Deciding the Value of Two-Clock Almost Non-Zeno Weighted Timed Games

Isa Vialard
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Abstract

The Value Problem for weighted timed games (WTGs) consists in determining, given a two-player weighted timed game with a reachability objective and a rational threshold, whether or not the value of the game exceeds the threshold. When restrained to WTGs with non-negative weight, this problem is known to be undecidable for weighted timed games with three or more clocks, and decidable for one-clock WTGs. The Value Problem for two-clock non-negative WTGs, which remained stubbornly open for a decade, was recently shown to be undecidable. In this paper, we show that the Value Problem is decidable when considering two-clock almost non-Zeno WTGs.

2012 ACM Subject Classification Theory of computation → Timed and hybrid models

Keywords and phrases Weighted timed games, decidability, real-time systems

Digital Object Identifier 10.4230/LIPIcs.CSL.2026.33

Related Version *arXiv Version*: https://arxiv.org/abs/2508.00014

Acknowledgements I want to thank Quentin Guilmant for his help in formalizing the proofs in Appendix A.1.

1 Introduction

Introduced by Alur and Dill ([2]) in the early 1990s, a *timed automaton* is an automaton where transitions are limited by time constraints on a set of finite clocks. *Weighted timed automata*, also known as priced timed automata, are timed automata with integer costs added to locations and transitions. These costs can be punctual, or linear in terms of time spent in a location. Timed automata and weighted timed automata are powerful models for real-time systems – for instance task scheduling, controller synthesis, energy-aware systems, etc.

Real-time systems often have to deal with perturbations from an uncontrollable environment (for instance, a user). This can be modelled by *timed games*: timed automata where transitions are divided among two players, the control, who has a reachability objective, versus the environment.

When adding costs to timed games, we obtain *weighted timed games* (WTGs): Min (the control) now attempts to reach a goal location while minimizing the cost of doing so, against her opponent Max (the environment).

A natural problem on WTGs is the following: Given a WTG, can we compute its Value, i.e., the infimum of the optimal cost[1] on all strategies of Min? Or, formulated as a decision problem: is its Value less than or equal to c?

This is the *Value Problem*, not to be confused with the *Existence Problem*: Given a WTG and a threshold c, does Min have a strategy to reach her goal location with cost at most c? These two problems can yield different answers (see Figure 1).

[1] where the optimal cost is the supremum on all possible strategies of Max of the weight of the path produced by the strategy profile.

34th EACSL Annual Conference on Computer Science Logic (CSL 2026).
Editors: Stefano Guerrini and Barbara König; Article No. 33; pp. 33:1–33:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

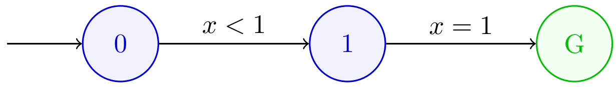

Figure 1 Example of a WTG where the Value is 0, but every strategy of Min yields cost > 0.

In this paper, we focus on the Value Problem. While decidable for weighted timed automata, the Value Problem is undecidable for weighted timed games ([6]).

However, one can recover decidability by restricting the number of clocks. Bouyer *et al.* ([7]) establish that the Value problem is decidable for one-clock WTG with non-negative weights. This decidability result is extended to one-clock WTG with arbitrary weights in [16]. On the other hand, Bouyer *et al.* ([6]) prove undecidability for WTGs with non-negative weight and 3-clocks or more. Brihaye *et al.* ([10]) show undecidability of two-clock WTG with arbitrary weights. Only recently, Guilmant *et al.* ([15]) proved undecidability of two-clock, time-bounded WTG with non-negative weights.

Another way to recover decidability is with non-Zeno (or divergence) properties. A WTG with non-negative weights has a *strictly non-Zeno cost property* when every cycle is of cost at least 1. Intuitively, this property forbids any "Zeno paradox" behaviour. Strictly non-Zeno WTGs can be enfolded into acyclic WTGs; hence, the Value Problem is decidable ([5]). This result was generalized to WTGs with arbitrary weight in [11], for which non-Zenoness becomes *divergence*.[2]

Divergence properties can be weakened into almost-divergence properties. A WTG with non-negative weight is said to be *almost non-Zeno* (or almost strictly non-Zeno, or almost strongly non-Zeno) if its cycles are of weight 0, or at least 1. Bouyer *et al.* ([6]) establish that the Value of such WTGs is approximable[3] (but still undecidable). Busatto-Gaston *et al.* ([12]) extend this result to *almost divergent*[4] WTGs with arbitrary weights.

Number of clocks	Weights in	WTG	Almost divergent WTG	Divergent WTG
1	$\mathbb{N}$	Decidable [7]	Decidable	Decidable
	$\mathbb{Z}$	Decidable [16]	Decidable	Decidable
2	$\mathbb{N}$	Undec. [15]	**Decidable** *(Our contribution)*	Decidable
	$\mathbb{Z}$	Undec.	*Decidability open*	Decidable
≥ 3	$\mathbb{N}$	Undec.	Undec. [6] *(Approximable)*[6]	Decidable [5]
	$\mathbb{Z}$	Undec. *(Non approx.)*[14]	Undec. *(Approximable)* [12]	Decidable [11]

Figure 2 Landscape of WTG decidability and approximability.

[2] A WTG is *divergent* if every strongly connected component has either cycles of weight in $(-\infty, -1]$ or cycles of weight in $[1, \infty)$.

[3] i.e., can be computed to arbitrary precision.

[4] A WTG is *almost divergent* if every strongly connected component has either cycles of weight in $(-\infty, -1] \cup \{0\}$, or cycles of weight in $\{0\} \cup [1, \infty)$.

Contributions

The main theorem of this paper is the following:

▶ **Theorem 1.** *Given a two-player, turn-based, two-clocks, almost non-Zeno weighted timed game with non-negative integer weights, the Value Problem is decidable.*

The proof of Theorem 1 relies on the partial unfolding used in the approximability proof of [6], and on several techniques to turn a WTG into an equivalent, simpler game with desirable properties, such as the relaxing of guards, or adding clock resets to every transition.

2 Definitions

Let $\mathcal{X}$ be a finite set of **clocks**. **Clock constraints** (or **guards**) over $\mathcal{X}$ are expressions of the form $x \bowtie n$, where $x, y \in \mathcal{X}$ are clocks, $\bowtie \in \{<, \leq, =, \geq, >\}$ is a comparison symbol, and $n \in \mathbb{N}$ is a natural number. We write $\mathcal{C}$ to denote the set of all clock constraints over $\mathcal{X}$. A **valuation** on $\mathcal{X}$ is a function $\nu : \mathcal{X} \to \mathbb{R}_{\geq 0}$. For $d \in \mathbb{R}_{\geq 0}$ we denote by $\nu + d$ the valuation such that, for every clock $x \in \mathcal{X}$, $(\nu + d)(x) = \nu(x) + d$. Let $X \subseteq \mathcal{X}$ be a subset of all clocks. We write $\nu[X := 0]$ for the valuation such that, for every clock $x \in X$, $\nu[X := 0](x) = 0$, and $\nu[X := 0](y) = \nu(y)$ for all other clocks $y \notin X$. For $C \subseteq \mathcal{C}$ a set of clock constraints over $\mathcal{X}$, we say that the valuation ν **satisfies** C, denoted $\nu \models C$, if and only if all the comparisons in C hold when replacing each clock x by its corresponding value $\nu(x)$.

▶ **Definition 2.** *A **(turn-based) weighted timed game** is given by a tuple $\mathcal{G} = (L_{\mathsf{Min}}, L_{\mathsf{Max}}, G, \mathcal{X}, T, w)$, where:*

- *L_{Min} and L_{Max} are the (disjoint) sets of **locations** belonging to Players* Min *and* Max *respectively; we let $L = L_{\mathsf{Min}} \cup L_{\mathsf{Max}}$ denote the set of all locations. (In drawings, locations belonging to* Min *are depicted by blue circles, and those belonging to* Max *are depicted by red squares.)*
- *$G \subseteq L_{\mathsf{Min}}$ are the **goal locations**.*
- *$\mathcal{X}$ is a set of clocks.*
- *$T \subseteq (L \setminus G) \times 2^{\mathcal{C}} \times 2^{\mathcal{X}} \times L$ is a set of **(discrete) transitions**. A transition $\ell \xrightarrow{C,X} \ell'$ enables moving from location ℓ to location ℓ', provided all clock constraints in C are satisfied, and afterwards resetting all clocks in X to zero.*
- *$w : (L \setminus G) \cup T \to \mathbb{Z}$ is a **weight function**.*

In the above, we assume that all data (set of locations, set of clocks, set of transitions, set of clock constraints) are finite.

Let $\mathcal{G} = (L_{\mathsf{Min}}, L_{\mathsf{Max}}, G, \mathcal{X}, T, w)$ be a WTG. A **configuration** over $\mathcal{G}$ is a pair (ℓ, ν), where $\ell \in L$ and ν is a valuation on $\mathcal{X}$. Let $d \in \mathbb{R}_{\geq 0}$ be a **delay** and $t = \ell \xrightarrow{C,X} \ell' \in T$ be a discrete transition. One then has a **delayed transition** (or simply a **transition** if the context is clear) $(\ell, \nu) \xrightarrow{d,t} (\ell', \nu')$ provided that $\nu + d \models C$ and $\nu' = (\nu + d)[X := 0]$. Intuitively, control remains in location ℓ for d time units, after which it transitions to location ℓ', resetting all the clocks in X to zero in the process. The **weight** of such a delayed transition is $d \cdot w(\ell) + w(t)$, taking account both of the time spent in ℓ as well as the weight of the discrete transition t.

As noted in [13], without loss of generality one can assume that no configuration (other than those associated with goal locations) is deadlocked; in other words, for any location $\ell \in L \setminus G$ and valuation $\nu \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$, there exists $d \in \mathbb{R}_{\geq 0}$ and $t \in T$ such that $(\ell, \nu) \xrightarrow{d,t} (\ell', \nu')$.[5]

[5] This can be achieved by adding unguarded transitions to a sink location for all locations controlled by Min and unguarded transitions to a goal location for the ones controlled by Max.

Let $k \in \mathbb{N}$. A **run** ρ of length k over $\mathcal{G}$ from a given configuration (ℓ_0, ν_0) is a sequence of matching delayed transitions, as follows:

$$\rho = (\ell_0, \nu_0) \xrightarrow{d_0,t_0} (\ell_1, \nu_1) \xrightarrow{d_1,t_1} \cdots \xrightarrow{d_{k-1},t_{k-1}} (\ell_k, \nu_k)\,.$$

The **weight** of ρ is the cumulative weight of the underlying delayed transitions:

$$\mathsf{weight}(\rho) = \sum_{i=0}^{k-1} (d_i \cdot w(\ell_i) + w(t_i))\,.$$

An infinite run ρ is defined in the obvious way; however, since no goal location is ever reached, its weight is defined to be infinite: $\mathsf{weight}(\rho) = +\infty$.

A run is **maximal** if it is either infinite or cannot be extended further. Thanks to our deadlock-freedom assumption, finite maximal runs must end in a goal location. We refer to maximal runs as **plays**.

We now define the notion of **strategy**. Recall that locations of $\mathcal{G}$ are partitioned into sets L_{Min} and L_{Max}, belonging respectively to Players Min and Max. Let Player $\mathsf{P} \in \{\mathsf{Min}, \mathsf{Max}\}$, and write $\mathcal{FR}^{\mathsf{P}}_{\mathcal{G}}$ to denote the collection of all non-maximal finite runs of $\mathcal{G}$ ending in a location belonging to Player P. A **strategy** for Player P is a mapping $\sigma_{\mathsf{P}} : \mathcal{FR}^{\mathsf{P}}_{\mathcal{G}} \to \mathbb{R}_{\geq 0} \times T$ such that for all finite runs $\rho \in \mathcal{FR}^{\mathsf{P}}_{\mathcal{G}}$ ending in configuration (ℓ, ν) with $\ell \in L_{\mathsf{P}}$, the delayed transition $(\ell, \nu) \xrightarrow{d,t} (\ell', \nu')$ is valid, where $\sigma_{\mathsf{P}}(\rho) = (d, t)$ and (ℓ', ν') is some configuration (uniquely determined by $\sigma_{\mathsf{P}}(\rho)$ and ν).

Let us fix a starting configuration (ℓ_0, ν_0), and let σ_{Min} and σ_{Max} be strategies for Players Min and Max respectively (one speaks of a *strategy profile*). Let us denote by $\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})$ the unique maximal run starting from configuration (ℓ_0, ν_0) and unfolding according to the strategy profile $(\sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})$: in other words, for every strict finite prefix ρ of $\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})$ in $\mathcal{FR}^{\mathsf{P}}_{\mathcal{G}}$, the delayed transition immediately following ρ in $\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})$ is labelled with $\sigma_{\mathsf{P}}(\rho)$.

Recall that the objective of Player Min is to reach a goal location through a play whose weight is as small possible. Player Max has an opposite objective, trying to avoid goal locations, and, if not possible, to maximise the cumulative weight of any attendant play. This gives rise to the following two symmetrical definitions:

$$\overline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0) = \inf_{\sigma_{\mathsf{Min}}} \left\{ \sup_{\sigma_{\mathsf{Max}}} \left\{ \mathsf{weight}(\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})) \right\} \right\} \quad \text{and}$$

$$\underline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0) = \sup_{\sigma_{\mathsf{Max}}} \left\{ \inf_{\sigma_{\mathsf{Min}}} \left\{ \mathsf{weight}(\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})) \right\} \right\}\,.$$

$\overline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0)$ represents the smallest possible weight that Player Min can possibly achieve, starting from configuration (ℓ_0, ν_0), against best play from Player Max, and conversely for $\overline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0)$: the latter represents the largest possible weight that Player Max can enforce, against best play from Player Min.[6] As noted in [13], turned-based WTGs are *determined*, and therefore $\overline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0) = \underline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0)$ for any starting configuration (ℓ_0, ν_0); we denote this common value by $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0)$.

▶ Remark 3. Note that $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0)$ can take on real numbers, or either of the values $-\infty$ and $+\infty$. However, since reachability is decidable in timed games, it is decidable whether $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0) = +\infty$ or not.

[6] Technically speaking, these values may not be literally achievable; however given any $\varepsilon > 0$, both players are guaranteed to have strategies that can take them to within ε of the optimal value.

In the remainder of this paper, every weighted timed game is turn-based, with non-negative weights, of value in $\mathbb{R}$.

3 Unfolding Almost Non-Zeno Weighted Timed Games

Let us first give an informal definition of the region construction $\mathcal{R}(\mathcal{G})$ of a WTG $\mathcal{G}$. In Section 4.2, we will give a more formal definition, in the special case of $[0,1)$-WTGs. The region partition of $\mathcal{G}$ is the finest partition of the clock valuations into regions, where each region is defined by guards of the form $x \bowtie k$ or $x - y \bowtie n$ with $x, y \in \mathcal{X}$, $\bowtie \in \{<, \leq, =, \geq, >\}$ and $n \leq N$ the largest constant in guards of $\mathcal{G}$. For instance, $\{(x,y) \ : \ x = 3, N < y\}$ and $\{(x,y) \ : \ 1 < x < 2, 3 < y < 4, x - 2 < y - 3\}$ are regions.

We denote by $\mathcal{R}(\mathcal{G})$ the region automaton associated with a WTG $\mathcal{G}$(see [2]). In $\mathcal{R}(\mathcal{G})$, every location ℓ is assigned a unique region $\mathsf{reg}(\ell)$ of accessible valuations.

Bouyer *et al.* ([6]) showed that, even though the Value Problem is undecidable for WTG with non-negative weight and 3 clocks or more, it is approximable in the subclass of almost non-Zeno WTG. In this section, we use the structure of their proof of approximability to prove decidability for almost non-Zeno WTGs with 2 clocks.

▶ **Definition 4** (Almost non-Zeno WTG). *A WTG $\mathcal{G}$ is* ***almost non-Zeno*** *if there exists $\kappa > 0$ such that for any finite run ρ in $\mathcal{G}$ that follows a region cycle of $\mathcal{R}(\mathcal{G})$, $\mathsf{weight}(\rho) = 0$ or $\geq \kappa$.*

▶ Remark 5. It is decidable whether a weighted timed game is almost non-Zeno or not (by enumerating all simple cycles in the corner-point abstraction of $\mathcal{G}$, see [4]).

In an acyclic WTG, the value is decidable and can be computed from the target locations up to the initial location, by computing for each node ℓ a function $W_\ell : \mathsf{reg}(\ell) \to \mathbb{R}$ which assigns to a valuation $\nu \in \mathsf{reg}(\ell)$ the optimal weight $\mathsf{Val}_{\mathcal{G}}(\ell, \nu)$. By construction, every W_ℓ is a piecewise linear function.

Intuitively, we will unfold cycles of weight $\geq \kappa$ to obtain a "tree-like" WTG where only cycles of weight 0 are left; we will deal with them separately.

Semi-unfolding

For any WTG $\mathcal{G}$, let $\tilde{\mathcal{G}}$ be the semi-unfolded WTG built from $\mathcal{R}(\mathcal{G})$ in [6]:

First color in green every location and edge that are part of a cycle of weight 0. Observe that you can modify any WTG such that any green location has weight 0[7]: in a trimmed[8] region WTG, if a location $\ell \in L_{\mathsf{P}}$ of weight $p > 0$ is part of a cycle of weight 0, then there exists an outgoing transition from ℓ with a guard $x = 0$ for some clock x. Therefore, as in Figure 3, one can add a location ℓ_0 of weight 0 in L_{P} such that:

- every transition arriving in ℓ arrives in ℓ_0 instead.
- every *green* transition leaving ℓ leaves ℓ_0 instead.
- there is a transition $\ell_0 \overset{x=0}{\to} \ell$.

Thus let us assume that every green location has weight 0. We define $\mathcal{K}$ the **kernel** of $\mathcal{G}$ as the restriction of $\mathcal{R}(\mathcal{G})$ to fully-green strongly connected components. Edges that leave $\mathcal{K}$ are called the **output edges** of $\mathcal{K}$.

---

7 [6] make a similar observation, but their construction implies adding a clock.
8 A trimmed region WTG is a region WTG where we have erased inaccessible locations and redundant guards. It will be detailed later in Definition 13.

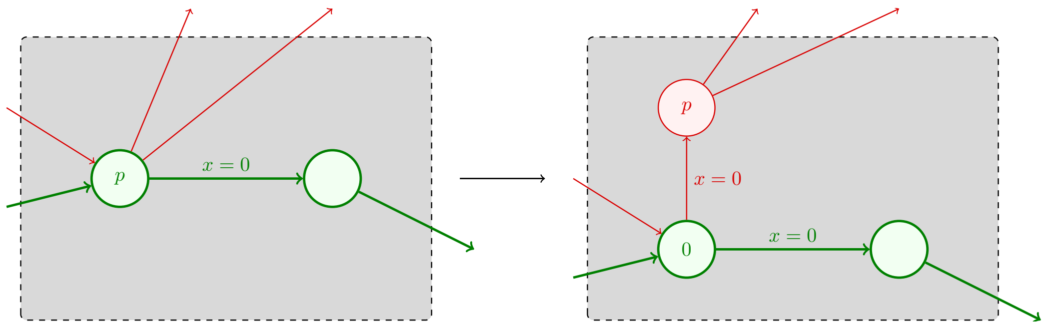

Figure 3 How to ensure that every green location has weight 0. Thick green transitions and locations are part of cycles of weight 0, locations labeled with weight 0 or p belong to the same player.

Then we partially unfold $\mathcal{R}(\mathcal{G})$ into a finite tree structure $\mathcal{T}(\mathcal{G})$: starting from the initial location i as a root, we follow every possible path in $\mathcal{G}$, with a node for each time we visit a (non kernel) location, as to avoid creating cycles. However when along a branch we enter the kernel in some location ℓ, we create a node $\mathcal{K}_\ell$ instead of ℓ, and for each output edge t of $\mathcal{K}$ accessible from ℓ, with t leading to a location ℓ', let ℓ' (or $\mathcal{K}'_\ell$ if $\ell' \in \mathcal{K}$) be a child of $\mathcal{K}_\ell$, and continue to unfold from there.

We stop unfolding when, along any branch, a location or edge with positive weight of $\mathcal{R}(\mathcal{G})$ is visited at least $W/\kappa + 2$ times, where W is an upper bound on the value of $\mathcal{G}$.[9]

To obtain $\tilde{\mathcal{G}}$ from $\mathcal{T}(\mathcal{G})$, replace each node $\mathcal{K}_\ell$ by a copy of the strongly connected component of $\mathcal{K}$ that contains ℓ (see [6] for the formal construction). Then $\mathsf{Val}_{\tilde{\mathcal{G}}}(i,\nu) = \mathsf{Val}_{\mathcal{G}}(i,\nu)$ for any $\nu \in \mathsf{reg}(i)$.

In the partially unfolded games $\tilde{\mathcal{G}}$ with three clocks or more, the cause of undecidability is inside the kernel nodes. Hence, in [6], the approximation happens in the kernels. However, with only two clocks, the value is decidable in kernel weighted timed games:

▶ **Definition 6** (Kernel weighted timed games). *A **kernel weighted timed game** $\mathcal{G}$ is a $[0,1]$-WTG $(L_{\mathsf{Min}}, L_{\mathsf{Max}}, G, \mathcal{X}, T, w, w_{\mathsf{out}})$ where every location or transition is of weight 0, and each target location $\ell \in G$ has an output weight function $w_{\mathsf{out}}(\ell,\cdot) : \mathsf{reg}(\ell) \to \mathbb{R}$ which is continuous and piecewise linear. In later notations, we omit w.*

▶ **Theorem 7.** *For any two-clock kernel WTG $\mathcal{G}$, for any location $i \in \mathcal{G}$, W_i is a continuous piecewise linear function which can be computed through the value iteration algorithm.*

This is the main technical result of this paper, which we will prove in Section 5. Let us show first how Theorem 7 entails value decidability of the partial enfolding $\tilde{\mathcal{G}}$:

▶ **Lemma 8.** *For every node n in the tree $\mathcal{T}(\mathcal{G})$, one can compute W_n a continuous piecewise linear function such that $W_n : \nu \mapsto \mathsf{Val}_{\tilde{\mathcal{G}}}(\ell,\nu)$, where ℓ is either n if $n \notin \mathcal{K}$, or the entrance location of $n = \mathcal{K}_\ell$.*

Proof. In the tree structure of $\tilde{\mathcal{G}}$, consider a node n: if n is a leaf, then $n \in G$. Thus let W_n be the constant null function. Now consider that n is not a leaf, and by induction hypothesis assume that for every child n' of n, W'_n is continuous and piecewise linear. If $n = \ell \notin \mathcal{K}$ then

$$W_n : \nu \mapsto \inf_{\substack{n \xrightarrow{C,X} n' \in \mathcal{T}(\mathcal{G}) \\ \nu+\delta \models C}} / \sup\ W_{n'}(\nu + \delta[X := 0])\,.$$

[9] obtained by using the corner-point abstraction, or considering a memoryless region-uniform strategy for Min.

Thus by induction W_n is also continuous and piecewise linear.

Otherwise, $n = \mathcal{K}_\ell$ for some $\ell \in \mathcal{K}$. Let K be the SCC containing ℓ, and T_{out} the output edges leaving from K. Consider the kernel WTG $K_\ell = (L'_{\mathsf{Min}}, L'_{\mathsf{Max}}, G', \mathcal{X}, T', w_{\mathsf{out}})$ where

- $G' = \{\ell_t \ : \ t \in T_{out}\}$.
- For every P, $L'_{\mathsf{P}} = K \cap L_{\mathsf{P}}$.
- $T' = T_{|K\times K} \cup \left\{ \ell' \overset{C,X}{\to} \ell_t \,|\, t : \ell' \overset{C,X}{\to} \ell'' \in T_{out} \right\}$
- for every $t : \ell' \overset{C,X}{\to} \ell'' \in T_{\mathsf{out}}$, $w_{\mathsf{out}}(\ell_t, \cdot) : \nu \mapsto W_{\ell''}(\nu) + w(t)$, which is piecewise linear by induction hypothesis.
- w' maps to 0 always.

Then $W_n = \mathsf{Val}_{K_\ell}(\ell, \cdot)$, which is piecewise linear according to Theorem 7. ◀

This is sufficient to conclude the proof of Theorem 1.

▶ **Theorem 1.** *Given a two-player, turn-based, two-clocks, almost non-Zeno weighted timed game with non-negative integer weights, the Value Problem is decidable.*

4 Simplifying Transformations of Kernel Games

Before proving Theorem 7, let us apply some useful simplifying transformations that preserve the value. These transformations happen in four steps:

Step 1: Transform a WTG into a WTG with clock values in $[0, 1)$.

Step 2: Transform a WTG into a region trimmed WTG, i.e., a WTG where to every location is assigned a region, and without any "useless" transition, or guard on transition.

Step 3: Transform a trimmed region kernel WTG by relaxing every strict guard into a strict-or-equal guard.

Step 4: Transform a relaxed trimmed region kernel WTG such that every transition resets at least one clock.

Commentaries

Step 1 only serves to lighten notations in the rest of this paper. In terms of state complexity, the transformations $1 + 2$ increase the number of locations as much as the classical region construction.

Trimming a WTG in step 2 is necessary: without it, Step 3 would create pathological cases where relaxing some guards would allow a player to take transitions that would have been unreachable in the original WTG.

Relaxing guards in step 3 is a technique that has merits of its own outside of the scope of the proof. For instance, the value of a WTG might not be reached by an optimal strategy for Min, but could be the infimum produced by a set of strategies ϵ-close to an optimal strategy in the relaxed WTG.

Relaxing guard is also a prerequisite for step 4, which is the most important of all four steps. In a two-clock WTG, resetting at least one clock in each transition allows us to consider only regions in one dimension. Reducing a two-dimension problem to a one-dimension one is key to the termination argument in Section 5.

4.1 Restraining Clock Values to [0,1)

Before presenting the well-known notions of regions and region WTGs, let us first restrict the setting to $[0, 1)$-WTGs, which will simplify the region notations.

▶ **Definition 9.** *A* ***[0,1)-*** WTG *is a weighted timed game where for every reachable configuration (ℓ, ν), $0 \le \nu(x) < 1$ for any clock x.*

▶ **Lemma 10.** *For any* WTG $\mathcal{G}$*, there is an equivalent* $[0,1)$-WTG $\mathcal{G}'$*.*

Proof. See [7], Proposition 2 for a detailed proof. Their proof is for 1-clock WTG, however, the construction can easily be generalized to any number of clocks.

The intuition of the construction is that the information of the integer parts of the clock can be contained in the locations, while the clocks keep track only of the fractional part: Let $\mathcal{G} = (L_{\mathsf{Min}}, L_{\mathsf{Max}}, G, \mathcal{X}, T, w)$ and let us build $\mathcal{G}' = (L'_{\mathsf{Min}}, L'_{\mathsf{Max}}, G, \mathcal{X}, T', w')$. First, w.l.o.g., let us assume that all clocks are bounded by an integer M ([3]). Then, for $\mathsf{P} \in \{ \mathsf{Min}, \mathsf{Max} \}$, let $L'_{\mathsf{P}} = L_{\mathsf{P}} \times M^{|\mathcal{X}|}$, and define T' and w' such that a valuation $\nu = (x_1, \dots, x_{|\mathcal{X}|}) \in [0,1)^{\mathcal{X}}$ in a location $(\ell, n_1, \dots, n_{|\mathcal{X}|})$ in $\mathcal{G}'$ is equivalent to a valuation $(n_1 + x_1, \dots, n_{|\mathcal{X}|} + x_{|\mathcal{X}|})$ in location ℓ in $\mathcal{G}$. Note that every transition of $\mathcal{G}'$, with a guard $x = 1$ for some clock x, resets x. ◀

4.2 Regions and Region Trimmed Games

▶ **Definition 11.** *Let $\mathcal{X}$ be a set of clocks in a $[0,1)$-*WTG*. A* ***region*** *over $\mathcal{X}$ is a tuple $r = (X_0, \dots, X_p, X_{=1})$ such that $X_i \neq \emptyset$ for all $1 \le i \le p$, and $\{X_0, \dots, X_p, X_{=1}\}$ is a partition of $\mathcal{X}$: $\mathcal{X} = \biguplus_{i=0}^{p} X_i$*

We denote by $\mathrm{Reg}_{\mathcal{X}}$ the set of regions over $\mathcal{X}$. A valuation ν is said to ***belong*** *to the region r, denoted by $\nu \sqsubset r$, whenever*

- $\forall x \in \mathcal{X}$, $\nu(x) = 0 \Leftrightarrow x \in X_0$,
- $\forall x \in \mathcal{X}$, $\nu(x) = 1 \Leftrightarrow x \in X_{=1}$,
- $\forall x, y \in \mathcal{X},\ \nu(x) < \nu(y) < 1 \Leftrightarrow \exists i, j \in \{0, \dots, p\}$ *s.t.* $i < j \wedge x \in X_i \wedge y \in X_j$.

For $r = (X_0, \dots, X_p, X_{=1})$ a region and X a non-empty subset of $\mathcal{X}$, we denote by $r[X := 0]$ the region $(X_0 \cup X, X_1 \setminus X, \dots, X_p \setminus X, X_{=1} \setminus X)$. In other words, $r[X := 0]$ is the region such that if ν belongs to r then $\nu[X := 0]$ belongs to $r[X := 0]$. A $[0,1)$***-region*** *is a region $(X_0, \dots, X_p, X_{=1})$ where $X_{=1} = \emptyset$. Let us abuse notation and denote r by $(X_0, \dots, X_p)$. We denote by $\mathrm{Reg}_{\mathcal{X}}^{<}$ the set $[0,1)$-regions over $\mathcal{X}$.*

A ***time-successor*** *of a region r, with $r = (X_0, \dots, X_p) \in \mathrm{Reg}_{\mathcal{X}}^{<}$ is a region $r' = (X'_0, \dots, X'_{p'}, X'_{=1})$ such that either $r' = r$; or $X'_0 = \emptyset$ and $X'_i = X_{i-1}$ for $1 \le i \le p$, and either $X_p = X'_{=1}$ or $X_p = X'_{p+1}$ (and then $X'_{=1} = \emptyset$).*

We often abuse notation and write r for the set of valuations $\nu \sqsubset r$ it represents.

▶ **Definition 12** (Region WTG [4, 12])**.** *Let $\mathcal{G} = (L_{\mathsf{Min}}, L_{\mathsf{Max}}, G, \mathcal{X}, T, w)$ be a $[0,1)$-*WTG*. The* ***region*** WTG *$\mathcal{R}(\mathcal{G})$ of $\mathcal{G}$ is the $[0,1)$-*WTG *$\mathcal{R}(\mathcal{G}) = (L'_{\mathsf{Min}}, L'_{\mathsf{Max}}, G', \mathcal{X}, T', w')$ with*

- $L'_{\mathsf{P}} = L_{\mathsf{P}} \times \mathrm{Reg}_{\mathcal{X}}^{<}$ *for* $\mathsf{P} \in \{\mathsf{Min}, \mathsf{Max}\}$.
- $G' = G \times \mathrm{Reg}_{\mathcal{X}}^{<}$.
- *For every $r = (X_0, \dots X_p) \in \mathrm{Reg}_{\mathcal{X}}^{<}$, for every $r' = (X'_0, \dots, X'_{p'}, X'_{=1}) \in \mathrm{Reg}_{\mathcal{X}}$ a time-successor of r, if $\ell \xrightarrow{C,X} \ell' \in T$ then $(\ell, r) \xrightarrow{C \cup C(r'),X} (\ell', r'[X := 0])$, with*

  $$C(r') = \{(x = 0)\ :\ x \in X'_0\} \cup \{(x = 1)\ :\ x \in X'_{=1}\} \cup \{(0 < x < 1)\ :\ x \in X'_i, 1 \le i \le p\}\,.$$

- *For $\ell \in L_{\mathsf{Min}} \cup L_{\mathsf{Max}}$ and $r \in \mathrm{Reg}_{\mathcal{X}}^{<}$, $w'(\ell, r) = w(\ell)$.*
- *For $t = (\ell, r) \xrightarrow{C \cup C(r'),X} (\ell', r'[X := 0]) \in T'$, $w'(t) = w\left(\ell \xrightarrow{C,X} \ell'\right)$.*

While applying simplifying transformations to $\mathcal{R}(\mathcal{G})$, we wish to preserve the "one-region-per-location" property. Thus let us formally define what is *a* region WTG (as opposed to *the* region WTG). A WTG $\mathcal{G}$ is **a region weighted timed game** if there is a region assignment $\mathsf{reg} : L \to \mathrm{Reg}^{<}_{\mathcal{X}}$, such that for any transition $t : \ell \xrightarrow{C,X} \ell'$, the valuations $\nu + \delta$ with $\nu \sqsubset \mathsf{reg}(\ell)$ and $\delta \geq 0$ that satisfy C are contained in a unique region r, such that $r[X := 0] = \mathsf{reg}(\ell')$. Furthermore, for any initial configuration (i, ν), we require $\nu \sqsubset \mathsf{reg}(i)$.

Obviously $\mathcal{R}(\mathcal{G})$ is a region WTG for any $[0, 1)$-WTG $\mathcal{G}$. For any location ℓ in a region WTG, let $X^{\uparrow}_{\ell} \stackrel{\mathsf{def}}{=} X_p$ with $\mathsf{reg}(\ell) = (X_0, \dots X_p)$.

Let us now "trim" $\mathcal{R}(\mathcal{G})$, i.e., delete every useless transition, and every useless guard on transitions :

▶ **Definition 13** (Trimmed region WTG)**.** *A region* $[0, 1)$*-WTG* $\mathcal{G}$ *is* ***trimmed*** *if for any transition* $t : \ell \xrightarrow{C,X} \ell'$ *and any region* $r = \mathsf{reg}(\ell)$ *in* $\mathcal{G}$*,*

- *for any valuation* $\nu \sqsubset r$ *there exist some* $\delta \geq 0$ *such that* $\nu + \delta \models C$*.*
- *for any* $c \in C$*, there exists a valuation* $\nu \sqsubset r$ *and some* $\delta \geq 0$ *such that* $\nu + \delta$ *is in* $[0, 1]^{\mathcal{X}}$ *and* $\nu + \delta \not\models c$*.*

In other words, there are no inaccessible transitions from any tuple location-region. Furthermore, there are no unnecessary clauses in C (the ones that are always verified from the region). Removing inaccessible transitions and unnecessary clauses can always be done from any region WTG without change in value.

Since every $[0, 1)$-WTG $\mathcal{G}$ is equivalent to the region $[0, 1)$-WTG $\mathcal{R}(\mathcal{G})$, and every region WTG is equivalent to a trimmed region WTG, we can always assume $\mathcal{G}$ to be a trimmed region WTG.

▶ **Observation 14.** *For* $\mathcal{G}$ *a trimmed region* $[0, 1)$*-WTG, for any transition* $t : \ell \xrightarrow{C,X} \ell'$*, with* $\mathsf{reg}(\ell) = (X_0, \dots, X_p)$*:*

- *if* $(y = 0) \in C$ *or* $(y > 0) \in C$ *for some clock* y *then* $y \in X_0$*, in other words* y *is* 0 *on the whole region* r*.*
- *if* $(y = 1) \in C$ *for some clock* y*, then* $y \in X_p$ *in other words* y *is one of the clocks with largest value in* r*.*
- *there cannot be both* $x = 0$ *and* $y = 1$ *in* C *for any two clocks* x, y*.*

Proof. For the first point, notice that if y was not 0, then either the transition or the clause would have been trimmed. For the second one, if $y \notin X_p$ then upon taking the transition with condition $y = 1$ there is a clock $x \in X_p$ such that $x > y = 1$. For the third point, for any valuation in r, there is no δ such that $\nu + \delta$ satisfies both clause. ◀

4.3 Relaxing Strict Guards

A kernel WTG can easily be transformed into a kernel WTG without strict guards, without change in value.

▶ **Definition 15.** *Let* $r = (X_0, \dots, X_p)$*. Then the* ***adherence*** *or* r*, denoted by* $\overline{r}$*, is the set of regions of the form* $(Y_0, \dots, Y_{p'}, Y_{=1})$ *with* $p' \leq p$ *and* $\iota : [0, p' + 1] \to [0, p]$ *strictly increasing such that* $\iota(p' + 1) = p$ *and* $Y_0 = X_0 \cup \dots \cup X_{\iota 0}$ *and* $Y_i = X_{\iota(i-1)+1} \cup \dots \cup X_{\iota(i)}$ *for all* $1 \leq i \leq p'$ *and* $Y_{=1} = X_{\iota(p')+1} \cup \dots \cup X_{\iota(p'+1)}$*.*

Let us abuse notation and write $\nu \sqsubset \overline{r}$ when $\nu \sqsubset r'$ for $r' \in \overline{r}$.

▶ **Lemma 16.** *Let $\mathcal{G}_\prec = (L^\prec_{\mathsf{Min}}, L^\prec_{\mathsf{Max}}, G, \mathcal{X}, T^\prec, w^\prec_{\mathsf{out}})$ be a trimmed region kernel* WTG. *Let $\mathcal{G}_\preceq = (L^\preceq_{\mathsf{Min}}, L^\preceq_{\mathsf{Max}}, G, \mathcal{X}, T^\preceq, w^\preceq_{\mathsf{out}})$ be a copy of $\mathcal{G}_\prec$ where*

- *every guard has been relaxed, i.e., every guard of the form $x > 0$ and $x < 1$ have been replaced by $x \geq 0$ or $x \leq 1$, respectively, for all $x \in \mathcal{X}$.*
- *For any $\ell \in G$, the output function ${w_{\mathsf{out}}}^\preceq(\ell, \cdot)$ is $w^\prec_{\mathsf{out}}(\ell, \cdot)$ extended continuously to $\overline{\mathsf{reg}(\ell)}$ in $\mathcal{G}_\preceq$.*

Then $\mathsf{Val}_{\mathcal{G}_\prec} = \mathsf{Val}_{\mathcal{G}_\preceq}$.

The proof of this theorem relies on a bisimulation argument that is detailed in the appendix.

Note that $\mathcal{G}_\preceq$ is not a $[0,1)$-WTG, but a $[0,1]$-WTG, i.e., every accessible valuation is in $[0,1]^{\mathcal{X}}$. Furthermore, $\mathcal{G}_\preceq$ is not a region WTG:

▶ **Definition 17.** *A* ***relaxed region*** WTG *is a $[0,1]$-*WTG *without strict guard where to each location ℓ is assigned a region $\mathsf{reg}(\ell)$: for any transition $t : \ell \xrightarrow{C,X} \ell'$, the valuations $\nu + \delta$, with $\nu \sqsubset \overline{\mathsf{reg}(\ell)}$ and $\delta \geq 0$, that satisfy C are contained in $\overline{r}$ for a unique region r, such that $r[X := 0] = \mathsf{reg}(\ell')$. Furthermore, initial configuration (i, ν) must verify $\nu \in \overline{\mathsf{reg}(i)}$. A* ***relaxed trimmed region*** WTG *is a relaxed region* WTG *if for any transition $t : \ell \xrightarrow{C,X} \ell'$ and any region $r = \mathsf{reg}(\ell)$ in $\mathcal{G}$,*

- *for any valuation $\nu \sqsubset \overline{r}$ there exist some $\delta \geq 0$ such that $\nu + \delta \models C$.*
- *for any $c \in C$, there exists a valuation $\nu \sqsubset \overline{r}$ and some $\delta \geq 0$ such that $\nu + \delta$ is in $[0,1]^{\mathcal{X}}$ and $\nu + \delta \not\models c$.*

▶ **Lemma 18.** *Let $\mathcal{G}_\prec$ be a region trimmed $[0,1)$-*WTG *of region assignment* reg. *Let $\mathcal{G}_\preceq$ be a copy of $\mathcal{G}_\prec$ where*

- *every guard has been relaxed, i.e., every guard of the form $x > 0$ and $x < 1$ have been replaced by $x \geq 0$ or $x \leq 1$ respectively, for all $x \in \mathcal{X}$.*
- *useless guards (see the second point of Definition 17) have been removed.*

Then $\mathcal{G}_\preceq$ is a relaxed trimmed region WTG, *of region assignment* reg.

4.4 Adding Resets to every Transition

▶ **Lemma 19.** *For any relaxed region trimmed kernel $[0,1]$-*WTG *$\mathcal{G}$, such that $\mathcal{G}$ has no requirement $x < 1$ for any $x \in \mathcal{X}$ then there exists a relaxed region trimmed kernel $[0,1]$-*WTG *$\mathcal{G}'$ of same value and verifying the same conditions such that every transition of $\mathcal{G}'$ is a reset transition or a transition to the target location. Furthermore, any transition of $\mathcal{G}'$ with, for some clock x, a guard of the form $x = 0$ or $x = 1$, resets x.*

See Appendix A.2 for the proof.

5 Value Iteration in Two-clock Kernel Games

In this section, we prove Theorem 7 using the value iteration paradigm (see [1, 9, 8]):

Let $\mathcal{G} = (L_{\mathsf{Min}}, L_{\mathsf{Max}}, G, \mathcal{X}, T, w$ or $w_{\mathsf{out}})$ be a trimmed region (kernel) WTG. In a WTG, the **value iteration algorithm** builds, for each location ℓ and for all $k \geq 0$, a function $\mathsf{opt}^\ell_k : \mathbb{R}^{\mathcal{X}}_{\geq 0} \to \mathbb{R}$ such that $\mathsf{opt}^\ell_k(\nu)$ is the value of the game started in ℓ with clock valuation ν, where Min has to win in at most k steps. The opt functions are built inductively:

- opt^ℓ_0 is the constant 0 function if $\ell \in G$ (or $w_{\mathsf{out}}(\ell, \cdot)$ in the case of a kernel WTG), or the constant $+\infty$ function otherwise.

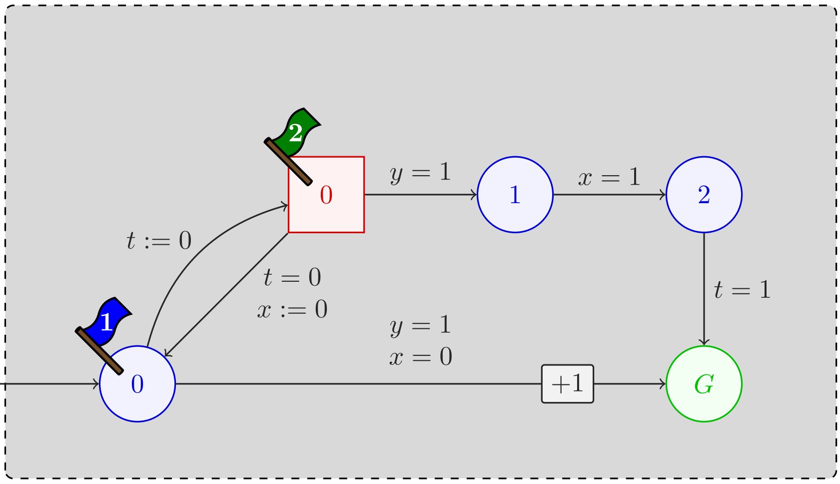

Figure 4 A WTG where the value iteration algorithm does not terminate. Blue circle locations belong to Min, red square locations belong to Max, the green circle location G is the target. The flags serve to easily refer to locations. The $+1$ label is a transition cost.

- for any $k \in \mathbb{N}$, $\mathsf{opt}^{\ell}_{k+1}$ is obtained from the opt functions at step k: if ℓ belongs to Min (resp. Max), then

$$\mathsf{opt}^{\ell}_{k+1}(\nu) = \inf \text{(resp. sup)} \left\{ \mathsf{opt}^{\ell'}_{k}((\nu+\delta)[X := 0]) \;:\; \ell \xrightarrow{C,X} \ell' \in T, \nu + \delta \models C \right\} .$$

Note that $\mathsf{opt}^{\ell}_{k}(\nu) = \mathsf{Val}^{\leq k}(\ell, \nu)$ the value of the game started from configuration ℓ, ν when Min must reach G in at most k steps. Naturally, $\mathsf{opt}^{\ell}_{k+1}(\nu) \leq \mathsf{opt}^{\ell}_{k}(\nu)$ for all valuation ν. If there exists k such that, for all locations ℓ, $\mathsf{opt}^{\ell}_{k+1} = \mathsf{opt}^{\ell}_{k}$, then the value iteration algorithm terminates.

In general, there is no termination guarantee. However, if there exists k such that $\mathsf{opt}^{\ell}_{k+1} = \mathsf{opt}^{\ell}_{k}$ for all ℓ, then $\mathsf{opt}^{\ell}_{k}(\nu) = \mathsf{Val}_{\mathcal{G}}(\ell, \nu)$. This means that the value of the WTG is obtained even when considering plays of length at most k.

Here is an example where the value iteration algorithm does not terminate.

▶ **Example 20.** The WTG $\mathcal{G}$ in Figure 4 is a 3-clock, almost non-Zeno WTG with a value of 1. The kernel of $\mathcal{G}$ contains only the cycle between  and . The cost of the output edge in  is $a + 2\delta$ for a valuation $(x, y, t) = (\delta, a + \delta, 0)$, whereas the cost of the output edge in  is exactly 1, for a valuation $(x, y, t) = (0, 1, 0)$.

An optimal strategy for Min is to loop in the kernel an arbitrary number of times: In this strategy, each time she enters  with valuation $(x, y, t) = (0, a, t)$, she should waits δ such that $2\delta = 1 - a$ and enters , then Max can either reach G with a cost of exactly $a + 2\delta = 1$, or return to . When Min decides to end the game, she then picks $\delta = 1 - a$ instead. Then Max chooses between going to G at cost $1 + (1 - a)$, or letting Min leave from  at cost 1. Each time Min takes the cycle with delay δ such that $2\delta = 1 - a$, y gets closer to 1, thus minimizing the cost of picking $\delta = 1 - a$ at some point. Thus Min has a strategy to reach G with cost > 1, but arbitrarily close to 1 depending on how long she plays. This entails that the value of this WTG is obtained by considering arbitrarily long plays. Thus, the value iteration algorithm does not terminate on this example.

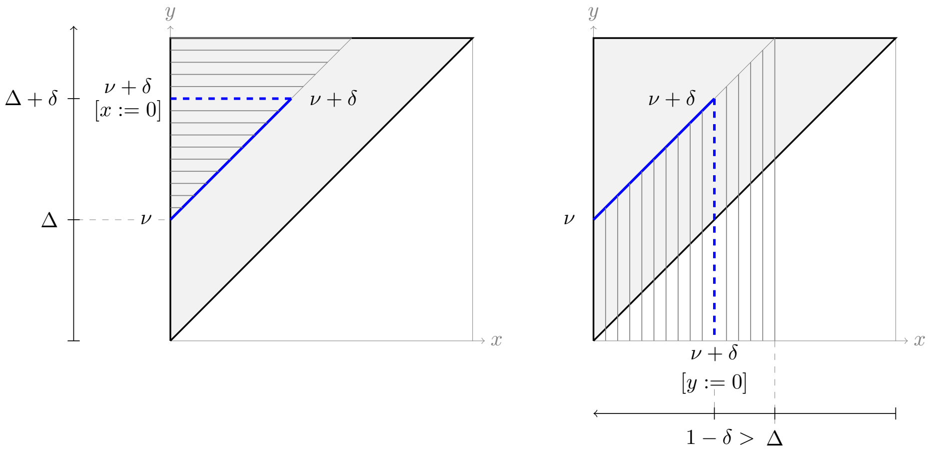

Figure 5 Evolution of Δ in a one-clock-reset transition, without guards $=0$ or $=1$, from region $\{(0,y) \,:\, y \in [0,1]\}$.

However, the value iteration algorithm terminates for two-clock kernel WTGs.

▶ **Theorem 7.** *For any two-clock kernel* WTG $\mathcal{G}$*, for any location* $i \in \mathcal{G}$*,* W_i *is a continuous piecewise linear function which can be computed through the value iteration algorithm.*

Proof. Without loss of generality, let us assume that $\mathcal{G}$ is a relaxed trimmed region kernel $[0,1]$-WTG where every transition is either a transition to a target location, or resets at least one clock (see Lemmas 10, 16, and 19). Furthermore, we assume that there is no transition to the initial location i.[10]

For every location $\ell \notin G \cup \{i\}$, the set of valuations $\overline{\mathsf{reg}(\ell)}$ is either $\{(0,y) \,:\, y \in [0,1]\}$ or $\{(x,0) \,:\, x \in [0,1]\}$, or the singleton $\{(0,0)\}$. Note that the opt^ℓ functions will be defined on $\overline{\mathsf{reg}(\ell)}$ for any location ℓ. This entails that the value iteration algorithm will mainly build 1-dimensional functions.

Let us highlight this observation with a subtle change of variable: Let Δ the **circular clock difference** of a valuation $\nu \in \overline{\mathsf{reg}(\ell)}$ be defined as:

$$\Delta(\nu) = \begin{cases} y \text{ when } \nu = (0,y) \text{ with } y \geq 0, \\ 1 - x \text{ when } \nu = (x,0) \text{ with } x > 0. \end{cases}$$

Now, for any $\ell \notin G \cup \{i\}$, for any $k \in \mathbb{N}$, let $\mathsf{Opt}^\ell_k(\Delta(\nu)) \overset{\mathsf{def}}{=} \mathsf{opt}^\ell_k(\nu)$ for all $\nu \in \overline{\mathsf{reg}(\ell)}$. The function Opt^ℓ_k is either defined on $\{0\}$ if $\overline{\mathsf{reg}(\ell)} = \{(0,0)\}$, or on $[0,1]$.

The following observation motivates this change of variable: In Figures 5 and 6, we consider a transition t, from locations ℓ to ℓ' such that the Opt^ℓ and $\mathsf{Opt}^{\ell'}$ functions are defined on $[0,1]$, with no $=0$ or $=1$ guards. For a transition $(\ell,\nu) \xrightarrow{t,\delta} (\ell',\nu')$, we then observe that $\Delta(\nu') \in [\Delta(\nu),1]$ if $\nu(x) = 0$ (Figure 5), and $\Delta(\nu') \in [0,\Delta(\nu)]$ if $\nu(y) = 0$ (Figure 6). Thus the variation of Δ only depends on the region of ℓ, not ℓ'. In other words, it does not matter which clocks are reset by a transition, just the number of clocks that are reset.

[10] This can be done by making a copy of the initial location, such that all transition that should enter the initial location only enter the copy instead.

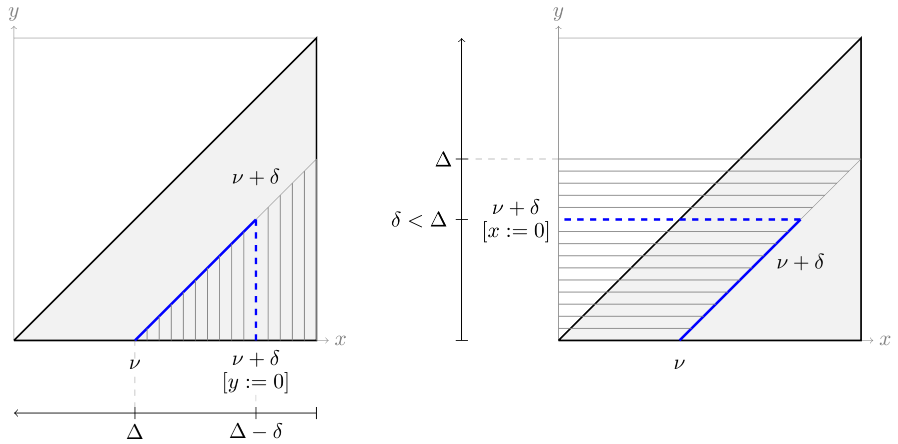

Figure 6 Evolution of Δ in a one-clock-reset transition, without guards $= 0$ or $= 1$, from region $\{(x, 0) \ : \ x \in [0, 1]\}$.

Let us now define the induction relation between the Opt functions. Let ℓ be a location such that $\ell \notin G \cup \{i\}$. Moreover, assume that ℓ belongs to Min (resp. Max). For $k = 0$, $\mathsf{Opt}_k^\ell = \Delta \mapsto +\infty$. Consider $\mathsf{Leaving}(\ell)$ the set of outgoing transition from a location ℓ belonging to Min (resp. Max), with $\ell \neq i$. For each $t : \ell \xrightarrow{C,X} \ell'$, for any $k \in \mathbb{N}$, let us define a function Opt_k^t such that $\mathsf{Opt}_{k+1}^\ell = \min_{t \in \mathsf{Leaving}(\ell)} \mathsf{Opt}_k^t$ (resp. max):

- if ℓ' is a goal location, then for all $k \in \mathbb{N}$,
  - If $\overline{\mathsf{reg}}(\ell) = \{(0, y) \ : \ y \in [0, 1]\}$, then

$$\mathsf{Opt}_k^t = \mathsf{Opt}_0^t : \Delta \mapsto \inf_{\substack{0 \leq \delta \leq 1 - \Delta \\ (\delta, \Delta + \delta) \models C}} w_{\mathsf{out}}(\ell', (\delta, \Delta + \delta)[X := 0]) \text{ (resp. sup)}.$$

  - If $\overline{\mathsf{reg}}(\ell) = \{(x, 0) \ : \ x \in [0, 1]\}$, then

$$\mathsf{Opt}_k^t = \mathsf{Opt}_0^t : \Delta \mapsto \inf_{\substack{0 \leq \delta \leq \Delta \\ (1 - \Delta + \delta, \delta) \models C}} w_{\mathsf{out}}(\ell', (1 - \Delta + \delta, \delta)[X := 0]) \text{ (resp. sup)}.$$

  - If $\overline{\mathsf{reg}}(\ell) = \{0, 0\}$, then $\mathsf{Opt}_k^t(0) = \mathsf{Opt}_0^t(0) = \inf_{\substack{0 \leq \delta \leq 1 \\ (\delta, \delta) \models C}} w_{\mathsf{out}}(\ell', (\delta, \delta)[X := 0])$ (resp. sup).

- if ℓ' is not a goal location, then $X \neq \emptyset$ (t resets one or two clocks).
  - If $X = \{x, y\}$ then, since $\mathcal{G}$ is almost trimmed, every valuation in $\overline{\mathsf{reg}}(\ell)$ can reach $(\ell', (0, 0))$. Therefore $\mathsf{Opt}_k^t = \Delta \mapsto \mathsf{Opt}_k^{\ell'}(0)$.
  - Otherwise, $\mathsf{Opt}_k^{\ell'}$ is defined on $[0, 1]$. Then:
    - If $\overline{\mathsf{reg}}(\ell) = \{(0, y) \ : \ y \in [0, 1]\}$,
      - If $(x = 0) \in C$ (thus $X = \{x\}$) or $(y = 1) \in C$ (thus $X = \{y\}$),[11] then this forces a delay such that t preserves Δ.
      - Otherwise, as observed in Figure 5,

$$\mathsf{Opt}_k^t : \Delta \mapsto \inf_{\Delta \leq \Delta' \leq 1} \mathsf{Opt}_k^{\ell'}(\Delta') \text{ (resp. sup)}.$$

[11] Note that since $\mathcal{G}$ is almost trimmed, both guards cannot be in C at the same time.

- Symetrically, if $\overline{\mathsf{reg}(\ell)} = \{(x,0) \ : \ x \in [0,1]\}$,
  - If $(y=0) \in C$ or $(x=1) \in C$, then $\mathsf{Opt}_k^t = \Delta \mapsto \mathsf{Opt}_k^{\ell'}(\Delta)$.
  - Otherwise, as observed in Figure 6,

$$\mathsf{Opt}_k^t : \Delta \mapsto \inf_{0 \leq \Delta' \leq \Delta} \mathsf{Opt}_k^{\ell'}(\Delta') \text{ (resp. sup)}.$$

- If $\overline{\mathsf{reg}(\ell)} = \{(0,0)\}$, $\mathsf{Opt}_k^t(0) = \inf_{0 \leq \Delta \leq 1} \mathsf{Opt}_k^{\ell'}(\Delta)$ (resp. sup).

We call the functions Opt_0^t, for all transitions t to a goal location, the **projected output functions** of $\mathcal{G}$. The projected output functions serve to initialize the value iteration algorithm on the Opt functions, as do $w_{\mathsf{out}}(\ell,\cdot)$ functions for the value iteration algorithm on the opt functions. Observe that, since $w_{\mathsf{out}}(\ell,\cdot)$ is piecewise linear and continuous for any $\ell \in G$, the projected output functions are continuous piecewise linear functions.

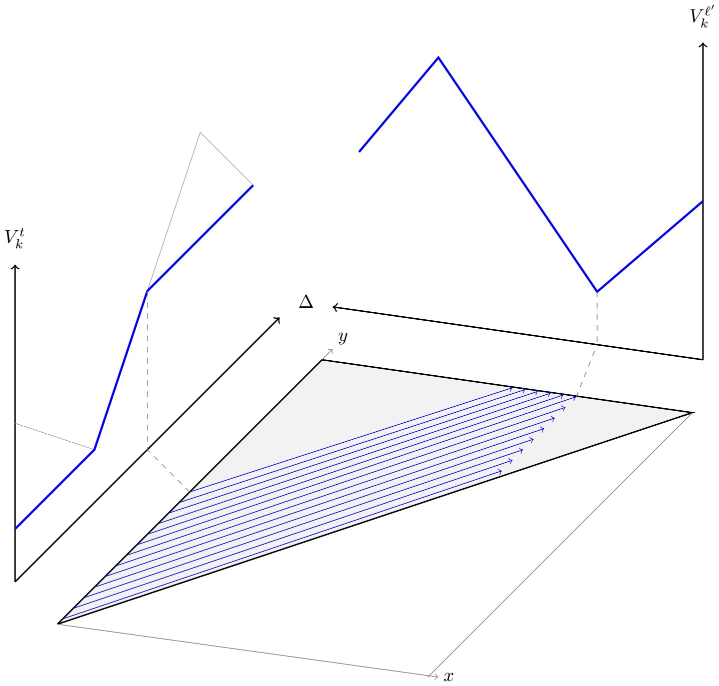

Figure 7 Opt_k induction relation, with t a transition from a Min location ℓ where $x \leq y$. t has no $=0$ or $=1$ guards, and applies $y := 0$.

Thus the functions Opt_k^ℓ are by construction continuous piecewise linear functions. Furthermore, as can be seen in Figure 7, Opt_k^t is obtained from $\mathsf{Opt}_k^{\ell'}$ by replacing $\mathsf{Opt}_k^{\ell'}$ on a finite number of intervals by constant functions $\Delta \mapsto c$ (while preserving continuity) where the constants c are taken among local extremums of $\mathsf{Opt}_k^{\ell'}$.

Hence every linear piece of a function Opt_k^ℓ is:

- either equal to some projected output function,
- or of slope 0, equal to some $z \in \mathbb{R}^+$, where z is a local extremum of some function $\mathsf{Opt}_{k'}^\ell$ for $k' < k$. Hence, by induction, z is either a local extremum of some projected output function, or a local extremum of the minimum or maximum of two projected output functions.

There is a finite number of such pieces, hence only a finite number of way they can be assembled to make a *continuous* piecewise linear function on $[0, 1]$. Thus there is a finite number of functions of this form.

Since the opt and Opt functions decrease at each iteration ($\mathsf{Opt}^{\ell}_{k+1}(\Delta) \leq \mathsf{Opt}^{\ell}_{k}(\Delta)$), then the value iteration algorithm on the Opt functions terminates.

Furthermore, since no transition enters i, opt^{i}_{k} does not affect $\mathsf{opt}^{\ell}_{k+1}$ for any location ℓ. Therefore, the value iteration algorithm on the opt functions aside from opt^{i} stabilizes. Thus, if it terminates in k steps for every location except i, then, adding opt^{i}, the value iteration algorithm terminates in at most $k + 1$ steps. ◀

Note that all transformations in Section 4.4 only serve to make the termination argument of Theorem 7 more visible. However, the value iteration algorithm on two-clock kernel WTGs terminates even without these simplifications. Indeed, termination in k steps entails that Min needs only to consider strategies that access a goal location in at most k steps. Since the transformations described in Section 4 do not make arbitrarily long paths equivalent to one shorter path, then termination of the value algorithm on the transformed kernel WTGs immediately implies termination on the original kernel WTGs. This in turn entails that the value iteration algorithm terminates on the semi-unfolding $\tilde{\mathcal{G}}$, thus on any two-clock WTG with non-negative weight.

Complexity analysis

Assume that the output weight functions of a kernel game $\mathcal{G}$ consist of a total of k pieces. Then the number of possible Opt functions, which bounds the number of steps of the value iteration algorithm according to Theorem 7, is exponential in k.

In the semi-unfolding of a weighted timed game $\mathcal{G}$, the number of pieces of the piecewise linear value functions W_n of nodes n increases at most exponentially along a branch. Each branch has length bounded by $|\mathcal{R}(\mathcal{G})| \cdot (W/\kappa + 2)$ (see [6]). Since $W \leq |\mathcal{R}(\mathcal{G})| \cdot P$, where P is the maximal rate appearing in the automaton, applying the Value Iteration algorithm requires double exponential time in $|\mathcal{R}(\mathcal{G})|$.

Finally, the size of $\mathcal{T}(\mathcal{G})$ is bounded by $|\mathcal{R}(\mathcal{G})|^{|\mathcal{R}(\mathcal{G})| \cdot (W/\kappa+2)+1}$, hence computing the Value is doubly exponential in $|\mathcal{R}(\mathcal{G})|$ (or triple-exponential in the size of the original game).

Note that the approximation algorithm for almost non-Zeno WTGs given in [6] is in double-exponential time. Hence computing the exact value of a two-clock almost non-Zeno WTG is exponentially more complex than computing an approximation.

Conclusion: Extension to arbitrary weights

Decidability actually holds even for almost divergent WTGs with arbitrary (negative and positive) weights. There exists a similar semi-unfolding construction in [12], for approximation of almost-divergent WTGs with arbitrary weights. The main technical difficulty is that a kernel in this semi-unfolding is defined as an SCC where every cycle is of weight 0. However, this does not mean that every location or transition of the kernel has weight 0. Thus accumulated cost can increase (or decrease) while crossing a kernel, even though cycling in a SCC does not change the accumulated cost. The construction to transform such a kernel game into a zero-weight kernel game is quite technical; it will be developed in a journal version of this article.

References

1 Rajeev Alur, Mikhail Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming*, pages 122–133, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. doi:10.1007/978-3-540-27836-8_13.

2 Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994. doi:10.1016/0304-3975(94)90010-8.

3 Gerd Behrmann, Ansgar Fehnker, Thomas Hune, Kim Guldstrand Larsen, Paul Pettersson, Judi Romijn, and Frits W. Vaandrager. Minimum-cost reachability for priced timed automata. In Maria Domenica Di Benedetto and Alberto L. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001, Rome, Italy, March 28-30, 2001, Proceedings*, volume 2034 of *Lecture Notes in Computer Science*, pages 147–161, Berlin, Heidelberg, 2001. Springer. doi:10.1007/3-540-45351-2_15.

4 Patricia Bouyer, Ed Brinkshma, and Kim G. Larsen. Optimal infinite scheduling for multi-priced timed automata. *Formal Methods Syst. Des.*, 32:3–23, 2008. doi:10.1007/s10703-007-0043-4.

5 Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim Guldstrand 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, 24th International Conference, Chennai, India, December 16-18, 2004, Proceedings*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160, Berlin, Heidelberg, 2004. Springer. doi:10.1007/978-3-540-30538-5_13.

6 Patricia Bouyer, Samy Jaziri, and Nicolas Markey. On the Value Problem in Weighted Timed Games. In Luca Aceto and David de Frutos Escrig, editors, *26th International Conference on Concurrency Theory (CONCUR 2015)*, volume 42 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 311–324, Dagstuhl, Germany, 2015. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CONCUR.2015.311.

7 Patricia Bouyer, Kim Guldstrand 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, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings*, volume 4337 of *Lecture Notes in Computer Science*, pages 345–356, Berlin, Heidelberg, 2006. Springer. doi:10.1007/11944836_32.

8 Thomas Brihaye, Gilles Geeraerts, Axel Haddad, Engel Lefaucheux, and Benjamin Monmege. One-clock priced timed games with negative weights. *Logical Methods in Computer Science*, Volume 18, Issue 3, August 2022. doi:10.46298/lmcs-18(3:17)2022.

9 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.

10 Thomas Brihaye, Gilles Geeraerts, Shankara Narayanan Krishna, Lakshmi Manasa, Benjamin Monmege, and Ashutosh Trivedi. Adding negative prices to priced timed games. *CoRR*, abs/1404.5894, 2014. doi:10.48550/arXiv.1404.5894.

11 Damien Busatto-Gaston, Benjamin Monmege, and Pierre-Alain Reynier. Optimal reachability in divergent weighted timed games. In Javier Esparza and Andrzej S. Murawski, editors, *Foundations of Software Science and Computation Structures*, pages 162–178, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg. doi:10.1007/978-3-662-54458-7_10.

12 Damien Busatto-Gaston, Benjamin Monmege, and Pierre-Alain Reynier. Symbolic Approximation of Weighted Timed Games. In Sumit Ganguly and Paritosh Pandya, editors, *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2018)*, volume 122 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2018.28.

13 Damien Busatto-Gaston, Benjamin Monmege, and Pierre-Alain Reynier. Optimal controller synthesis for timed systems. *Log. Methods Comput. Sci.*, 19(1), 2023. doi:10.46298/lmcs-19(1:20)2023.

14 Quentin Guilmant and Joël Ouaknine. Inaproximability in Weighted Timed Games. In Rupak Majumdar and Alexandra Silva, editors, *35th International Conference on Concurrency Theory (CONCUR 2024)*, volume 311 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CONCUR.2024.27.

15 Quentin Guilmant, Joël Ouaknine, and Isa Vialard. The value problem for weighted timed games with two clocks is undecidable, 2025. doi:10.48550/arXiv.2507.10550.

16 Benjamin Monmege, Julie Parreaux, and Pierre-Alain Reynier. Decidability of one-clock weighted timed games with arbitrary weights. *CoRR*, abs/2207.01608, 2022. doi:10.48550/arXiv.2207.01608.

A Proofs of Section 4.4

A.1 Proof of Lemma 16

▶ **Definition 21** (Simulation and bisimulation). *Let* $\mathcal{G} = (L_{\mathsf{Min}}, L_{\mathsf{Max}}, G, \mathcal{X}, T, w)$ *and* $\mathcal{G}' = (L'_{\mathsf{Min}}, L'_{\mathsf{Max}}, G', \mathcal{X}, T', w')$ *be* WTGs. *Then a relation* $R \subseteq (L_{\mathsf{Min}} \times \mathbb{R}^{\mathcal{X}} \times L'_{\mathsf{Min}} \times \mathbb{R}^{\mathcal{X}}) \cup (L_{\mathsf{Max}} \times \mathbb{R}^{\mathcal{X}} \times L'_{\mathsf{Max}} \times \mathbb{R}^{\mathcal{X}})$ *is a* ***simulation relation*** *when*

$$(\ell, \nu)\, R\, (\ell', \nu') \implies \begin{cases} \ell \in G \text{ and } \ell' \in G', \\ \text{or, for all } t \in T,\ \delta \in \mathbb{R}, \text{ there exist } t' \text{ and } \delta' \in \mathbb{R} \text{ such that if} \\ (\ell, \nu) \xrightarrow{\delta, t} (l_2, \nu_2) \text{ then } (\ell', \nu') \xrightarrow{\delta', t'} (\ell'_2, \nu'_2) \text{ where } (\ell_2, \nu_2)\, R\, (\ell'_2, \nu'_2). \end{cases}$$

*R is a **bisimulation** if R and its converse are both simulations.*

▶ **Example 22.** The region relation R between configurations of a same WTG, where $(l_1, \nu_1) R (l_2, \nu_2)$ iff $l_1 = l_2$ and ν_1, ν_2 are in the same region, is a bisimulation.

A (bi)simulation relation R can be extended to runs: $\rho_1\, R\, \rho_2$ when $|\rho_1| = |\rho_2|$, and $\rho_1^{\mathcal{C}}(n)\, R\, \rho_2^{\mathcal{C}}(n)$ for all $n < |\rho_i|$.

▶ **Lemma 23.** *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two* WTGs. *Let R be a bisimulation relation between $\mathcal{G}_1$ and $\mathcal{G}_2$, such that $\mathcal{G}_1$ and $\mathcal{G}_2$ start from configurations $c_1\, R\, c_2$. Assume that $\mathsf{Val}_{\mathcal{G}_i} \neq +\infty$ for $i \in \{1, 2\}$. Then*

$$|\mathsf{Val}_{\mathcal{G}_1} - \mathsf{Val}_{\mathcal{G}_2}| \leq \sup \left\{ |\mathsf{weight}(\rho_1) - \mathsf{weight}(\rho_2)| \;\middle|\; \begin{array}{c} \rho_1, \rho_2 \text{ plays of } \mathcal{G}_1, \mathcal{G}_2 \\ \rho_1\, R\, \rho_2 \end{array} \right\}.$$

Proof. For all σ_1^{Min} a strategy for Min on $\mathcal{G}_1$ and σ_2^{Max} a strategy for Max on $\mathcal{G}_2$, there exists σ_2^{Min} and σ_1^{Max} strategies for Min and Max on $\mathcal{G}_2$ and $\mathcal{G}_1$ respectively, such that $\rho_1\, R\, \rho_2$ for $\rho_i = play(c_i, \sigma_i^{\mathsf{Min}}, \sigma_i^{\mathsf{Max}})$. (In any Min location, σ_2^{Min} simulates σ_1^{Min} following R. Similarly in any Max location, σ_1^{Max} simulates σ_2^{Max} following R.)

Note that $\sigma_1^{\mathsf{Max}}, \sigma_2^{\mathsf{Min}}$ and ρ_1, ρ_2 are functions of $\sigma_1^{\mathsf{Min}}, \sigma_2^{\mathsf{Max}}$. However, we omit these arguments to lighten notations.

Since the set of strategies obtained by such a simulation is included in the set of all strategies,

$$\sup_{\sigma_2^{\mathsf{Max}}} \mathsf{weight}(\rho_1) \leq \sup_{\sigma^{\mathsf{Max}}} V_{\mathcal{G}_1}(\sigma_1^{\mathsf{Min}}, \sigma^{\mathsf{Max}}) \text{ for a fixed } \sigma_1^{\mathsf{Min}},$$

and

$$\inf_{\sigma_1^{\mathsf{Min}}} \mathsf{weight}(\rho_2) \geq \inf_{\sigma^{\mathsf{Min}}} V_{\mathcal{G}_2}(\sigma^{\mathsf{Min}}, \sigma_2^{\mathsf{Max}}) \text{ for a fixed } \sigma_2^{\mathsf{Max}}.$$

Therefore

$$\inf_{\sigma_1^{\mathsf{Min}}} \sup_{\sigma_2^{\mathsf{Max}}} \mathsf{weight}(\rho_1) \leq \inf_{\sigma^{\mathsf{Min}}} \sup_{\sigma^{\mathsf{Max}}} \mathsf{weight}(play_{\mathcal{G}_1}(c_1, \sigma^{\mathsf{Min}}, \sigma^{\mathsf{Max}})) = \mathsf{Val}_{\mathcal{G}_1}$$

and

$$\inf_{\sigma_1^{\mathsf{Min}}} \sup_{\sigma_2^{\mathsf{Max}}} \mathsf{weight}(\rho_2) \geq \sup_{\sigma_2^{\mathsf{Max}}} \inf_{\sigma_1^{\mathsf{Min}}} \mathsf{weight}(\rho_2)$$
$$\geq \sup_{\sigma^{\mathsf{Max}}} \inf_{\sigma^{\mathsf{Min}}} \mathsf{weight}(play_{\mathcal{G}_2}(c_2, \sigma^{\mathsf{Min}}, \sigma^{\mathsf{Max}})) = \mathsf{Val}_{\mathcal{G}_2}$$

Therefore $\mathsf{Val}_{\mathcal{G}_2} - \mathsf{Val}_{\mathcal{G}_1} \leq \sup \{\mathsf{weight}(\rho_2) - \mathsf{weight}(\rho_1) \ : \ \rho_1 \, R \, \rho_2\}$.
Reasoning in mirror, one can obtain

$$\mathsf{Val}_{\mathcal{G}_1} - \mathsf{Val}_{\mathcal{G}_2} \leq \sup \{\mathsf{weight}(\rho_1) - \mathsf{weight}(\rho_2) \ : \ \rho_1 \, R \, \rho_2\} \ .$$

and combine to conclude. ◀

Now let us define the following relation between valuations: For $\epsilon > 0$, two valuations ν, ν' are **ϵ-neighbours** if there exists $\epsilon_1, \epsilon_2 \geq 0$ such that $\epsilon_1 + \epsilon_2 < \epsilon$, and for any clock $x \in \mathcal{X}$, $\nu(x) - \nu'(x) \in [-\epsilon_1, \epsilon_2]$.

Let $\mathcal{G}_{\prec} = (L^{\prec}_{\mathsf{Min}}, L^{\prec}_{\mathsf{Max}}, G, \mathcal{X}, T^{\prec}, w^{\prec}_{\mathsf{out}})$ be a trimmed region weighted timed game. We consider $\mathcal{G}_{\preceq} = (L^{\preceq}_{\mathsf{Min}}, L^{\preceq}_{\mathsf{Max}}, G, \mathcal{X}, T^{\preceq}, w^{\preceq}_{\mathsf{out}})$ a copy of $\mathcal{G}_{\prec}$ where every strict guard has been relaxed into a strict-or-equal guard.

▶ **Lemma 24.** *For any configurations $(\ell_{\prec}, \nu_{\prec})$ in $\mathcal{G}_{\prec}$, and $(\ell_{\preceq}, \nu_{\preceq})$ in $\mathcal{G}_{\preceq}$,*
Let $(\ell_{\prec}, \nu_{\prec}) \, R_\epsilon \, (\ell_{\preceq}, \nu_{\preceq})$ iff
- $\ell_{\prec} = \ell_{\preceq}$
- $\nu_{\prec} \sqsubset \mathsf{reg}(\ell_{\prec})$ *and* $\nu_{\preceq} \sqsubset \overline{\mathsf{reg}(\ell_{\prec})}$
- $\nu_{\prec}$ *and* $\nu_{\preceq}$ *are ϵ-neighbours*

Then R_ϵ is a bisimulation relation.

Proof. First observe that one side of the bisimulation is easier than the other one, since transitions in $\mathcal{G}_{\preceq}$ are more permissive. Therefore we will detail only the other side:

Let $t : \ell_1 \xrightarrow{C,X} \ell_2$. Let $\overline{C}$ be the relaxation of C. Let ν_1 be a valuation belonging to $\mathsf{reg}(\ell_1)$ in $\mathcal{G}_{\prec}$ and let ν'_1 be an ϵ-neighbour of ν_1. By definition, there are $\epsilon_1, \epsilon_2 \geq 0$ such that $\epsilon_1 + \epsilon_2 < \epsilon$ and for any clock $x \in \mathcal{X}$, $\nu_1(x) - \nu'_1(x) \in [-\epsilon_1, \epsilon_2]$.

Let $\delta' \geq 0$ such that $\nu'_1 + \delta' \models \overline{C}$ and let $\nu'_2 = (\nu'_1 + \delta')[X := 0]$. Then let us show that there exists $\delta \geq 0$ such that $\nu_1 + \delta \models C$, and $(\ell_2, \nu_2) R_\epsilon (\ell'_2, \nu'_2)$ with $\nu_2 = (\nu_1 + \delta)[X := 0]$. To prove that, one only needs to show that $\nu'_1 + \delta'$ is in the adherence of the region of $\nu_1 + \delta$, and that $\nu_1 + \delta$ and $\nu'_1 + \delta'$ are ϵ-neighbours.

Consider the interval

$$\Delta = \left\{ \delta \geq 0 \;\middle|\; \begin{array}{c} \nu_1 + \delta \models C \\ \nu_1 + \delta \sqsubset r \text{ s.t. } \nu'_1 + \delta' \sqsubset \overline{r} \end{array} \right\}$$

Since $\mathcal{G}_{\prec}$ is region trimmed, there exists δ such that $\nu_1 + \delta \models C$, and every valuation of the form $\nu + \delta$ that satisfy C with $\nu \sqsubset \mathsf{reg}(\ell_1)$ belong to a unique region r_t. Since $\nu'_1 \sqsubset \overline{\mathsf{reg}(\ell_1)}$ and $\nu'_1 + \delta' \models \overline{C}$, $\nu'_1 + \delta' \sqsubset \overline{r_t}$. Therefore, Δ is not empty.

If $\delta' \in \Delta$, then $(\nu_1 + \delta')[X := 0]$ and $\nu_1' + \delta'$ are ϵ-neighbours. Otherwise, when $\delta' \notin \Delta$, it entails that Δ is constrained by some guards in C. These guards can be of the form $x = 1$, $x > 0$, $x < 1$ or $x \leq 1$ for any $x \in \mathcal{X}$ [12] .

- If there is a guard $x = 1$ in C for some $x \in \mathcal{X}$, then Δ is the singleton $\{1 - \nu_1(x)\}$. Indeed Δ is not empty and $\delta \overset{\text{def}}{=} 1 - \nu_1(x)$ is the only delay that can satisfy the guard $x = 1$. Note that, for the same reason, $\delta' = 1 - \nu_1'(x)$. By definition of Δ, $\nu_2 = (\nu_1 + \delta)$ and ν_2' are in the same region. Moreover, for all $y \in \mathcal{X}$, if $y \in X$, then $\nu_2(y) = \nu_2'(y) = 0$ and if $y \in \mathcal{X} \setminus X$, then $(\nu_1(y) + \delta) - (\nu_1'(y) + \delta' \in [-(\epsilon_1 + \delta' - \delta), \epsilon_2 - (\delta' - \delta)]$. Now, $\delta' - \delta = \nu_1(x) - \nu_1'(x) \in [-\epsilon_1, \epsilon_2]$ entails that $\epsilon_2' \overset{\text{def}}{=} \epsilon_2 - (\delta' - \delta) \geq 0$, and $\epsilon_1' \overset{\text{def}}{=} \epsilon_1 + (\delta' - \delta) \geq 0$. Finally, $\epsilon_1' + \epsilon_2' = \epsilon_1 + \epsilon_2 < \epsilon$.

Now assume that there are no such guards in C:

- If there is a guard $x > 0$ in C for some $x \in \mathcal{X}$ such that $\nu_1 + \delta' \not\models (x > 0)$ but $\nu_1' + \delta' \models (x > 0)$ then $\nu_1'(x) + \delta' = 0$. Furthermore, since $\mathcal{G}_{\prec}$ is trimmed, $\nu_1(x) = 0$. Therefore, Δ is an interval of the form $]0, \dots]$. Pick $\delta \in \Delta$ such that $0 < \delta < \epsilon - (\epsilon_1 + \epsilon_2)$. Let $\epsilon_2' \overset{\text{def}}{=} \epsilon_2 - (\delta' - \delta) = \epsilon_2 + \delta$, and $\epsilon_1' \overset{\text{def}}{=} \min\{0, \epsilon_1 + (\delta' - \delta)\} \geq 0$. Then $\epsilon_1' + \epsilon_2' \leq \epsilon_1 + \epsilon_2 + \delta < \epsilon$, hence $\nu_1 + \delta$ and $\nu_1' + \delta'$ are ϵ-neighbourss.
- If there is a guard $x < 1$ in C (resp. $x \leq 1$) for some $x \in \mathcal{X}$, such that $\nu_1' + \delta' \models (x \leq 1)$ but $\nu_1 + \delta' \not\models (x < 1)$ (resp. $x \leq 1$), then $\Delta < \delta'$. If $\delta' + \nu_1'(x) - \nu_1(x) \in \Delta$ then pick $\delta = \delta' + \nu_1'(x) - \nu_1(x)$. Since $\delta' - \delta = \nu_1(x) - \nu_1'(x) \in [-\epsilon_1, \epsilon_2]$, $\nu_1 + \delta$ and $\nu_1' + \delta'$ are ϵ-neighbours.
  Otherwise pick $\delta \in \Delta$ such that $1 - \epsilon + (\epsilon_1 + \epsilon_2) < \nu_1(x) + \delta < 1$. Then $\nu_1(x) - \nu_1'(x) < \delta' - \delta < \nu_1(x) - \nu_1'(x) + \epsilon - (\epsilon_1 + \epsilon_2)$, with $\nu_1(x) - \nu_1'(x) \in [-\epsilon_1, \epsilon_2]$. Let $\epsilon_2' \overset{\text{def}}{=} \min(0, \epsilon_2 - (\delta' - \delta)) \geq 0$ and $\epsilon_1' \overset{\text{def}}{=} \epsilon_1 + (\delta' - \delta) \geq 0$. Then $\epsilon_1' + \epsilon_2' < \epsilon$, hence $\nu_1' + \delta'$ is an ϵ-neighbour of $\nu_1 + \delta$. ◀

We are now ready to prove Lemma 16.

▶ **Lemma 16.** *Let $\mathcal{G}_{\prec} = (L^{\prec}_{\mathsf{Min}}, L^{\prec}_{\mathsf{Max}}, G, \mathcal{X}, T^{\prec}, w^{\prec}_{\mathsf{out}})$ be a trimmed region kernel* WTG. *Let $\mathcal{G}_{\preceq} = (L^{\preceq}_{\mathsf{Min}}, L^{\preceq}_{\mathsf{Max}}, G, \mathcal{X}, T^{\preceq}, w^{\preceq}_{\mathsf{out}})$ be a copy of $\mathcal{G}_{\prec}$ where*

- *every guard has been relaxed, i.e., every guard of the form $x > 0$ and $x < 1$ have been replaced by $x \geq 0$ or $x \leq 1$, respectively, for all $x \in \mathcal{X}$.*
- *For any $\ell \in G$, the output function ${w_{\mathsf{out}}}^{\preceq}(\ell, \cdot)$ is $w^{\prec}_{\mathsf{out}}(\ell, \cdot)$ extended continuously to $\overline{\mathsf{reg}(\ell)}$ in $\mathcal{G}_{\preceq}$.*

Then $\mathsf{Val}_{\mathcal{G}_{\prec}} = \mathsf{Val}_{\mathcal{G}_{\preceq}}$.

Proof. $\mathcal{G}_{\prec}$ and $\mathcal{G}_{\preceq}$ start from the same configuration, thus their first configurations are in bisimulation R_ϵ for any $\epsilon > 0$.

Therefore, according to Lemma 23,

$$|\mathsf{Val}_{\mathcal{G}_{\prec}} - \mathsf{Val}_{\mathcal{G}_{\preceq}}| \leq \sup \left\{ |\mathsf{weight}(\rho_{\prec}) - \mathsf{weight}(\rho_{\preceq})| \;\middle|\; \begin{array}{c} \rho_{\prec}, \rho_{\preceq} \text{ plays of } \mathcal{G}_{\prec}, \mathcal{G}_{\preceq} \\ \rho_{\prec} \, R_\epsilon \, \rho_{\preceq} \end{array} \right\} .$$

In a kernel, for any finite run ρ of length n, $\mathsf{weight}(\rho) = w_{\mathsf{out}}(\rho^C(n))$. Let s be the maximal slope (in absolute value, in one variable) in functions $w_{\mathsf{out}}(\ell, \cdot)$. Then

$$\begin{aligned} &\sup \{|\mathsf{weight}(\rho_1) - \mathsf{weight}(\rho_2)| \;:\; \rho_1 \, R_\epsilon \, \rho_2\} \\ &\quad \leq \sup \{|w_{\mathsf{out}}(\ell, \nu_1) - w_{\mathsf{out}}(\ell, \nu_2)| \;:\; \ell \in G, (\ell, \nu_1) \, R_\epsilon \, (\ell, \nu_2)\} \\ &\quad \leq \epsilon \cdot s \cdot |\mathcal{X}| \end{aligned}$$ ◀

Conclude with Lemma 23.

[12] There cannot be any guard of the form $x = 0$ in C, otherwise $\Delta = \{\delta'\} = \{0\}$.

A.2 Proof of Lemma 19

▶ **Lemma 25.** *For* $A, B = \mathsf{Min}, \mathsf{Max}$ *or* $\mathsf{Max}, \mathsf{Min}$*. In a relaxed region kernel* $[0,1]$*-WTG, consider a transition* $t : \ell_A \xrightarrow{C,X} \ell_B$ *between* ℓ_A *a location belonging to Player* A*, and* ℓ_B *a location belonging to Player* B*, such that* C *has no guard of the form* $x = 0, x = 1, x < 1$*, and* $X = \emptyset$*. Then adding the guard* $x = 1$ *to* C *for some* $x \in X^{\uparrow}_{\ell_A}$ *does not change the value.*

Proof. Pick some $x \in X^{\uparrow}_{\ell_A}$. By picking a delay $\delta < 1 - \nu(x)$, Player A offers Player B more options than if they picked $\delta = 1 - \nu(x)$. From the perspective of B, if a larger delay is advantageous, then they can take it from ℓ_B at cost 0. Hence it is optimal for either A or B to pick the largest delay possible, i.e. $\delta = 1 - \nu(x)$. However, since $w(\ell_A) = w(\ell_B) = 0$, forcing A to take a delay in ℓ_A which would have been taken in ℓ_B by Player B given the chance does not change the value. Thus, restricting A to strategies which, when choosing t from a valuation ν, choose a delay δ such that $(\nu + \delta)(x) = 1$ for all $x \in X^{\uparrow}_{\ell_A}$ does not change the value of the WTG. ◀

▶ **Lemma 19.** *For any relaxed region trimmed kernel* $[0,1]$*-WTG* $\mathcal{G}$*, such that* $\mathcal{G}$ *has no requirement* $x < 1$ *for any* $x \in \mathcal{X}$ *then there exists a relaxed region trimmed kernel* $[0,1]$*-WTG* $\mathcal{G}'$ *of same value and verifying the same conditions such that every transition of* $\mathcal{G}'$ *is a reset transition or a transition to the target location. Furthermore, any transition of* $\mathcal{G}'$ *with, for some clock* x*, a guard of the form* $x = 0$ *or* $x = 1$*, resets* x*.*

Proof. Let $\mathcal{G} = (L_{\mathsf{Min}}, L_{\mathsf{Max}}, G, \mathcal{X}, T, w, w_{\mathsf{out}})$. We assume that Max does not have full control over any cycle (i.e., in any cycle there is a Min location from which she can decide to leave the cycle). Indeed, if Max could reach such a cycle, the Value of $\mathcal{G}$ would be $+\infty$. Furthermore, let us assume that there is no Min self-loop (a transition $\ell \in L_{\mathsf{Min}} \xrightarrow{C,X} \ell$) with $X = \emptyset$ in $\mathcal{G}$: it does not make strategic sense for Min to take such a loop, hence they can be deleted without change in value.

Let us transform $\mathcal{G}$ through the following operations. For any $t : \ell_1 \xrightarrow{C,X} \ell_2$ of $\mathcal{G}$ such that ℓ_2 is not a target location, and $X = \emptyset$ and C has no guards of the form $x = 1$ for any clock x:

- If C has a $x = 0$ requirement for some clock x, then add x to X.
- If C has no $x = 0$ requirement for every clock x, and ℓ_1 and ℓ_2 belong to the same player, then remove t and, for any $t' : \ell_2 \to \ell_3$ with $\ell_3 \neq \ell_1$, [13] create a transition $t' : \ell_1 \to \ell_3$ such that $C(t'') = C(t) \cup C(t')$
- If C has no $x = 0$ requirement for every clock x, and ℓ_1 and ℓ_2 do not belong to the same player, then let us add a $x = 1$ requirement to C where $x \in X^{\uparrow}_{\ell_1}$. According to Lemma 25, it does not change the value.

After these transformations, every transition without reset in $\mathcal{G}$ is either a transition to a target location, or has a $x = 1$ guard for some $x \in \mathcal{X}$.

Then let us build $\mathcal{G}' = (L'_{\mathsf{Min}}, L'_{\mathsf{Max}}, G', \mathcal{X}, T', w', w'_{\mathsf{out}})$ a kernel WTG where:

- For any player P, let $L'_{\mathsf{P}} = L_{\mathsf{P}} \cup \{\ell_{\downarrow} \;:\; \ell \in L_{\mathsf{P}}\}$.
- Let $G' = G \cup \{\ell_{\downarrow} \;:\; \ell \in G\}$.
- Start from $T' = \emptyset$. For any $t : \ell \xrightarrow{C,X} \ell'$ in T,

[13] Adding a transition in the case $\ell_3 = \ell_1$ would create a self-loop: Min has no use for self-loops without reset, and we assume that Max has full control over no cycle, so this situation never happens when ℓ_1 belongs to Max.

  - if $\ell' \in G$ then for all $X \subseteq X^{\uparrow}_{\ell}$ add $t' : \ell \overset{C,X}{\to} \ell'$ and $t' : \ell_{\downarrow} \overset{C_{\downarrow},X}{\to} \ell'_{\downarrow}$ to T', where $C_{\downarrow}$ is $C \cup \left\{x = 0 \ : \ x \in X^{\uparrow}_{\ell}\right\}$ deprived of guards $(x = 1)$ for all $x \in X^{\uparrow}_{\ell}$. Note that here $X^{\uparrow}_{\ell}$ is defined according to $\mathsf{reg}(\ell)$ the region assignment in the relaxed trimmed region WTG $\mathcal{G}$.
  - if $\ell' \notin G$ and $X = \emptyset$ then $(x = 1) \in C$ for some $x \in X^{\uparrow}_{\ell}$. Then add $t' : \ell \overset{C,X^{\uparrow}_{\ell}}{\to} \ell'_{\downarrow}$ and $t'' : \ell_{\downarrow} \overset{C_{\downarrow},X^{\uparrow}_{\ell}}{\to} \ell'_{\downarrow}$ to T'.
  - if $\ell' \notin G$ and $X = X^{\uparrow}_{\ell}$, then add $t' : \ell \overset{C,X}{\to} \ell'$ and $t'' : \ell_{\downarrow} \overset{C_{\downarrow},X}{\to} \ell'$ to T', such that $X' = X^{\uparrow}_{\ell}$ and $C' = C \cup \{x = 0 \ : \ x \in X_1\} \setminus \{x = 1 \ : \ x \in X_1\}$.
  - if $\ell' \notin G$ and $X \neq \emptyset$, then add $t' : \ell \overset{C,X}{\to} \ell'$ and $t'' : \ell_{\downarrow} \overset{C',X'}{\to} \ell'_{\downarrow}$ to T' such that $X' = X^{\uparrow}_{\ell}$ and $C' = C \cup \{x = 0 \ : \ x \in X_1\} \setminus \{x = 1 \ : \ x \in X_1\}$.
- For all $\ell \in G$, let $w'_{\mathsf{out}}(\ell, \nu) = w_{\mathsf{out}}(\ell, \nu)$, $w'_{\mathsf{out}}(\ell_{\downarrow}, \nu) = w_{\mathsf{out}}(\ell, \nu')$ where $\nu'(x) = \nu(x)$ for all $x \notin X$, and $\nu'(x) = 1$ otherwise.

Intuitively, in $\mathcal{G}$, when one or several clocks are made to reach 1 by a guard, there will usually be some urgent transitions taken until all these clocks have been reset. In $\mathcal{G}'$, those clocks are immediately reset, and control moves to a location $\ell_{\downarrow}$. All paths leaving $\ell_{\downarrow}$ have $(x = 0)$ conditions (for all x that have been reset in $\mathcal{G}'$ but not in $\mathcal{G}$) to guarantee urgency. A valuation ν' in a location $\ell_{\downarrow}$ in $\mathcal{G}'$ is thus equivalent to a valuation ν in ℓ in $\mathcal{G}$ iff $\nu(x) = \nu'(x)$ if $x \notin X^{\uparrow}_{\ell}$, and $\nu(x) = 1$ and $\nu'(x) = 0$ for $x \in X^{\uparrow}_{\ell}$.

$\mathcal{G}'$ is a relaxed region WTG with region assignment $\mathsf{reg}'(\ell) = \mathsf{reg}(\ell)$ and $\mathsf{reg}'(\ell_{\downarrow}) = (X_0 \cup X_p, X_1, \ldots, X_{p-1})$ when $\mathsf{reg}(\ell) = (X_0, X_1, \ldots, X_p)$. From there, trim $\mathcal{G}'$ to obtain a relaxed trimmed region WTG. ◀