the operation Φ , it is again not surprising that:

► **Lemma 31.** *Let $\rho \in \text{FPlays}_{\bar{\mathcal{G}}}$ be a play conforming to $\tau_{\bar{\mathcal{G}}}$. Then $\Phi(\rho)$ is conforming to $\tau_{\mathcal{U}}$.*

Proof. We reason by induction on the length of ρ . If $\rho = (\ell_i, \nu)$, then $\Phi(\rho) = (\ell_i, \nu)$ and the property is trivial. Otherwise, we suppose that $\rho' = \rho \xrightarrow{t, \delta} (\ell, \nu)$. By induction hypothesis, $\Phi(\rho)$ conforms to $\tau_{\mathcal{U}}$.

1. If $\Phi(\rho)$ ends in $\mathfrak{t}_{+\infty}$, we have $\Phi(\rho') = \Phi(\rho)$ that conforms to $\tau_{\mathcal{U}}$.
2. If δ contains a reset and $\Phi(\rho) = \rho_1 \xrightarrow{t', \delta'} \rho_2$ with $\Delta\text{proj}(\delta') = \delta$, letting π the first location of ρ_2 , we have $\Phi(\rho') = \rho_1 \xrightarrow{t', \delta'} (\pi, 0)$. This is a prefix of $\Phi(\rho)$ that conforms to $\tau_{\mathcal{U}}$, so $\Phi(\rho')$ conforms to $\tau_{\mathcal{U}}$ too.
3. Otherwise, $\Phi(\rho') = \Phi(\rho) \xrightarrow{t, \delta'} (\pi', \nu)$ if $\text{NEXT}(\pi, \delta) = (\pi', \delta')$ with π the last location of $\Phi(\rho)$. If $\Phi(\rho)$ ends in a location of **Min**, since it is conforming to $\tau_{\mathcal{U}}$, so does $\Phi(\rho')$. Otherwise, $\tau_{\bar{\mathcal{G}}}(\rho) = (t, \delta)$ which implies that $\tau_{\mathcal{U}}(\Phi(\rho)) = (t, \delta')$ with $\Delta\text{proj}(\delta'') = \delta$, meaning that $\text{NEXT}(\pi, \delta) = (\pi', \delta'')$, i.e. $\delta'' = \delta'$: in this case too, $\Phi(\rho')$ is conforming to $\tau_{\mathcal{U}}$. ◀

Now, we prove Proposition 29. Notice that contrary to Proposition 26, we do not aim at comparing $\text{Val}_{\mathcal{U}}^{\tau_{\mathcal{U}}}(\ell_i, \nu)$ with $\text{Val}_{\bar{\mathcal{G}}}^{\tau_{\bar{\mathcal{G}}}}(\ell_i, \nu)$ but instead directly with $\text{Val}_{\bar{\mathcal{G}}}(\ell_i, \nu)$. This is helpful here, since we do not need to start with any play ρ conforming to $\tau_{\bar{\mathcal{G}}}$. Instead, we pick a special play, choosing well the strategy followed by **Min**. Indeed, let **Min** follow an ε -optimal (switching) strategy σ in $\bar{\mathcal{G}}$, as given in [12, 13]. As we explained before Definition 21, in WTG without resets, this ensures that in all plays $\rho_{\bar{\mathcal{G}}}$ conforming to σ , the target is reached fast enough (with a number of transitions bounded by κ). We can easily enrich the result of [12, 13] to take into account resets. Indeed, as performed in [12, Theorem 10], to show that all one-clock WTG have a (a priori non computable) value function that is piecewise affine with a finite number of cutpoints, we can replace each transition with a reset with a new transition jumping in a fresh target location of value given by the value function we aim at computing. From a strategy perspective, this means that in each component of our unfolding (in-between two transitions with a reset), **Min** follows a switching strategy. Notice that such strategies are a priori not knowing to be computable (since we cannot perform the transformation described above, using the value function), but we use only its existence in this proof.

We finally obtain an ε -optimal strategy σ for **Min** in $\bar{\mathcal{G}}$ such that that in all plays $\rho_{\bar{\mathcal{G}}}$ conforming to σ , in-between two transitions with a reset and after the last such transition, the number of transitions is bounded by κ .

Proof of Proposition 29. We now consider the special play ρ from (ℓ_i, ν) conforming to σ and $\tau_{\bar{\mathcal{G}}}$. It reaches a target, since σ is ε -optimal and $\text{Val}_{\bar{\mathcal{G}}}(\ell_i, \nu) \neq +\infty$. We show that

$$\exists \rho_{\mathcal{U}} \in \text{FPlays}_{\mathcal{U}} \text{ conforming to } \tau_{\mathcal{U}} \quad \text{P}(\rho_{\mathcal{U}}) \leq \text{P}(\rho) \quad (\star)$$

15:22 Decidability of One-Clock Weighted Timed Games with Arbitrary Weights

As a consequence, we obtain

$$\text{Val}_{\mathcal{U}}^{\tau_{\mathcal{U}}}(\ell_i, \nu) = \inf_{\sigma_{\mathcal{U}} \in \text{Strat}_{\text{Min}, \mathcal{U}}} \text{P}(\text{Play}((\ell_i, \nu), \sigma_{\mathcal{U}}, \tau_{\mathcal{U}})) \leq \text{P}(\rho_{\mathcal{U}}) \leq \text{P}(\rho) \leq \text{Val}_{\overline{\mathcal{G}}}(\ell_i, \nu) + \varepsilon$$

Since this holds for all $\varepsilon > 0$, we have $\text{Val}_{\mathcal{U}}^{\tau_{\mathcal{U}}}(\ell_i, \nu) \leq \text{Val}_{\overline{\mathcal{G}}}(\ell_i, \nu)$ as expected.

To show (\star) , we proceed by induction on the prefixes ρ' of ρ , proving that (\star) holds, or that $\Phi(\rho')$ does not end in $\mathbf{t}_{+\infty}$ and $\text{wt}_{\Sigma}(\Phi(\rho')) \leq \text{wt}_{\Sigma}(\rho')$. At the end of the induction, we therefore obtain (\star) or that $\Phi(\rho)$ does not end in $\mathbf{t}_{+\infty}$ and $\text{wt}_{\Sigma}(\Phi(\rho)) \leq \text{wt}_{\Sigma}(\rho)$. We let $\text{last}(\Phi(\rho)) = (\pi, \nu)$. By Lemma 30, if $\pi \notin L_t$, then $\text{last}(\rho) = (\text{last}(\pi), \nu)$, with $\text{last}(\pi) \notin L_t$: this contradicts the fact that ρ reaches the target. Thus, $\pi \in L_t$, and $\text{last}(\rho) = (\pi, \nu)$. Therefore, $\text{P}(\Phi(\rho)) = \text{wt}_{\Sigma}(\Phi(\rho)) + \text{wt}'_t(\pi, \nu) \leq \text{wt}_{\Sigma}(\rho) + \text{wt}'_t(\pi, \nu) = \text{P}(\rho)$. Since $\Phi(\rho)$ conforms to $\tau_{\mathcal{U}}$, we obtain (\star) here too.

For $\rho' = (\ell_i, \nu)$, $\text{wt}_{\Sigma}(\Phi(\rho')) = 0 = \text{wt}_{\Sigma}(\rho')$. Suppose then that $\rho' = \rho'' \xrightarrow{t, \delta} (\ell, \nu)$. By induction on ρ'' , if (\star) does not (already) hold, we know that $\Phi(\rho'')$ does not end in $\mathbf{t}_{+\infty}$ and $\text{wt}_{\Sigma}(\Phi(\rho'')) \leq \text{wt}_{\Sigma}(\rho'')$. We follow the three cases of the definition of $\Phi(\rho')$.

1. We cannot have $\Phi(\rho')$ ending in $\mathbf{t}_{+\infty}$ by hypothesis.
2. Suppose now that δ contains a reset and $\Phi(\rho'') = \rho_1 \xrightarrow{t', \delta'} \rho_2$ with $\Delta \text{proj}(\delta') = \delta$. Letting π the first location of ρ_2 , we have $\Phi(\rho') = \rho_1 \xrightarrow{t', \delta'} (\pi, 0)$. Thus

$$\text{wt}_{\Sigma}(\Phi(\rho')) = \text{wt}_{\Sigma}(\Phi(\rho'')) - \text{wt}_{\Sigma}(\rho_2) \leq \text{wt}_{\Sigma}(\rho'') - \text{wt}_{\Sigma}(\rho_2) \quad (4)$$

Let $(\pi', \nu') = \text{last}(\rho_2)$, and $\rho_{\mathcal{U}} = \Phi(\rho'') \xrightarrow{t, \delta''} (\pi'', 0)$, with $\text{NEXT}(\pi', \delta) = (\pi'', \delta'')$. It contains twice a transition with a reset coming from the same transition δ of $\overline{\mathcal{G}}$, therefore $\pi'' \in \{\mathbf{t}_{<0}, \mathbf{t}_{\geq 0}\}$. Notice that $\rho_{\mathcal{U}}$ is conforming to $\tau_{\mathcal{U}}$, since $\Phi(\rho'')$ does and if π' belongs to Max , this follows directly from the definition of $\tau_{\overline{\mathcal{G}}}$ from $\tau_{\mathcal{U}}$ (since $\tau_{\overline{\mathcal{G}}}(\rho''_{\overline{\mathcal{G}}}) = (t, \delta)$ and $\Phi(\rho'') \notin \mathbf{t}_{+\infty}$). Therefore, if $\pi'' = \mathbf{t}_{<0}$, $\text{P}(\rho_{\mathcal{U}}) = -\infty$ and (\star) holds. If $\pi'' = \mathbf{t}_{\geq 0}$, by Lemma 23 applied on $\rho_{\mathcal{U}}$, $\text{wt}_{\Sigma}(\rho_2 \xrightarrow{t, \delta''} (\mathbf{t}_{\geq 0}, 0)) \geq 0$. Combined with (4), we obtain that

$$\begin{aligned} \text{wt}_{\Sigma}(\Phi(\rho')) &\leq \text{wt}_{\Sigma}(\rho'') + \text{wt}_{\Sigma}((\pi', \nu') \xrightarrow{t, \delta''} (\mathbf{t}_{\geq 0}, 0)) \\ &= \text{wt}_{\Sigma}(\rho'') + t \text{wt}'(\pi') + \text{wt}'(\delta'') \\ &= \text{wt}_{\Sigma}(\rho'') + t \overline{\text{wt}}(\ell') + \overline{\text{wt}}(\delta) = \text{wt}_{\Sigma}(\rho') \end{aligned}$$

where we have set ℓ' the last location of ρ'' , that is also the last location of π' .

3. Otherwise, $\Phi(\rho') = \Phi(\rho'') \xrightarrow{t, \delta'} (\pi', \nu)$ if $\text{NEXT}(\pi, \delta) = (\pi', \delta')$ with π the last location of $\Phi(\rho'')$. In this case,

$$\text{wt}_{\Sigma}(\Phi(\rho')) = \text{wt}_{\Sigma}(\Phi(\rho'')) + t \text{wt}'(\pi) + \text{wt}'(\delta') \leq \text{wt}_{\Sigma}(\rho'') + t \overline{\text{wt}}(\ell') + \overline{\text{wt}}(\delta) = \text{wt}_{\Sigma}(\rho')$$

where we have written ℓ' the last location of ρ'' .

This ends the proof by induction. ◀