Symbolic Approximation of Weighted Timed Games

Damien Busatto-Gaston
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
damien.busatto-gaston@lis-lab.fr

Benjamin Monmege
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
benjamin.monmege@univ-amu.fr

Pierre-Alain Reynier
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
pierre-alain.reynier@univ-amu.fr

Abstract

Weighted timed games are zero-sum games played by two players on a timed automaton equipped with weights, where one player wants to minimise the accumulated weight while reaching a target. Weighted timed games are notoriously difficult and quickly undecidable, even when restricted to non-negative weights. For non-negative weights, the largest class that can be analysed has been introduced by Bouyer, Jaziri and Markey in 2015. Though the value problem is undecidable, the authors show how to approximate the value by considering regions with a refined granularity. In this work, we extend this class to incorporate negative weights, allowing one to model energy for instance, and prove that the value can still be approximated, with the same complexity. In addition, we show that a symbolic algorithm, relying on the paradigm of value iteration, can be used as an approximation scheme on this class.

2012 ACM Subject Classification D.2.4 Software/Program Verification

Keywords and phrases Weighted timed games, Real-time systems, Game theory, Approximation

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2018.yy

1 Introduction

The design of programs verifying some real-time specifications is a notoriously difficult problem, because such programs must take care of delicate timing issues, and are difficult to debug a posteriori. One research direction to ease the design of real-time software is to automatise the process. The situation may usually be modelled into a timed game, played by a *controller* and an antagonistic *environment*: they act, in a turn-based fashion, over a *timed automaton* [2], namely a finite automaton equipped with real-valued variables, called clocks, evolving with a uniform rate. A simple, yet realistic, objective for the controller is to reach a target location. We are thus looking for a *strategy* of the controller, that is a recipe dictating how to play so that the target is reached no matter how the environment plays. Reachability timed games are decidable [4], and EXPTIME-complete [19].

Weighted extensions of these games have been considered in order to measure the quality of the winning strategy for the controller [9, 1]: when the controller has a winning strategy in a given reachability timed game, the quantitative version of the game helps choosing a good one with respect to some metrics. This means that the game now takes place over a *weighted (or priced) timed automaton* [5, 3], where transitions are equipped with weights, and locations with rates of weights (the cost is then proportional to the time spent in this location, with the rate as proportional coefficient). While solving weighted timed automata

38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science.
Editors: Paritosh Pandya, and Sumit Ganguly; Article No. yy; pp. yy:1–yy:32
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

has been shown to be PSPACE-complete [6] (i.e. the same complexity as the non-weighted version), weighted timed games are known to be undecidable [12]. This has led to many restrictions in order to regain decidability, the first and most interesting one being the class of strictly non-Zeno cost with only non-negative weights (in transitions and locations) [9]: this hypothesis requires that every execution of the timed automaton that follows a cycle of the region automaton has a weight far from 0 (in interval $[1, +\infty)$, for instance).

Negative weights are crucial when one wants to model energy or other resources that can grow or decrease during the execution of the system to study. In [16], we have recently extended the strictly non-Zeno cost restriction to weighted timed games in the presence of negative weights in transitions and/or locations. We have described there the class of *divergent weighted timed games* where each execution that follows a cycle of the region automaton has a weight far from 0, i.e. in $(-\infty, -1] \cup [1, +\infty)$. We were able to obtain a doubly-exponential-time algorithm to compute the values and almost-optimal strategies, while deciding the divergence of a weighted timed game is PSPACE-complete. These complexity results match the ones that could be obtained in the non-negative case from [9, 1].

The techniques used to obtain these results cannot be extended if the conditions are slightly relaxed. For instance, if we add the possibility for an execution of the timed automaton following a cycle of the region automaton to have weight *exactly 0*, the decision problem is known to be undecidable [10], even with non-negative weights only. For this extension, in the presence of non-negative weights only, it has been proposed an approximation scheme to compute arbitrarily close estimates of the optimal value [10]. To this end, the authors consider regions with a refined granularity so as to control the precision of the approximation. In this work, our contribution is two-fold: first, we extend the class considered in [10] to the presence of negative weights; second, we show that the approximation can be obtained using a symbolic computation, based on the paradigm of value iteration.

More precisely, we define the class of *almost-divergent weighted timed games* where, for each strongly connected component (SCC) of the region automaton, executions following a cycle of this SCC have a weight either all in $(-\infty, -1] \cup \{0\}$, or all in $\{0\} \cup [1, +\infty)$. In contrast, the *divergent* condition is equivalent to the same property on the strongly connected components, but without the presence of singleton $\{0\}$. Given an almost-divergent weighted timed game, an initial configuration c and a threshold ε, we compute a value that we guarantee to be ε-close to the optimal value when the play starts from c. Moreover, deciding if a weighted timed game is almost-divergent is a PSPACE-complete problem.

In order to approximate almost-divergent weighted timed games, we first adapt the approximation scheme of [10] to our setting. At the very core of their scheme is the notion of *kernels* that collect all cycles of weight exactly 0 in the game. Then, a semi-unfolding of the game (in which kernels are not unfolded) of bounded depth is shown to be equivalent to the original game. Adapting this scheme to negative weights requires to address new issues:

- The definition and the approximation of these kernels is much more intricate in our setting (see Sections 3 and 5). Indeed, with only non-negative weights, a cycle of weight 0 only encounters locations and transitions with weight 0. It is no longer the case with arbitrary weights, both for discrete weights on transitions (that could alternate between weight $+1$ and -1, e.g.) and continuous rates on locations: for this continuous part, this requires to keep track of the real-time dynamics of the game.
- Some valuations may have value $-\infty$. While it is undecidable in general whether a configuration has value $-\infty$ (see Appendix A.1), we prove that it is decidable for almost-divergent weighted timed games (see Lemma 8).
- The identification of an adequate bound to define an equivalent semi-unfolding of bounded

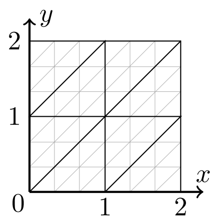
depth is more difficult in our setting, as having guarantees on weight accumulation is harder (we can lose accumulated weight). We deal with this by evaluating how large the value of a configuration can be, provided it is not infinite. This is presented in Section 4.

We also develop, in Section 6, a second approximation schema, more *symbolic* than [10], in the sense that it avoids the a priori refinement of regions. Instead, all computations are performed in a symbolic way using the techniques developed in [1]. This allows to mutualise as much as possible the different computations: comparing these schemas with the evaluation of MDPs or quantitative games like mean-payoff or discounted-payoff, it is the same improvement as the use of *value iteration* techniques instead of techniques based on the unfolding of the model into a finite tree that contains many times the same location.

Due to lack of space, omitted proofs can be found in the appendix.

2 Weighted timed games

Clocks, guards and regions. We let X be a finite set of variables called clocks. A valuation of clocks is a mapping $\nu\colon X \to \mathbb{R}_{\geqslant 0}$. For a valuation ν, $d \in \mathbb{R}_{\geqslant 0}$ and $Y \subseteq X$, we define the valuation $\nu+d$ as $(\nu+d)(x) = \nu(x)+d$, for all $x \in X$, and the valuation $\nu[Y := 0]$ as $(\nu[Y := 0])(x) = 0$ if $x \in Y$, and $(\nu[Y := 0])(x) = \nu(x)$ otherwise. The valuation $\mathbf{0}$ assigns 0 to every clock. A guard on clocks of X is a conjunction of atomic constraints of the form $x \bowtie c$, where $\bowtie \in \{\leqslant, <, =, >, \geqslant\}$ and $c \in \mathbb{Q}$ (we allow for rational coefficients as we will need to refine the granularity in the following). Guard $\overline{g}$ is the closed version of guard g where every open constraint $x < c$ or $x > c$ is replaced by its closed version $x \leqslant c$ or $x \geqslant c$. A valuation $\nu\colon X \to \mathbb{R}_{\geqslant 0}$ satisfies an atomic constraint $x \bowtie c$ if $\nu(x) \bowtie c$. The satisfaction relation is extended to all guards g naturally, and denoted by $\nu \models g$. We let $\mathsf{Guards}(X)$ denote the set of guards over X.



We rely on the crucial notion of regions, as introduced in the seminal work on timed automata [2]: a region is a set of valuations, that are all time-abstract bisimilar. We will also need some refinement of regions, with respect to a granularity $1/N$, with $N \in \mathbb{N}$. Formally, with respect to the set X of clocks and a constant M, a $1/N$-region r is a subset of valuations characterised by the vector $(\iota_x)_{x\in X} = (\min(MN, \lfloor \nu(x)N \rfloor))_{x\in X} \in [0, MN]^X$ and the order of fractional parts of $\nu(x)N$, given as a partition $X = X_0 \uplus X_1 \uplus \cdots \uplus X_m$ of clocks: a valuation ν is in this $1/N$-region r if (i) $\lfloor \nu(x)N \rfloor = \iota_x$, for all clocks $x \in X$; (ii) $\nu(x) = 0$ for all $x \in X_0$; (iii) all clocks $x \in X_i \neq \emptyset$ verify that $\nu(x)N$ have the same fractional part, for all $1 \leqslant i \leqslant m$. We denote by $\mathsf{Reg}_N(X, M)$ the set of $1/N$-regions, and we write $\mathsf{Reg}(X, M)$ as a shortcut for $\mathsf{Reg}_1(X, M)$. We recover the traditional notion of region for $N = 1$. E.g., the figure on the right depicts regions $\mathsf{Reg}(\{x, y\}, 2)$ as well as a their refinement $\mathsf{Reg}_3(\{x, y\}, 2)$. For any integer guard g, either all valuations of a given $1/N$-region satisfy g, or none of them do. A $1/N$-region r' is said to be a time successor of the $1/N$-region r if there exist $\nu \in r$, $\nu' \in r'$, and $d > 0$ such that $\nu' = \nu + d$. Moreover, for $Y \subseteq X$, we let $r[Y := 0]$ be the $1/N$-region where clocks of Y are reset.

Weighted timed games. A weighted timed game (WTG) is then a tuple $\mathcal{G} = \langle L = L_{\mathsf{Min}} \uplus L_{\mathsf{Max}}, \Delta, \mathsf{wt}, L_T, \mathsf{wt}_T\rangle$ where L_{Min} and L_{Max} are finite disjoint subsets of locations belonging to Min and Max, respectively, $\Delta \subseteq L \times \mathsf{Guards}(X) \times 2^X \times L$ is a finite set of transitions, $\mathsf{wt}\colon \Delta \uplus L \to \mathbb{Z}$ is the weight function, associating an integer weight with each transition and location, $L_T \subseteq L_{\mathsf{Min}}$ is a subset of target locations for player Min, and $\mathsf{wt}_T\colon L_T \times \mathbb{R}^X_{\geqslant 0} \to \mathbb{R}_\infty$ is a function mapping each target location and valuation of the clocks

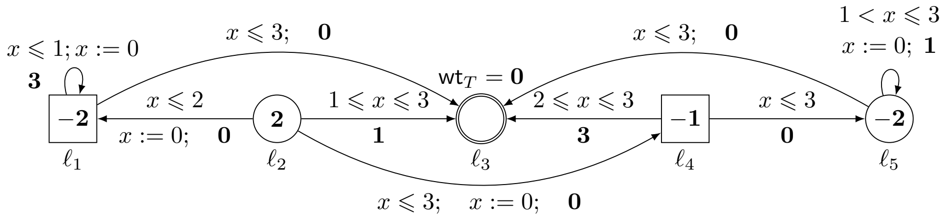
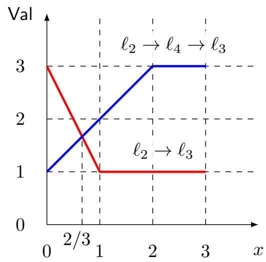


Figure 1 A weighted timed game. Locations belonging to Min (resp. Max) are depicted by circles (resp. squares). The target location is ℓ_3, whose output weight function is null. It is easy to observe that location ℓ_1 (resp. ℓ_5) has value $+\infty$ (resp. $-\infty$). As a consequence, the value in ℓ_4 is determined by the edge to ℓ_3, and depicted in blue on the picture below. In location ℓ_2, the value associated with the transition to ℓ_3 is depicted in red, and the value in ℓ_2 is obtained as the minimum of these two curves. Observe the intersection in $x = 2/3$ requiring to refine the regions.

to a final weight of $\mathbb{R}_\infty = \mathbb{R} \uplus \{-\infty, +\infty\}$ (possibly 0, $+\infty$, or $-\infty$). The addition of target weights is not standard, but we will use it in the process of solving those games: anyway, it is possible to simply map each target location to the weight 0, allowing us to recover the standard definition. Without loss of generality, we suppose the absence of deadlocks except on target locations, i.e. for each location $\ell \in L \backslash L_T$ and valuation ν, there exists $(\ell, g, Y, \ell') \in \Delta$ such that $\nu \models g$, and no transition starts in L_T.



The semantics of a WTG $\mathcal{G}$ is defined in terms of a game played on an infinite transition system whose vertices are configurations of the WTG. A configuration is a pair (ℓ, ν) with a location and a valuation of the clocks. Configurations are split into players according to the location. A configuration is final if its location is a target location of L_T. The alphabet of the transition system is given by $\mathbb{R}_{\geqslant 0} \times \Delta$ and will encode the delay that a player wants to spend in the current location, before firing a certain transition. For every delay $d \in \mathbb{R}_{\geqslant 0}$, transition $\delta = (\ell, g, Y, \ell') \in \Delta$ and valuation ν, there is an edge $(\ell, \nu) \xrightarrow{d,\delta} (\ell', \nu')$ if $\nu + d \models g$ and $\nu' = (\nu + d)[Y := 0]$. The weight of such an edge e is given by $d \times \mathsf{wt}(\ell) + \mathsf{wt}(\delta)$. An example is depicted on Figure 1.

A *finite play* is a finite sequence of consecutive edges $\rho = (\ell_0, \nu_0) \xrightarrow{d_0,\delta_0} (\ell_1, \nu_1) \xrightarrow{d_1,\delta_1} \cdots (\ell_k, \nu_k)$. We denote by $|\rho|$ the length k of ρ. The concatenation of two finite plays ρ_1 and ρ_2, such that ρ_1 ends in the same configuration as ρ_2 starts, is denoted by $\rho_1\rho_2$. We let $\mathsf{FPlays}_{\mathcal{G}}$ be the set of all finite plays in $\mathcal{G}$, whereas $\mathsf{FPlays}_{\mathcal{G}}^{\mathsf{Min}}$ (resp. $\mathsf{FPlays}_{\mathcal{G}}^{\mathsf{Max}}$) denote the finite plays that end in a configuration of Min (resp. Max). A *play* is then a maximal sequence of consecutive edges (it is either infinite or it reaches L_T).

A *strategy* for Min (resp. Max) is a mapping $\sigma \colon \mathsf{FPlays}_{\mathcal{G}}^{\mathsf{Min}} \to \mathbb{R}_{\geqslant 0} \times \Delta$ (resp. $\sigma \colon \mathsf{FPlays}_{\mathcal{G}}^{\mathsf{Max}} \to \mathbb{R}_{\geqslant 0} \times \Delta$) such that for all finite plays $\rho \in \mathsf{FPlays}_{\mathcal{G}}^{\mathsf{Min}}$ (resp. $\rho \in \mathsf{FPlays}_{\mathcal{G}}^{\mathsf{Max}}$) ending in non-target configuration (ℓ, ν), there exists an edge $(\ell, \nu) \xrightarrow{\sigma(\rho)} (\ell', \nu')$. A play or finite play $\rho = (\ell_0, \nu_0) \xrightarrow{d_0,\delta_0} (\ell_1, \nu_1) \xrightarrow{d_1,\delta_1} \cdots$ conforms to a strategy σ of Min (resp. Max) if for all k such that (ℓ_k, ν_k) belongs to Min (resp. Max), we have that $(d_k, \delta_k) = \sigma((\ell_0, \nu_0) \xrightarrow{d_0,\delta_0} \cdots (\ell_k, \nu_k))$. A strategy σ is *memoryless* if for all finite plays ρ, ρ' ending in the same configuration, we have that $\sigma(\rho) = \sigma(\rho')$. For all strategies σ_{Min} and σ_{Max} of players Min and Max, respectively, and for all configurations (ℓ_0, ν_0), we let $\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})$ be the outcome of σ_{Max} and σ_{Min}, defined as the only play conforming to σ_{Max} and σ_{Min} and starting in (ℓ_0, ν_0).

The objective of Min is to reach a target configuration, while minimising the accumulated weight up to the target. Hence, we associate to every finite play $\rho = (\ell_0, \nu_0) \xrightarrow{d_0,\delta_0} (\ell_1, \nu_1) \xrightarrow{d_1,\delta_1} \cdots (\ell_k, \nu_k)$ its cumulated weight, taking into account both discrete and continuous costs: $\mathsf{wt}_\Sigma(\rho) = \sum_{i=0}^{k-1} \mathsf{wt}(\ell_i) \times d_i + \mathsf{wt}(\delta_i)$. Then, the weight of a play ρ, denoted by $\mathsf{wt}_\mathcal{G}(\rho)$, is defined by $+\infty$ if ρ is infinite (does not reach L_T), and $\mathsf{wt}_\Sigma(\rho) + \mathsf{wt}_T(\ell_T, \nu)$ if it ends in (ℓ_T, ν) with $\ell_T \in L_T$. Then, for all locations ℓ and valuation ν, we let $\mathsf{Val}_\mathcal{G}(\ell, \nu)$ be the value of $\mathcal{G}$ in (ℓ, ν), defined as $\mathsf{Val}_\mathcal{G}((\ell, \nu)) = \inf_{\sigma_{\mathsf{Min}}} \sup_{\sigma_{\mathsf{Max}}} \mathsf{wt}_\mathcal{G}(\mathsf{play}((\ell, \nu), \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}}))$, where the order of the infimum and supremum does not matter, since WTGs are known to be determined[1]. We say that a strategy $\sigma^\star_{\mathsf{Min}}$ of Min is ε-optimal if, for all (ℓ, ν), and all strategies σ_{Max} of Max, $\mathsf{wt}_\mathcal{G}(\mathsf{play}((\ell, \nu), \sigma_{\mathsf{Max}}, \sigma^\star_{\mathsf{Min}})) \leqslant \mathsf{Val}_\mathcal{G}(\ell, \nu) + \varepsilon$. It is said optimal if this holds for $\varepsilon = 0$. A symmetric definition holds for optimal strategies of Max. If the game is clear from the context, we may drop the index $\mathcal{G}$ from all previous notations.

As usual in related work [1, 9, 10], we assume that the starting WTGs have guards where all constants are integers, and all clocks are *bounded*, i.e. there is a constant $M \in \mathbb{N}$ such that every transition of the WTG is equipped with a guard g such that $\nu \models g$ implies $\nu(x) \leqslant M$ for all clocks $x \in X$. We denote by $w^L_{\max}$ (resp. $w^\Delta_{\max}$, $w^e_{\max}$) the maximal weight in absolute values of locations (resp. of transitions, edges) of $\mathcal{G}$, i.e. $w^L_{\max} = \max_{\ell \in L} |\mathsf{wt}(\ell)|$ (resp. $w^\Delta_{\max} = \max_{\delta \in \Delta} |\mathsf{wt}(\delta)|$, $w^e_{\max} = M w^L_{\max} + w^\Delta_{\max}$). For technical reasons that will become clear later, we also assume that the output weight functions are piecewise linear with a finite number of pieces and are continuous on each region.

Region and corner abstractions. The region automaton, or region game, $\mathcal{R}_N(\mathcal{G})$ (abbreviated as $\mathcal{R}(\mathcal{G})$ when $N = 1$) of a game $\mathcal{G} = \langle L = L_{\mathsf{Min}} \uplus L_{\mathsf{Max}}, \Delta, \mathsf{wt}, L_T, \mathsf{wt}_T \rangle$ is the WTG with locations $S = L \times \mathsf{Reg}_N(X, M)$ and all transitions $((\ell, r), g'', Y, (\ell', r'))$ with $(\ell, g, Y, \ell') \in \Delta$ such that the model of guard g'' (i.e. all valuations ν such that $\nu \models g''$) is a $1/N$-region r'', time successor of r such that r'' satisfies the guard g, and $r' = r''[Y := 0]$. Distribution of locations to players, final locations and weights are taken according to $\mathcal{G}$. We call *path* a finite or infinite sequence of transitions in this automaton, and we denote by π the paths. A play ρ in $\mathcal{G}$ is projected on a path π in $\mathcal{R}_N(\mathcal{G})$, by replacing actual valuations by the $1/N$-regions containing them: we say that ρ *follows* the path π. It is important to notice that, even if π is a *cycle* (i.e. starts and ends in the same location of the region game), there may exist plays following it in $\mathcal{G}$ that are not cycles, due to the fact that regions are sets of valuations. By projecting away the region information of $\mathcal{R}_N(\mathcal{G})$, we simply obtain:

▶ **Lemma 1.** *For all $\ell \in L$, $1/N$-regions r, and $\nu \in r$, $\mathsf{Val}_\mathcal{G}(\ell, \nu) = \mathsf{Val}_{\mathcal{R}_N(\mathcal{G})}((\ell, r), \nu)$.*

On top of regions, we will need the corner-point abstraction techniques introduced in [8]. A valuation v is said to be a corner of a $1/N$-region r, if it belongs to the topological closure $\overline{r}$ and has coordinates multiple of $1/N$ ($v \in (1/N)\mathbb{N}^X$). We call corner state a triple (ℓ, r, v) that contains information about a location (ℓ, r) of the region-game $\mathcal{R}_N(\mathcal{G})$, and a corner v of the $1/N$-region r. Every region has at most $|X|+1$ corners. We now define the corner-point abstraction $\mathcal{C}_N(\mathcal{G})$ of a WTG $\mathcal{G}$ as the WTG obtained as a refinement of $\mathcal{R}_N(\mathcal{G})$ where guards on transitions are enforced to stay on one of the corners of the current $1/N$-region: the locations of $\mathcal{C}_N(\mathcal{G})$ are all corner states of $\mathcal{R}_N(\mathcal{G})$, associated to each player accordingly, and transitions are all $((\ell, r, v), g'', Y, (\ell', r', v'))$ such that $t = ((\ell, r), g, Y, (\ell', r'))$ is a transition of $\mathcal{R}_N(\mathcal{G})$ such that the model of guard g'' is a corner v'' satisfying the guard $\overline{g}$ (recall that $\overline{g}$ is the closed version of g), $v' = v''[Y := 0]$, and there exist two

[1] The determinacy result is stated in [13] for WTG (called priced timed games) with one clock, but the proof does not use the assumption on the number of clocks.

valuations $\nu \in r$, $\nu' \in r'$ such that $((\ell,r),\nu) \xrightarrow{d',t} ((\ell',r'),\nu')$ for some $d' \in \mathbb{R}_{\geqslant 0}$ (this constraint must be added to ensure that the transition between corners is not spurious). Because of this closure operation, we must also define properly the final weight function: we simply define it over the only valuation v reachable in location (ℓ,r,v) (with $\ell \in L_T$) by $\mathsf{wt}_T((\ell,r,v),v) = \lim_{\nu\to v,\nu\in r} \mathsf{wt}_T(\ell,\nu)$ (the limit is well defined since wt_T is piecewise linear with a finite number of pieces on region r).

The WTG $\mathcal{C}_N(\mathcal{G})$ can be seen as a *weighted game* (with final weights), i.e. a WTG without clocks (which means that there are only weights on transitions), by removing guards, resets and rates of locations, and replacing the weights of transitions by the actual weight of jumping from one corner to another: for instance, the previous transition $(((\ell,r),v),g'',Y,((\ell',r'),v'))$ becomes an edge from $((\ell,r),v)$ to $((\ell',r'),v')$ with weight $d \times \mathsf{wt}(\ell) + \mathsf{wt}(t)$ (for all possible values of d, which requires to allow for multi-edges[2]). Note that delay d is necessarily a rational of the form α/N with $\alpha \in \mathbb{N}$, since it must relate corners of $1/N$-regions. In particular, this proves that the cumulated weight $\mathsf{wt}_\Sigma(\rho)$ of a finite play ρ in $\mathcal{C}_N(\mathcal{G})$ is indeed a rational number with denominator N.

We will call *corner play* a play ρ in the corner-point abstraction $\mathcal{C}_N(\mathcal{G})$: it can also be interpreted as a timed execution in $\mathcal{G}$ where all guards are closed (as explained in the definition before). It straightforwardly projects on a finite path π in the region game $\mathcal{R}_N(\mathcal{G})$: in this case, we say again that ρ follows π. Corner plays allow one to obtain faithful information on the plays that follow the same path:

▶ **Lemma 2.** *If π is a finite path in $\mathcal{R}_N(\mathcal{G})$, the set $\{\mathsf{wt}_\Sigma(\rho) \mid \rho$ finite play following $\pi\}$ is an interval bounded by the minimum and the maximum values of the set $\{\mathsf{wt}_\Sigma(\rho) \mid \rho$ finite corner play of $\mathcal{C}_N(\mathcal{G})$ following $\pi\}$.*

Value iteration. We will rely on the value iteration algorithm described in [1] for a WTG $\mathcal{G}$.

If V represents a value function—i.e. a mapping from configurations of $L \times \mathbb{R}^X_{\geqslant 0}$ to a value in $\mathbb{R}_\infty$—we denote by $V_{\ell,\nu}$ the image $V(\ell,\nu)$, for better readability, and by V_ℓ the function mapping each valuation ν to $V_{\ell,\nu}$. One step of the game is summarised in the following operator $\mathcal{F}$ mapping each value function V to a value function $V' = \mathcal{F}(V)$ defined by $V'_{\ell,\nu} = \mathsf{wt}_T(\ell,\nu)$ if $\ell \in L_T$, and otherwise

$$V'_{\ell,\nu} = \begin{cases} \sup_{(\ell,\nu)\xrightarrow{d,\delta}(\ell',\nu')} \big[d \times \mathsf{wt}(\ell) + \mathsf{wt}(\delta) + V_{\ell',\nu'}\big] & \text{if } \ell \in L_{\mathsf{Max}} \\ \inf_{(\ell,\nu)\xrightarrow{d,\delta}(\ell',\nu')} \big[d \times \mathsf{wt}(\ell) + \mathsf{wt}(\delta) + V_{\ell',\nu'}\big] & \text{if } \ell \in L_{\mathsf{Min}} \end{cases} \tag{1}$$

where $(\ell,\nu) \xrightarrow{d,\delta} (\ell',\nu')$ ranges over valid edges in $\mathcal{G}$. Then, starting from V^0 mapping every configuration to $+\infty$, except for the target mapped to wt_T, we let $V^i = \mathcal{F}(V^{i-1})$ for all $i > 0$. The value function V^i represents the value $\mathsf{Val}^i_{\mathcal{G}}$, which is intuitively what Min can guarantee when forced to reach the target in at most i steps.

More formally, we define $\mathsf{wt}^i_{\mathcal{G}}(\rho)$ the weight of a maximal play ρ at horizon i, as $\mathsf{wt}_{\mathcal{G}}(\rho)$ if ρ reaches a target state in at most i steps, and $+\infty$ otherwise. Using this alternative definition of the weight of a play, we can obtain a new game value $\mathsf{Val}^i_{\mathcal{G}}(\ell,\nu) = \inf_{\sigma_{\mathsf{Min}}} \sup_{\sigma_{\mathsf{Max}}} \mathsf{wt}^i_{\mathcal{G}}(\mathsf{play}((\ell,\nu),\sigma_{\mathsf{Max}},\sigma_{\mathsf{Min}}))$. Then, if $\mathcal{G}$ is a tree of depth d, $V^i{=}\mathsf{Val}_{\mathcal{G}}$ if $i \geq d$.

The mappings V^0_ℓ are piecewise linear for all ℓ, and $\mathcal{F}$ preserves piecewise linearity over regions, so all iterates V^i_ℓ are piecewise linear with a finite number of pieces. In [1],

[2] The only case where several edges could link two corners using the same transition is when all clocks are reset in Y, in which case there is a choice for delay d.

it is proved that V_ℓ^i has a number of pieces (and can be computed within a complexity) exponential in i and in the size of $\mathcal{G}$ when $\mathsf{wt}_T = 0$. This result can be extended to handle negative weights in $\mathcal{G}$ and output weights $\mathsf{wt}_T \neq 0$.

Problems. We consider the *value problem* that asks, given a WTG $\mathcal{G}$, a location ℓ and a threshold $\alpha \in \mathbb{Z} \cup \{-\infty, +\infty\}$, to decide whether $\mathsf{Val}_{\mathcal{G}}(\ell, \mathbf{0}) \leqslant \alpha$. In the context of timed games, optimal strategies may not exist. We generally focus on finding ε-optimal strategies, that guarantee the optimal value, up to a small error ε. Moreover, when the value problem is undecidable, we also consider the *approximation problem* that consists, given a precision $\varepsilon \in \mathbb{Q}_{>0}$, in computing an ε-approximation of $\mathsf{Val}_{\mathcal{G}}(\ell, \mathbf{0})$.

Related work. In the one-player case, computing the optimal value and an ε-optimal strategy for weighted timed automata is known to be PSPACE-complete [6]. In the two-player case, the value problem of WTGs (also called priced timed games in the literature) is undecidable with 3 clocks [12, 10], or even 2 clocks in the presence of negative weights [15] (for the existence problem asking if a strategy of player Min can guarantee a given threshold). To obtain decidability, one possibility is to limit the number of clocks to 1: then, there is an exponential-time algorithm to compute the value as well as ε-optimal strategies in the presence of non-negative weights only [7, 20, 17], whereas the problem is only known to be PTIME-hard. A similar result can be lifted to arbitrary weights, under restrictions on the resets of the clock in cycles [13].

The other possibility to obtain a decidability result [9, 16] is to enforce a semantical property of divergence (originally called strictly non-Zeno cost): it asks that every play following a cycle in the region automaton has weight far from 0. It allows the author to prove that playing for only a bounded number of steps is equivalent to the original game, which boils down to the problem of computing the value of a tree-shaped weighted timed game $\mathcal{G}$ using the value iteration algorithm.

Other objectives, not directly related to optimal reachability, have been considered in [11] for weighted timed games, like mean-payoff and parity objectives. In this work, the authors manage to solve these problems for the so-called class of δ-robust WTGs that they introduce. This class includes the class we consider, but is decidable in 2-EXPSPACE.

Our results. A cycle π of $\mathcal{R}(\mathcal{G})$ is said to be a positive cycle (resp. a 0-cycle, or a negative cycle) if every finite play ρ following π satisfies $\mathsf{wt}_\Sigma(\rho) \geqslant 1$ (resp. $\mathsf{wt}_\Sigma(\rho) = 0$, or $\mathsf{wt}_\Sigma(\rho) \leqslant -1$). A strongly connected component (SCC) S of $\mathcal{R}(\mathcal{G})$ is said to be positive (resp. negative) if every cycle $\pi \in S$ is positive (resp. negative). An SCC S of $\mathcal{R}(\mathcal{G})$ is said to be non-negative (resp. non-positive) if every play ρ following a cycle in S satisfies either $\mathsf{wt}_\Sigma(\rho) \geqslant 1$ or $\mathsf{wt}_\Sigma(\rho) = 0$ (resp. either $\mathsf{wt}_\Sigma(\rho) \leqslant -1$ or $\mathsf{wt}_\Sigma(\rho) = 0$).

▶ **Definition 3.** A WTG $\mathcal{G}$ is divergent (as defined in [16]) if every SCC of $\mathcal{R}(\mathcal{G})$ is either positive or negative. As a generalisation, a WTG $\mathcal{G}$ is almost-divergent when every SCC of $\mathcal{R}(\mathcal{G})$ is either non-negative or non-positive.

In [16], we showed that we can decide in 2-EXPTIME the value problem for divergent WTGs. Unfortunately, it is shown in [10] that this problem is undecidable for almost-divergent WTGs (already with non-negative weights only, where almost-divergent WTGs are called *simple*). They propose a solution to the approximation problem, again with non-negative weights only. Our main result is the following extension of their result:

▶ **Theorem 4.** *Given an almost-divergent WTG $\mathcal{G}$, a location ℓ and $\varepsilon \in \mathbb{Q}_{>0}$, we can compute an ε-approximation of $\mathsf{Val}_{\mathcal{G}}(\ell, \mathbf{0})$ in complexity doubly-exponential in the size of $\mathcal{G}$ and polynomial in $1/\varepsilon$. Moreover, deciding if a WTG is almost-divergent is* PSPACE*-complete.*

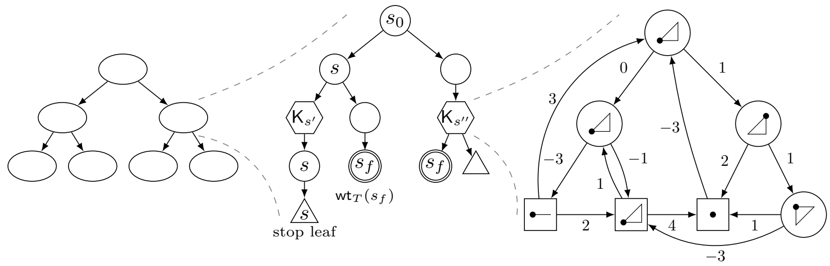


Figure 2 Static approximation schema: SCC decomposition of $\mathcal{R}(\mathcal{G})$, semi-unfolding of an SCC, corner-point abstraction for the kernels

To obtain this result, we follow an approximation schema that we now outline. First, we will always reason on the region game $\mathcal{R}(\mathcal{G})$ of the almost-divergent WTG $\mathcal{G}$. The goal is to compute an ε-approximation of $\mathsf{Val}_{\mathcal{R}(\mathcal{G})}(s_0, \mathbf{0})$ for some state $s_0 = (\ell_0, r_0)$, with r_0 the region where every clock value is 0. As already recalled, techniques of [1] allows one to compute the (exact) values of a WTG played on a finite tree, using operator $\mathcal{F}$. The idea is thus to decompose as much as possible the game $\mathcal{R}(\mathcal{G})$ in a WTG over a tree. First, we decompose the region game into SCCs (left of Figure 2).

During the approximation process, we must think about the final weight functions as the previously computed approximations of the values of SCCs below. We will keep as an invariant that final weight functions are piecewise linear functions with a finite number of pieces, and are continuous on each region.

For an SCC of $\mathcal{R}(\mathcal{G})$ and an initial state s_0 of $\mathcal{R}(\mathcal{G})$ provided by the SCC decomposition, we show that the game on the SCC is equivalent to a game on a tree built from a semi-unfolding (see middle of Figure 2) of $\mathcal{R}(\mathcal{G})$ from s_0 of finite depth, with certain nodes of the tree being *kernels*. These kernels are some parts of $\mathcal{R}(\mathcal{G})$ that contain all cycles of weight 0.

▶ Remark. In a weighted-timed game, it is easy to detect the set of states with value $+\infty$: these are all the states from which Min cannot ensure reachability of a target location $\ell \in L_T$ with $\mathsf{wt}_T(\ell) < +\infty$. It can therefore be computed by an attractor computation, and is indeed a property constant on each region. In particular, removing those states from $\mathcal{R}(\mathcal{G})$ does not affect the value of any other state and can be done in complexity linear in $|\mathcal{R}(\mathcal{G})|$. We will therefore assume that the considered WTG have no configurations with value $+\infty$.

3 Kernels of an almost-divergent WTG

The approximation procedure described before uses the so-called *kernels* in order to group together all cycles of weight 0. We study those kernels and give a characterisation allowing computability. Contrary to the non-negative case, the situation is more complex in our arbitrary case, since weights of both locations and transitions may differ from 0 in the kernel. Moreover, it is not trivial (and may not be true in a non almost-divergent WTG) to know whether it is sufficient to consider only simple cycles, i.e. cycles without repetitions.

To answer these questions, let us first analyse the cycles of $\mathcal{R}(\mathcal{G})$ that we will encounter. Since we are in an almost-divergent game, by Lemma 2, all cycles $\pi = t_1 \cdots t_n$ of $\mathcal{R}(\mathcal{G})$ (with $t_1, \dots, t_n$ transitions of $\mathcal{R}(\mathcal{G})$) are either 0-cycles, positive cycles or negative cycles. Additionally, in an SCC S of $\mathcal{R}(\mathcal{G})$, we cannot find both positive and negative cycles by definition. Moreover, we can classify a cycle by looking only at the corner plays following it.

▶ **Lemma 5.** *A cycle π is a 0-cycle iff there exists a corner play ρ following π with* $\mathsf{wt}_\Sigma(\rho)=0$.

An important result is that 0-cycles are stable by rotation. This is not trivial because plays following a cycle can start and end in different valuations, therefore changing the starting state of the cycle could a priori change the plays that follow it and their weights.

▶ **Lemma 6.** *Let π and π' be paths of $\mathcal{R}(\mathcal{G})$. Then, $\pi\pi'$ is a 0-cycle iff $\pi'\pi$ is a 0-cycle.*

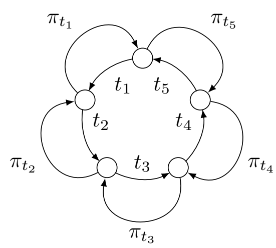
Sketch of proof. This stems from a pumping argument on the corner plays following cycles. Indeed, there are finitely many corners, so by constructing a long enough play following an iterate of $\pi'\pi$, we can obtain a corner play that starts and ends in the same corner. This play can then be considered as a play following an iterate of $\pi\pi'$, which ensures that it has weight 0. This allows us to conclude because in an almost-divergent WTG, if $(\pi\pi')^m$ is a 0-cycle then $\pi\pi'$ is a 0-cycle. ◀

We will now construct the kernel K as the subgraph of $\mathcal{R}(\mathcal{G})$ containing all 0-cycles. Formally, let T_{K} be the set of transitions of $\mathcal{R}(\mathcal{G})$ belonging to a *simple* 0-cycle, and S_{K} be the set of states covered by T_{K}. We define the kernel K of $\mathcal{R}(\mathcal{G})$ as the subgraph of $\mathcal{R}(\mathcal{G})$ defined by S_{K} and T_{K}. Transitions in $T \backslash T_{\mathsf{K}}$ with starting state in S_{K} are called the output transitions of K. We define it using only simple 0-cycles in order to ensure its computability. However, we now show that this is of no harm, since the kernel contains exactly all the 0-cycles, which will be crucial in the approximation schema we present in Section 5.

▶ **Proposition 7.** *A cycle of $\mathcal{R}(\mathcal{G})$ is entirely in K if and only if it is a 0-cycle.*

Proof. We prove that every 0-cycle is in K by induction on the length of the cycles. The initialisation contains only cycles of length 1, that are in K by construction. If we consider a cycle π of length $n > 1$, it is either simple or it can be rotated and decomposed into $\pi'\pi''$, π' and π'' being smaller cycles. Let ρ be a corner play following $\pi'\pi''$. We denote by ρ' the prefix of ρ following π' and ρ'' the suffix following π''. It holds that $\mathsf{wt}_\Sigma(\rho') = -\mathsf{wt}_\Sigma(\rho'')$, and in an almost-divergent SCC this implies $\mathsf{wt}_\Sigma(\rho') = \mathsf{wt}_\Sigma(\rho'') = 0$. Therefore, by Lemma 5 both π' and π'' are 0-cycles, and they must be in K by induction hypothesis. Note that this reasoning proves that every cycle contained in a longer 0-cycle is also a 0-cycle.

We now prove that every cycle in K is a 0-cycle. By construction, every transition $t \in T_{\mathsf{K}}$ is part of a simple 0-cycle. Thus, to every transition $t \in T_{\mathsf{K}}$, we can associate a path π_t such that $t\pi_t$ is a simple 0-cycle (rotate the simple cycle if necessary). We can prove the following property by relying on another pumping arguments on corners (see Lemma 17 in Appendix B): If $t_1 \cdots t_n$ is a path in K, then $t_1 t_2 \cdots t_n \pi_{t_n} \cdots \pi_{t_2} \pi_{t_1}$ is a 0-cycle of $\mathcal{R}(\mathcal{G})$. Now, if π is a cycle of $\mathcal{R}(\mathcal{G})$ in K, there exists a cycle π' such that $\pi\pi'$ is a 0-cycle, therefore π is a 0-cycle. ◀



4 Semi-unfolding of almost-divergent WTGs

Given an almost-divergent WTG $\mathcal{G}$, we describe the construction of its *semi-unfolding* $\mathcal{T}(\mathcal{G})$ (as depicted in Figure 2), which is a WTG that has the same value as $\mathcal{G}$. Moreover, the SCC-decomposition of $\mathcal{T}(\mathcal{G})$ is tree-shaped and each non-trivial SCC is a kernel. In the following, the depth of $\mathcal{T}(\mathcal{G})$ refers to the depth of its SCC-decomposition. This construction crucially relies on the absence of states with value $-\infty$, so we explain how to deal with them:

▶ **Lemma 8.** *In an SCC of $\mathcal{R}(\mathcal{G})$, the set of configurations with value $-\infty$ is a union of regions computable in time linear in the size of $\mathcal{R}(\mathcal{G})$.*

Sketch of proof. If the SCC is non-negative, the cumulated weight cannot decrease along a cycle, thus, the only way to obtain value $-\infty$ is to jump in a final state with final weight $-\infty$. We can therefore compute this set of states with an attractor for Min.

If the SCC is non-positive, we let $S_f^{\mathbb{R}}$ (resp. $S_f^{-\infty}$) be the set of target states where wt_T is bounded (resp. has value $-\infty$). We also define $T_f^{\mathbb{R}}$ (resp. $T_f^{-\infty}$), the set of transitions of $\mathcal{R}(\mathcal{G})$ whose end state belongs to $S_f^{\mathbb{R}}$ (resp. $S_f^{-\infty}$). Notice that the kernel cannot contain target states since they do not have outgoing transitions. We can prove that a configuration has value $-\infty$ iff it belongs to a state where player Min can ensure the LTL formula on transitions: $(\mathrm{G}\,\neg T_f^{\mathbb{R}} \wedge \neg \mathrm{FG}\, T_{\mathsf{K}}) \vee \mathrm{F}\, T_f^{-\infty}$. The procedure to detect $-\infty$ states thus consists of four attractor computations, which can be done in time linear in $|\mathcal{R}(\mathcal{G})|$. ◀

We can now assume that no states of $\mathcal{G}$ has value $-\infty$, and that the output weight function maps all configurations to $\mathbb{R}$. Since wt_T is piecewise linear with finitely many pieces, wt_T is bounded. Let $\sup|\mathsf{wt}_T|$ denote the bound of $|\mathsf{wt}_T|$, ranging over all target configurations. We turn to the construction of the semi-unfolding $\mathcal{T}(\mathcal{G})$ and prove:

▶ **Proposition 9.** *Let $\mathcal{G}$ be an almost-divergent WTG with initial state (ℓ_0, r_0). There exists a semi-unfolding $\mathcal{T}(\mathcal{G})$ with initial state $(\tilde{\ell}_0, r_0)$ such that for all $\nu_0 \in r_0$, $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0) = \mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$. The depth of $\mathcal{T}(\mathcal{G})$ is polynomial in $|\mathcal{R}(\mathcal{G})|$, $w^e_{\max}$ and* $\sup|\mathsf{wt}_T|$.

Sketch of proof. We only build the semi-unfolded game $\mathcal{T}(\mathcal{G})$ of an SCC of $\mathcal{G}$ starting from some initial state (ℓ_0, r_0), since it is then easy to glue all the semi-unfoldings together to get the one of the full game. Since every configuration has finite value, we can prove that values of the game are bounded by $|\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T|$. As a consequence, we can find a bound γ linear in $|\mathcal{R}(\mathcal{G})|$, $w^e_{\max}$ and $\sup|\mathsf{wt}_T|$ such that a play that visits some state outside the kernel more than γ times has weight strictly above $|\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T|$, hence is useless for value computation. This leads to considering the semi-unfolding $\mathcal{T}(\mathcal{G})$ of $\mathcal{G}$ (nodes in the kernel are not unfolded, see Figure 2) such that each node not in the kernel is encountered at most γ times along a branch. In particular, the depth of $\mathcal{T}(\mathcal{G})$ is bounded by $|\mathcal{R}(\mathcal{G})|\gamma$. ◀

5 Approximation of almost-divergent WTGs

Approximation of kernels. We start by approximating a kernel $\mathcal{G}$ by extending the region-based approximation schema of [10]. In their setting, all runs in kernels had weight 0, allowing a simple reduction to a finite weighted game. In our setting, we have to approximate the timed dynamics of runs, and therefore resort to the corner-point abstraction (as shown in the right of Figure 2).

Since output weight functions are piecewise linear with a finite number of pieces and continuous on regions, they are K-Lipschitz-continuous[3], for a given constant $K \geqslant 0$. We let $\mathbf{B} = w^L_{\max}\,|L||\mathsf{Reg}(X, M)| + K$.

Let N be an integer. Consider the game $\mathcal{C}_N(\mathcal{G})$ described in the preliminary section, with locations of the form (ℓ, r, v) with v a corner of the $1/N$-region r. Two plays ρ of $\mathcal{G}$ and ρ' of $\mathcal{C}_N(\mathcal{G})$ are said to be $1/N$-*close* if they follow the same path π in $\mathcal{R}_N(\mathcal{G})$. In particular, at each step the configurations (ℓ, ν) in ρ and (ℓ', r', v') in ρ' (with v' a corner of the $1/N$-region r') satisfy $\ell = \ell'$ and $\nu \in r'$, and the transitions taken in both plays have the same discrete weights. Close plays have *close* weights, in the following sense:

[3] The function wt_T is said to be K-Lipschitz-continuous when $|\mathsf{wt}_T(s,\nu) - \mathsf{wt}_T(s,\nu')| \leqslant K\|\nu - \nu'\|_\infty$ for all valuations ν, ν', where $\|v\|_\infty = \max_{x \in X} |v(x)|$ is the ∞-norm of vector $v \in \mathbb{R}^X$. The function wt_T is said to be Lipschitz-continuous if it is K-Lipschitz-continuous, for some K.

▶ **Lemma 10.** *For all $1/N$-close plays ρ of $\mathcal{G}$ and ρ' of $\mathcal{C}_N(\mathcal{G})$, $|\mathsf{wt}_{\mathcal{G}}(\rho)-\mathsf{wt}_{\mathcal{C}_N(\mathcal{G})}(\rho')| \leqslant \mathbf{B}/N$.*

In particular, if we start in configurations (ℓ_0, ν_0) of $\mathcal{G}$, and $((\ell_0, r_0, v_0), v_0)$ of $\mathcal{C}_N(\mathcal{G})$, with $\nu_0 \in r_0$, since both players have the ability to stay $1/N$-close all along the plays, a bisimulation argument permits to obtain that the values of the two games are also close in (ℓ_0, ν_0) and $((\ell_0, r_0, v_0), v_0)$:

▶ **Lemma 11.** *For all locations $\ell \in L$, $1/N$-regions r, $\nu \in r$ and corners v of r, $|\mathsf{Val}_{\mathcal{G}}(\ell,\nu) - \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell, r, v), v)| \leqslant \mathbf{B}/N$.*

Using this result, picking N an integer larger than $\mathbf{B}/\varepsilon$, we can thus obtain $|\mathsf{Val}_{\mathcal{G}}(\ell,\nu) - \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell, r, v), v)| \leqslant \varepsilon$. Recall that $\mathcal{C}_N(\mathcal{G})$ can be considered as an untimed weighted game (with reachability objective). Thus we can apply the result of [14], where it is shown that the optimal values of such games can be computed in pseudo-polynomial time (i.e. polynomial time with weights encoded in unary, instead of binary). We then define an ε-approximation of $\mathsf{Val}_{\mathcal{G}}$, named Val'_N, on each $1/N$-region by interpolating the values of its $1/N$-corners in $\mathcal{C}_N(\mathcal{G})$ with a piecewise linear function: therefore, we can control the Lipschitz constant of the approximated value for further use.

▶ **Lemma 12.** Val'_N *is an ε-approximation of* $\mathsf{Val}_{\mathcal{G}}$*, that is piecewise linear with a finite number of pieces and $2\mathbf{B}$-Lipschitz-continuous over regions.*

Approximation of almost-divergent WTGs. We now explain how to approximate the value of an almost-divergent WTG $\mathcal{G}$, thus proving Theorem 4. First, we compute a semi-unfolding $\mathcal{T}(\mathcal{G})$ as described in the previous section. Then we perform a bottom-up computation of the approximation. As already recalled, techniques of [1] allow us to compute exact values of a tree-shape WTG. In consequence, we know how to compute the value of a non-kernel node of $\mathcal{T}(\mathcal{G})$, depending of the values of its children. There is no approximation needed here, so that if all children are ε-approximation, we can compute an ε-approximation of the node. Therefore, the only approximation lies in the kernels, and we explained before how to compute arbitrarily close an approximation of a kernel's value. We crucially rely on the fact that the value function is 1-Lipschitz-continuous[4]. This entails that imprecisions will sum up along the bottom-up computations, as computing an ε-approximation of the value of a game whose output weights are ε'-approximations yields an $(\varepsilon+\varepsilon')$-approximation. Therefore we compute approximations with threshold $\varepsilon' = \varepsilon/\alpha$ for kernels in $\mathcal{T}(\mathcal{G})$, where α is the maximal number of kernels along a branch of $\mathcal{T}(\mathcal{G})$: α is smaller than the depth of $\mathcal{T}(\mathcal{G})$, which is bounded by Proposition 9.

The subregion granularity considered before for kernel approximation crucially depends on the Lipschitz constant of output weights. The growth of these constants is bounded for kernels in $\mathcal{T}(\mathcal{G})$ by Lemma 12. For non-kernel nodes of $\mathcal{T}(\mathcal{G})$, using a careful analysis of the algorithm of [1] (see details in Appendix D.2), we obtain the following bound:

▶ **Lemma 13.** *If all the output weights of a WTG $\mathcal{G}$ are K-Lipschitz-continuous over regions (and piecewise linear, with finitely many pieces), then $\mathsf{Val}^i_{\mathcal{G}}$ is KK'-Lipschitz-continuous over regions, with K' polynomial in $w^L_{\max}$ and $|X|$ and exponential in i.*

The overall time complexity of this method is doubly-exponential in the size of the input game and polynomial in $1/\varepsilon$. An example of execution of the approximation scheme can be found in Appendix D.3, and its complexity is analyzed in Appendix D.4.

[4] Indeed, inf and sup are 1-Lipschitz-continuous functions, and with a fixed play ρ, the mapping $\mathsf{wt}_T \to \mathsf{wt}_\Sigma(\rho) + \mathsf{wt}_T(\mathsf{last}(\rho))$ is 1-Lipschitz-continuous.

6 Symbolic approximation algorithm

The previous approximation result suffers from several drawbacks: it relies on the SCC decomposition of the region automaton, which have to be analysed in a sequential way, and their analysis requires an a priori refinement of the granularity of regions. This approach is thus not easily amenable to implementation. We instead prove in this section that the symbolic approach based on the value iteration paradigm, i.e. the computation of iterates of the operator $\mathcal{F}$ recalled in page 6, is an approximation scheme:

▶ **Theorem 14.** *Let $\mathcal{G}$ be an almost-divergent WTG such that $\mathsf{Val}_{\mathcal{G}} > -\infty$ for all configurations. Then the sequence $(\mathsf{Val}^k_{\mathcal{G}})_{k\geqslant 0}$ converges towards $\mathsf{Val}_{\mathcal{G}}$ and for every $\varepsilon \in \mathbb{Q}_{>0}$, we can compute an integer P such that $\mathsf{Val}^P_{\mathcal{G}}$ is an ε-approximation of $\mathsf{Val}_{\mathcal{G}}$ for all configurations.*

Sketch of proof. The proof relies on the semi-unfolding considered in the previous approximation scheme and on the following arguments:

1. For a kernel, one can bound the number of computation steps of value iteration that are needed to achieve a given precision. This number depends on the Lipschitz constants of the functions given as output weights.
2. When applying value iteration, one can bound how the Lipschitz constant of the value function increases after a bounded number of steps.
3. As the operator $\mathcal{F}$ is 1-Lipschitz-continuous, imprecisions will sum up along the way.

The only new property is the first one, and it can be derived from the $1/N$ corner-point abstraction techniques developed in Section 5. Then, we can use all three properties to prove that a semi-unfolding can be approximated by an unfolding (without kernels) of the game that mirrors the computation of Val^P, and conclude. ◀

This symbolic procedure avoids the three drawbacks (SCC decomposition, sequential analysis of the SCCs, and refinement of the granularity of regions) of the previous approximation scheme. Moreover, it allows one to easily obtain an almost-optimal strategy w.r.t. the computed value. Its proof relies on Section 5, and would not hold with the approximation scheme of [10] (that does not maintain the continuity on regions of the computed value functions, in turn needed to define output weights on $1/N$-corners). If one has the guarantee that no configurations of $\mathcal{G}$ have value $-\infty$, then one can directly apply the value iteration approach. If this is not the case, then one can perform the SCC decomposition of $\mathcal{R}(\mathcal{G})$, and, as $\mathcal{G}$ is almost-divergent, identify and remove regions whose value is $-\infty$, by Lemma 8.

7 Conclusion

We have given an approximation procedure for a large class of weighted timed games with unbounded number of clocks and arbitrary integer weights that can be executed in doubly-exponential time with respect to the size of the game. In addition, we proved the correction of a symbolic approximation scheme, that does not start by splitting exponentially every region, but only does so when necessary (as dictated by [1]). We argue that this paves the way towards an implementation of value approximation for weighted timed games.

Another perspective is to extend this work to the concurrent setting, where both players play simultaneously and the shortest delay is selected. We did not consider this setting in this work because concurrent WTGs are not determined, and several of our proofs rely on this property for symmetrical arguments (mainly to lift results of non-negative SCCs to non-positive ones). Another extension is the exploration of the effect of almost-divergence in the case of multiple weight dimensions, and/or with mean-payoff objectives.

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.
2. Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
3. Rajeev Alur, Salvatore La Torre, and George J. Pappas. Optimal paths in weighted timed automata. *Theoretical Computer Science*, 318(3):297–322, 2004.
4. 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.
5. Gerd Behrmann, Ansgar Fehnker, Thomas Hune, Kim G. Larsen, Judi Romijn, and Frits W. Vaandrager. Minimum-cost reachability for priced timed automata. In *Proceedings of the 4th International Workshop on Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *LNCS*, pages 147–161. Springer, 2001.
6. Patricia Bouyer, Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On the optimal reachability problem of weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, 2007.
7. Patricia Bouyer, Thomas Brihaye, and Nicolas Markey. Improved undecidability results on weighted timed automata. *Information Processing Letters*, 98(5):188–194, 2006.
8. Patricia Bouyer, Ed Brinksma, and Kim G. Larsen. Optimal infinite scheduling for multi-priced timed automata. *Formal Methods in System Design*, 32(1):3–23, 2008.
9. Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. Optimal strategies in priced timed game automata. In *Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'04)*, volume 3328 of *LNCS*, pages 148–160. Springer, 2004.
10. 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`.
11. Romain Brenguier, Franck Cassez, and Jean-François Raskin. Energy and mean-payoff timed games. In *17th International Conference on Hybrid Systems: Computation and Control (part of CPS Week), HSCC'14, Berlin, Germany, April 15-17, 2014*, pages 283–292. ACM, 2014.
12. Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On optimal timed strategies. In *Proceedings of the Third international conference on Formal Modeling and Analysis of Timed Systems (FORMATS'05)*, volume 3829 of *LNCS*, pages 49–64. Springer, 2005.
13. Thomas Brihaye, Gilles Geeraerts, Axel Haddad, Engel Lefaucheux, 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.
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*, 2016. `doi:10.1007/s00236-016-0276-z`.
15. Thomas Brihaye, Gilles Geeraerts, Shankara Narayanan Krishna, Lakshmi Manasa, Benjamin Monmege, and Ashutosh Trivedi. Adding negative prices to priced timed games. In *Proceedings of the 25th International Conference on Concurrency Theory (CONCUR'14)*, volume 8704, pages 560–575. Springer, 2014. `doi:10.1007/978-3-662-44584-6_38`.

16 Damien Busatto-Gaston, Benjamin Monmege, and Pierre-Alain Reynier. Optimal reachability in divergent weighted timed games. In Javier Esparza and Andrzej S. Murawski, editors, *Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'17)*, volume 10203 of *Lecture Notes in Computer Science*, pages 162–178, Uppsala, Sweden, April 2017. Springer. doi:10.1007/978-3-662-54458-7_10.

17 Thomas Dueholm Hansen, Rasmus Ibsen-Jensen, and Peter Bro Miltersen. A faster algorithm for solving one-clock priced timed games. In *Proceedings of the 24th International Conference on Concurrency Theory (CONCUR'13)*, volume 8052 of *LNCS*, pages 531–545. Springer, 2013.

18 Neil Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17:935–938, 1988.

19 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.

20 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.

21 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

22 Róbert Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26(3):279–284, 1988.

A Proofs of Section 2

Proof of Lemma 2. The set $\{\mathsf{wt}_\Sigma(\rho) \mid \rho$ finite play following $\pi\}$ is an interval as the image of a convex set by an linear function (see [6, Sec. 3.2] for an explanation). The good properties of the corner-point abstraction allows us to conclude, since for every play ρ following π, one can find a corner play following π of smaller weight and one of larger weight, and for every corner play ρ following π and every $\varepsilon > 0$, one can find a play following π whose weight is at most ε away from $\mathsf{wt}_\Sigma(\rho)$ [8]. ◀

A.1 Undecidability of value $-\infty$

We prove that given a WTG $\mathcal{G}$ (not necessarily almost-divergent) and an initial location ℓ_0, it is undecidable whether $\mathsf{Val}_\mathcal{G}(\ell_0, \mathbf{0}) = -\infty$. We reduce it to the existence problem on turn-based WTG: given a WTG $\mathcal{G}$ (without output weight function), an integer threshold α and a starting location ℓ_0, does there exist a strategy for Min that can guarantee reaching the unique target location ℓ_t from ℓ_0 with weight $< \alpha$. In the non-negative setting, it is proved in [7] that the problem is undecidable for the comparison $\leqslant \alpha$. In the negative setting, formal proofs are given for all comparison signs in [15].

Consider $\mathcal{G}'$ the WTG built from $\mathcal{G}$ by adding a transition from ℓ_t to ℓ_0, without guards and resetting all the clocks, of discrete weight $-\alpha$. We add a new target location ℓ'_t, and add transitions of weight 0 from ℓ_t to ℓ'_t. Location ℓ_t is then given to Min. Let us prove that $\mathsf{Val}_{\mathcal{G}'}(\ell_0, \mathbf{0}) = -\infty$ if and only if Min has a strategy to guarantee a weight $< \alpha$ in $\mathcal{G}$.

Assume first $\mathsf{Val}_{\mathcal{G}'}(\ell_0, \mathbf{0}) = -\infty$. If $\mathsf{Val}_\mathcal{G}(\ell_0, \mathbf{0}) = -\infty$, we are done. Otherwise, Min must follow in $\mathcal{G}'$ the new transition from ℓ_t to ℓ_0 to enforce a cycle of negative value, and thus enforce a play from $(\ell_0, \mathbf{0})$ to ℓ_t with weight less than α. Therefore, there exists a strategy for Min in $\mathcal{G}$ that can guarantee a weight $< \alpha$.

Reciprocally, if there exists a strategy for Min that can guarantee a weight $< \alpha$, then Min can force a negative cycle play and $\mathsf{Val}_{\mathcal{G}'}(\ell_0, \mathbf{0}) = -\infty$.

A.2 Decision of the almost-divergence of a WTG

First, we state that a WTG $\mathcal{G}$ is not almost-divergent if and only if $\mathcal{R}(\mathcal{G})$ contains an SCC with either both a positive play following one of its cycles and a negative play following one of its cycles, or a play with weight in $(-1, 0) \cup (0, 1)$ following one of its cycles. We will now explain how we can test both of those properties (and thus if a game is not almost-divergent) in PSPACE.

A corner play following a cycle of the region game is said to be simple if it does not visit the same corner twice (but the first and last corners can be the same). A simple corner play following a cycle has length bounded by $|S| \times (|X| + 1)$. By Lemma 2, $\mathcal{R}(\mathcal{G})$ contains an SCC with either both a positive play following one of its cycles and a negative play following one of its cycles if and only if $\mathcal{R}(\mathcal{G})$ contains both a positive corner play following one of its cycles and a negative corner play following one of its cycles. We will extend this to simple corner plays.

▶ **Lemma 15.** *$\mathcal{R}(\mathcal{G})$ contains an SCC with either both a positive play following one of its cycles and a negative play following one of its cycles if and only if $\mathcal{R}(\mathcal{G})$ contains an SCC with both a positive simple corner play following one of its cycles and a negative simple corner play following one of its cycles.*

Proof. All that is left to prove is that, in an SCC of $\mathcal{R}(\mathcal{G})$, if all simple corner plays following a cycle have non-negative weight (resp. non-positive weight), then all corner plays following a cycle have non-negative weight (resp. non-positive weight).

By contradiction, we consider ρ, the shortest corner play following a cycle π, such that $\mathsf{wt}_\Sigma(\rho) < 0$ (resp. $\mathsf{wt}_\Sigma(\rho) > 0$). Corner play ρ cannot be simple, so it must contain a simple loop. That loop is a simple corner play following a cycle of $\mathcal{R}(\mathcal{G})$, so it must have non-negative weight (resp. non-positive weight). This means that ρ without that loop satisfies $\mathsf{wt}_\Sigma(\rho) < 0$ (resp. $\mathsf{wt}_\Sigma(\rho) > 0$), and therefore was not the shortest corner play with the desired property. ◀

We can test the existence of such simple corner plays in a SCC of $\mathcal{R}(\mathcal{G})$ in NPSPACE, by guessing them corner after corner and by keeping the cumulated weight in memory. The check that both plays are in the same SCC is a reachability check in a timed automaton, which can be done in PSPACE. We described a similar procedure in [16] where we were testing the existence of a non-negative corner play and a non-positive one in the same SCC instead of a negative one and a positive one.

Now, we will assume in this second part that this test failed, so every SCC of $\mathcal{R}(\mathcal{G})$ either satisfies that all plays following a cycle have non-negative weight or satisfies that they all have non-positive weight. We will now explain how to check if $\mathcal{R}(\mathcal{G})$ contains a play with weight in $(-1,0) \cup (0,1)$ following one of its cycles. Let $B = (|S| \times (|X|+1))^2$.

▶ **Lemma 16.** *$\mathcal{R}(\mathcal{G})$ contains a play with weight in $(-1,0) \cup (0,1)$ following one of its cycles if and only if $\mathcal{R}(\mathcal{G})$ contains a cycle π of length at most B such that there is a corner play following π with weight zero and another one with non-zero weight.*

Proof. By Lemma 2, $\mathcal{R}(\mathcal{G})$ contains a play with weight in $(-1,0) \cup (0,1)$ following one of its cycles if and only if that cycle satisfies that there is a corner play following it with weight zero and another one with non-zero weight.

We only need to show that if there are no such cycles of length at most B, then there are no such cycles of any length. Therefore, we assume that no cycle of length less than B allows a play with weight in $(-1,0) \cup (0,1)$. By contradiction, let π be the shortest cycle such that there exist two corner plays ρ and ρ' following π, with $\mathsf{wt}_\Sigma(\rho) = 0$ and $\mathsf{wt}_\Sigma(\rho') \neq 0$. Then $|\pi| > B$. Let v_i be the i-th corner of ρ, and v'_i be the i-th corner of ρ'. There are at most $(|S| \times (|X|+1))^2$ different pairs (v_i, v'_i), which implies that there must be two indexes, j and k, such that $(v_j, v'_j) = (v_k, v'_k)$ and $j < k$. The portion of ρ between indexes j and k follows a cycle, and have opposite weight to the play constructed by considering ρ and removing the loop between indexes j and k. Since the sum of their weight is 0 and they both follow cycles of $\mathcal{R}(\mathcal{G})$ in the same SCC, both of those plays have weight 0. The portion of π between indexes j and k is a cycle shorter than π, and it contains a corner play of weight 0, therefore all of its corner plays have weight 0, and the portion of ρ' between indexes j and k has weight 0 too. But then the cycle defined by taking π and removing the loop between indexes j and k contains a corner play of weight 0 (derived from ρ), and a corner play of weight non-zero (derived from ρ'), and that contradicts π being the shortest cycle with that property. ◀

Once again, we can check the existence of such a cycle of length bounded by B in NPSPACE by guessing it and its two relevant corner plays on-the-fly and storing the cumulated weight of each. This imply that deciding if a game $\mathcal{G}$ is almost divergent is decidable in coNPSPACE = NPSPACE = PSPACE (using the theorems of Immerman-Szelepcsényi [18, 22] and Savitch [21]).

Let us now show the PSPACE-hardness (indeed the coPSPACE, which is identical) by a reduction from the reachability problem in a timed automaton. We consider a timed automaton with a starting state and a different target state without outgoing transitions. We construct from it a weighted timed game by distributing all states to Min, and equipping all transitions with weight 0, and all states with weight 0. We also add a loop with weight 1 on the initial state, one with weight -1 on the target state, and a transition from the target state to the initial state with weight 0, all three resetting all clocks and with no guard. Then, the weighted timed game is not almost-divergent if and only if the target can be reached from the initial state in the timed automaton.

B Proofs of the kernel characterisation (Section 3)

Proof of Lemma 5. If π is a 0-cycle, every such corner play ρ will have weight 0, by Lemma 2. Reciprocally, if such a corner play exists, all corner plays following π have weight 0, otherwise the set $\{\mathsf{wt}_\Sigma(\rho) \mid \rho \text{ play following } \pi\}$ would have non-empty intersection with the set $(-1,1)\setminus\{0\}$ which would contradict the almost-divergence. ◀

Proof of Lemma 6. First, notice that since $\pi_1 = \pi\pi'$ is a cycle, $\mathsf{first}(\pi) = \mathsf{last}(\pi')$ and $\mathsf{first}(\pi') = \mathsf{last}(\pi)$, so $\pi_2 = \pi'\pi$ is correctly defined. Then, let us define two sequences of region corners $(v_i \in \mathsf{first}(\pi))_i$ and $(v'_i \in \mathsf{first}(\pi'))_i$. We start by choosing any $v_0 \in \mathsf{first}(\pi)$. Let v'_0 be a corner of $\mathsf{first}(\pi')$ such that v'_0 is accessible from v_0 by following π. For every $i > 0$, let v_i be a corner of $\mathsf{first}(\pi)$ such that v_i is accessible from v'_{i-1} by following π', and let v'_i be a corner of $\mathsf{first}(\pi')$ such that v'_i is accessible from v_i by following π. We stop the construction at the first l such that there exists $k < l$ with $v_k = v_l$. Additionally, we let $v'_l = v'_k$ and $v_{l+1} = v_{k+1}$. This process is bounded since $\mathsf{first}(\pi)$ has at most $|X|+1$ corners.

For every $0 \leqslant i \leqslant l$, let w_i be the weight of a play ρ_i from v_i to v'_i along π, and let w'_i be the weight of a play ρ'_i from v'_i to v_{i+1} along π'. The concatenation of the two plays has weight $w_i + w'_i = 0$, since it follows the 0-cycle π_1. Therefore, all corner plays from v_i to v'_i following π have the same weight w_i, and the same applies for w'_i. For every $0 \leqslant i < l$, the concatenation of ρ'_i and ρ_{i+1} is a play from v'_i to v_{i+1}, of weight $w'_i + w_{i+1} = -w_i + w_{i+1}$, following π_2. Since π_2 is a cycle, and the game is almost-divergent, all possible values of $w_{i+1} - w_i$ have the same sign.

Finally, we can construct a corner play from v'_k to v'_l by concatenating the plays $\rho'_k, \rho_{k+1}, \rho'_{k+1}, \rho_{k+2}, \dots, \rho'_{l-1}, \rho_l$. That play has weight $\sum_{i=k}^{l-1}(w_{i+1} - w_i) = w_l - w_k = 0$. This implies that the terms $w_{i+1} - w_i$, of constant sign, are all equal to 0. As a consequence, the concatenation of ρ'_k and ρ_{k+1} is a corner play following π_2 of weight 0. By Lemma 5, we deduce that π_2 is a 0-cycle. ◀

▶ **Lemma 17.** *If $t_1 \cdots t_n$ is a path in K, then $t_1 t_2 \cdots t_n \pi_{t_n} \cdots \pi_{t_2} \pi_{t_1}$ is a 0-cycle of $\mathcal{R}(\mathcal{G})$.*

Proof. We prove the property by induction on n. For $n = 1$, the property is immediate since $t_1 \pi_{t_1}$ is a 0-cycle. Consider then n such that the property holds for n, and prove it for $n+1$. We will exhibit two corner plays following $t_1 \cdots t_{n+1} \pi_{t_{n+1}} \cdots \pi_{t_1}$ of opposite weight and conclude with Lemma 5.

Let v_0 be a corner of $\mathsf{last}(t_{n+1})$. Since $t_{n+1}\pi_{t_{n+1}}$ is a 0-cycle, there exists $w \in \mathbb{Z}$, a corner play ρ_0 following t_{n+1} ending in v_0 with weight w and a corner play ρ'_0 following $\pi_{t_{n+1}}$ beginning in v_0 with weight $-w$. We name v'_0 the corner of $\mathsf{last}(t_n)$ where ends ρ'_0. We consider any corner play ρ_1 following t_{n+1} from corner v'_0. The corner play $\rho'_0\rho_1$ follows the path $\pi_{t_{n+1}} t_{n+1}$ that is also a 0-cycle by Lemma 6, therefore ρ_1 has weight w. We denote by v_1 the corner where ends ρ_1. By iterating this construction, we obtain some corner plays

$\rho_0, \rho_1, \rho_2, \ldots$ following t_{n+1} and $\rho'_0, \rho'_1, \rho'_2, \ldots$ following $\pi_{t_{n+1}}$ such that ρ'_i goes from corner v_i to v'_i, and ρ_{i+1} from corner v'_i to v_{i+1}, for all $i \geqslant 0$. Moreover, all corner plays ρ_i have weight w and all corner plays ρ'_i have weight $-w$. Consider the first index l such that $v_l = v_k$ for some $k < l$, which exists because the number of corners is finite.

We apply the induction to find a corner play following $t_1 \cdots t_n \pi_{t_n} \cdots \pi_{t_1}$, going through the corner v'_k in the middle: more formally, there exists w_α, a corner play ρ_α following $t_1 \cdots t_n$ ending in v'_k with weight w_α and a corner play ρ'_α following $\pi_{t_n} \cdots \pi_{t_1}$ beginning in v'_k with weight $-w_\alpha$. We apply the induction a second time with corner v'_{l-1}: there exists w_β, a corner play ρ_β following $t_1 \cdots t_n$ ending in v'_{l-1} with weight w_β and a corner play ρ'_β following $\pi_{t_n} \cdots \pi_{t_1}$ beginning in v'_{l-1} with weight $-w_\beta$.

The corner play $\rho_\alpha \rho_{k+1} \rho'_{k+1} \rho_{k+2} \rho'_{k+2} \cdots \rho'_{l-1} \rho'_\beta$, of weight $w_\alpha + (w - w)^{l-k} - w_\beta = w_\alpha - w_\beta$, follows the cycle $t_1 \cdots t_n (t_{n+1} \pi_{t_{n+1}})^{l-k} \pi_{t_n} \cdots \pi_{t_1}$. The corner play $\rho_\beta \rho_l \rho'_k \rho'_\alpha$, of weight $w_\beta + w - w - w_\alpha = w_\beta - w_\alpha$, follows the cycle $t_1 \cdots t_n t_{n+1} \pi_{t_{n+1}} \pi_{t_n} \cdots \pi_{t_1}$. Since the game is almost-divergent, and those two corner plays are in the same SCC, both have weight 0. The second corner play of weight 0 ensures that the cycle $t_1 \cdots t_{n+1} \pi_{t_{n+1}} \cdots \pi_{t_1}$ is a 0-cycle, by Lemma 5. ◀

C Proofs of the semi-unfolding (Section 4)

Proof of Lemma 8. We detail the case of non-negative SCCs. Let us prove that a configurations has value $-\infty$ if and only if it belongs to a state where player Min can ensure the LTL formula on transitions: $\phi = (\mathrm{G}(\neg T_f^{\mathbb{R}}) \wedge \neg \mathrm{FG} T_{\mathsf{K}}) \vee \mathrm{F} T_f^{-\infty}$. Since ω-regular games are determined, this is equivalent to saying that a configuration has finite value if and only if it belongs to a state where Max can ensure $\neg\phi$.

If s is a state where Min can ensure ϕ, he can ensure $-\infty$ value from all configurations in s by either reaching $S_f^{-\infty}$ or avoiding $S_f^{\mathbb{R}}$ for as long as he desires, while not getting stuck in K, and thus going through an infinite number of negative cycles by Proposition 7. This proves that a state where Max cannot ensure $\neg\phi$ contains only valuations of value $-\infty$. Conversely, if s is a state where Max can ensure $\neg\phi = (\mathrm{F} T_f^{\mathbb{R}} \vee \mathrm{FG} T_{\mathsf{K}}) \wedge \mathrm{G} \neg T_f^{-\infty}$, then from s, Max must be able to avoid $S_f^{-\infty}$, and eventually enforce either $S_f^{\mathbb{R}}$ reachability or staying in K forever. In both cases, Max can ensure a value above $-\infty$. ◀

C.1 Semi-unfolding construction

In order to prove Proposition 9, we will construct the desired semi-unfolding $\mathcal{T}(\mathcal{G})$ of a (non-negative or non-positive SCC) $\mathcal{G}$.

If (ℓ, r) is in K, we let $\mathsf{K}_{\ell,r}$ be the part of K accessible from (ℓ, r) (note that $\mathsf{K}_{\ell,r}$ is an SCC as K is a disjoint set of SCCs). We define the output transitions of $\mathsf{K}_{\ell,r}$ as being the output transitions of K accessible from (ℓ, r). If (ℓ, r) is not in K, the output transitions of (ℓ, r) are the transitions of $\mathcal{R}(\mathcal{G})$ starting in (ℓ, r).

Formally, we define a tree T whose nodes will either be labelled by region graph states $(\ell, r) \in S \backslash S_{\mathsf{K}}$ or by kernels $\mathsf{K}_{\ell,r}$, and whose edges will be labelled by output transitions in $\mathcal{R}(\mathcal{G})$. The root of the tree T is labelled with (ℓ_0, r_0), or K_{ℓ_0,r_0} (if (ℓ_0, r_0) belongs to the kernel), and the successors of a node of T are then recursively defined by its output transitions. When a state (ℓ, r) is reached by an output transition, the child is labelled by $\mathsf{K}_{\ell,r}$ if $(\ell, r) \in \mathsf{K}$, otherwise it is labelled by (ℓ, r). Edges in T are labelled by the transitions used to create them. Along every branch, we stop the construction when either a final state is reached (i.e. a state not inside the current SCC) or the branch contains $3|\mathcal{R}(\mathcal{G})|w^e_{\max} + 2 \sup |\mathsf{wt}_T| + 2$ nodes labelled by the same state ((ℓ, r) or $\mathsf{K}_{\ell,r}$). Since $\mathcal{R}(\mathcal{G})$

has a finite number of states, T is finite. Leaves of T with a location belonging to L_T are called *target leaves*, others are called *stopped leaves*.

We now transform T into a WTG $\mathcal{T}(\mathcal{G})$, by replacing every node labelled by a state (ℓ, r) by a different copy $(\tilde{\ell}, r)$ of (ℓ, r). Those states are said to inherit from (ℓ, r). Edges of T are replaced by the transitions labelling them, and have a similar notion of inheritance. Every non-leaf node labelled by a kernel $\mathsf{K}_{\ell,r}$ is replaced by a copy of the WTG $\mathsf{K}_{\ell,r}$, output transitions being plugged in the expected way. We deal with stopped leaves labelled by a kernel $\mathsf{K}_{\ell,r}$ by replacing them with a single node copy of (ℓ, r), like we dealt with node labelled by a state (ℓ, r). State partition between players and weights are inherited from the copied states of $\mathcal{R}(\mathcal{G})$. The only initial state of $\mathcal{T}(\mathcal{G})$ is the state denoted by $(\tilde{\ell}_0, r_0)$ inherited from (ℓ_0, r_0) in the root of T (either (ℓ_0, r_0) or K_{ℓ_0,r_0}). The final states of $\mathcal{T}(\mathcal{G})$ are the states derived from leaves of T. If $\mathcal{R}(\mathcal{G})$ is a non-negative (resp. non-positive) SCC, the output weight function wt_T is inherited from $\mathcal{R}(\mathcal{G})$ on target leaves and set to $+\infty$ (resp. $-\infty$) on stopped leaves.

C.2 Semi-unfolding correction

We will now prove that Proposition 9 holds on this $\mathcal{T}(\mathcal{G})$.

▶ **Lemma 18.** *All finite plays in $\mathcal{R}(\mathcal{G})$ have cumulated weight (ignoring output weights) at least $-|\mathcal{R}(\mathcal{G})|w^e_{\max}$ in the non-negative case, and at most $|\mathcal{R}(\mathcal{G})|w^e_{\max}$ in the non-positive case. Moreover, values of the game are bounded by $|\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T|$.*

Proof. Suppose first that $\mathcal{R}(\mathcal{G})$ is a non-negative SCC. Consider a play ρ following a path π. π can be decomposed into $\pi = \pi_1\pi_1^c \cdots \pi_n\pi_n^c$ such that every π_i^c is a cycle, and $\pi_1 \dots \pi_n$ is a simple path in $\mathcal{R}(\mathcal{G})$ (thus $\sum_{i=1}^n |\pi_i| \leqslant |\mathcal{R}(\mathcal{G})|$). Let us define all plays ρ_i and ρ_i^c as the restrictions of ρ on π_i and π_i^c. Now, since all plays following cycles have cumulated weight at least 0, $\mathsf{wt}_\Sigma(\rho) = \sum_{i=1}^n \mathsf{wt}_\Sigma(\rho_i) + \mathsf{wt}_\Sigma(\rho_i^c) \geqslant \sum_{i=1}^n -w^e_{\max}|\rho_i| + 0 \geqslant -|\mathcal{R}(\mathcal{G})|w^e_{\max}$. Similarly, we can show that every play in a non-positive SCC has cumulated weight at most $|\mathcal{R}(\mathcal{G})|w^e_{\max}$.

For the bound on the values, consider again two cases. If $\mathcal{R}(\mathcal{G})$ is non-negative, consider any memoryless attractor strategy σ_{Min} for Min toward S_f. Since all states have values below $+\infty$, all plays obtained from strategies of Max will follow simple paths of $\mathcal{R}(\mathcal{G})$, that have cumulated weight at most $|\mathcal{R}(\mathcal{G})|w^e_{\max}$ in absolute value. Similarly, if $\mathcal{R}(\mathcal{G})$ is non-positive, following the proof of Lemma 8, since all values are above $-\infty$, Max can ensure $\neg\phi \Rightarrow \mathrm{F}T_f^{\mathbb{R}} \vee \mathrm{FG}T_{\mathsf{K}}$ on all states. Then we can construct a strategy σ_{Max} for Max combining an attractor strategy toward S_f on states satisfying $\mathrm{F}T_f^{\mathbb{R}}$, a safety strategy on states satisfying GT_{K}, and an attractor strategy toward the latter on all other states. Then, all plays obtained from strategies of Min will either not be winning (GT_{K}) or follow simple paths of $\mathcal{R}(\mathcal{G})$. Both cases imply that the values of the game are bounded by $|\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T|$. ◀

▶ **Lemma 19.** *All plays in $\mathcal{T}(\mathcal{G})$ from the initial state to a stopped leaf have cumulated weight at least $2|\mathcal{R}(\mathcal{G})|w^e_{\max} + 2\sup|\mathsf{wt}_T| + 1$ if the SCC $\mathcal{R}(\mathcal{G})$ is non-negative, and at most $-2|\mathcal{R}(\mathcal{G})|w^e_{\max} - 2\sup|\mathsf{wt}_T| - 1$ if it is non-positive.*

Proof. Note that by construction, all finite paths in $\mathcal{T}(\mathcal{G})$ from the initial state to a stopped leaf can be decomposed as $\pi'\pi_1 \cdots \pi_{3|\mathcal{R}(\mathcal{G})|w^e_{\max}+2\sup|\mathsf{wt}_T|+1}$ with all π_i being cycles. Additionally, those cycles cannot be 0-cycles by Proposition 7, since they take at least one transition outside of K. Therefore the restriction of ρ to $\pi_1 \cdots \pi_{3|\mathcal{R}(\mathcal{G})|w^e_{\max}+1}$ has weight at least $3|\mathcal{R}(\mathcal{G})|w^e_{\max} + 2\sup|\mathsf{wt}_T| + 1$ (in the non-negative case) and at most $-3|\mathcal{R}(\mathcal{G})|w^e_{\max} -$

$2\sup|\mathsf{wt}_T| - 1$ (in the non-positive case). The beginning of the play, following π', has cumulated weight at least $-|\mathcal{R}(\mathcal{G})|w^e_{\max}$ (in the non-negative case) and at most $|\mathcal{R}(\mathcal{G})|w^e_{\max}$ (in the non-positive case), by Lemma 18. ◀

Two plays $\rho = ((\ell_1, r_1), \nu_1) \xrightarrow{d_1, t_1} \cdots \xrightarrow{d_{n-1}, t_{n-1}} ((\ell_n, r_n), \nu_n)$ and $\tilde\rho = ((\tilde\ell_1, r_1), \nu_1) \xrightarrow{d_1, \tilde t_1} \cdots \xrightarrow{d_{n-1}, \tilde t_{n-1}} ((\tilde\ell_n, r_n), \nu_n)$ in $\mathcal{R}(\mathcal{G})$ and $\mathcal{T}(\mathcal{G})$, respectively, are said to *mimic* each other if every $(\tilde\ell_i, r_i)$ is inherited from (ℓ_i, r_i) and every transition $\tilde t_i$ is inherited from the transition δ_i. Combining Lemmas 19 and 18, we obtain

▶ **Lemma 20.** *If $\mathcal{R}(\mathcal{G})$ is a non-negative (resp. non-positive) SCC, every play from the initial state and with cumulated weight less than $|\mathcal{R}(\mathcal{G})|w^e_{\max} + 2\sup|\mathsf{wt}_T| + 1$ (resp. greater than $-|\mathcal{R}(\mathcal{G})|w^e_{\max} - 2\sup|\mathsf{wt}_T| - 1$) can be mimicked in $\mathcal{T}(\mathcal{G})$ without reaching a stopped leaf. Conversely, every play in $\mathcal{T}(\mathcal{G})$ reaching a target leaf can be mimicked in $\mathcal{R}(\mathcal{G})$.*

Proof. We prove only the non-negative case. Let ρ be a play of $\mathcal{R}(\mathcal{G})$ with cumulated weight less than $|\mathcal{R}(\mathcal{G})|w^e_{\max} + 2\sup|\mathsf{wt}_T| + 1$. Consider the branch of the unfolded game it follows. If ρ cannot be mimicked in $\mathcal{T}(\mathcal{G})$, then a prefix of ρ reaches the stopped leaf of that branch when mimicked in $\mathcal{T}(\mathcal{G})$. In this situation, ρ starts by a prefix of weight at least $2|\mathcal{R}(\mathcal{G})|w^e_{\max} + 2\sup|\mathsf{wt}_T| + 1$ by Lemma 19 and then ends with a suffix play of weight at least $-|\mathcal{R}(\mathcal{G})|w^e_{\max}$ by Lemma 18, and that contradicts the initial assumption. The non-positive case is proved exactly the same way, and the converse is true by construction. ◀

Then, the plays of $\mathcal{R}(\mathcal{G})$ starting in an initial configuration that cannot be mimicked in $\mathcal{T}(\mathcal{G})$ are not useful for value computation, which is formalised by Proposition 21:

▶ **Proposition 21.** *For all valuations $\nu_0 \in r_0$, $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0) = \mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde\ell_0, r_0), \nu_0)$.*

Proof. By Lemma 1, we already know that $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0) = \mathsf{Val}_{\mathcal{R}(\mathcal{G})}((\ell_0, r_0), \nu_0)$. Recall that we only left finite values in $\mathcal{R}(\mathcal{G})$ (in the final weight functions, in particular), and more precisely $|\mathsf{Val}_{\mathcal{R}(\mathcal{G})}((\ell_0, r_0), \nu_0)| \leqslant |\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T|$ by Lemma 18. We first show that the value is also finite in $\mathcal{T}(\mathcal{G})$. Indeed, if $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde\ell_0, r_0), \nu_0) = +\infty$, since we assumed all output weights of $\mathcal{R}(\mathcal{G})$ bounded, we are necessarily in the non-negative case, and Max is able to ensure stopped leaves reachability.

Claim 1. If $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde\ell_0, r_0), \nu_0) = +\infty$, then there are no winning strategies in $\mathcal{R}(\mathcal{G})$ for Min ensuring weight less than $|\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T| + 1$ from (ℓ_0, r_0).

Thus, we can obtain the contradiction $\mathsf{Val}_{\mathcal{R}(\mathcal{G})}((\ell_0, r_0), \nu_0) > |\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T|$.

Proof of Claim 1. By contradiction, consider a strategy σ_{Min} of Min ensuring weight $A \leqslant |\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T| + 1$ in $\mathcal{R}(\mathcal{G})$. Then, for all σ_{Max}, the cumulated weight of $\mathsf{play}_{\mathcal{R}(\mathcal{G})}(((\tilde\ell_0, r_0), \nu_0), \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})$ (reaching target configuration (ℓ, ν)) is at most $A - \mathsf{wt}_T(\ell, \nu) \leqslant |\mathcal{R}(\mathcal{G})|w^e_{\max} + 2\sup|\mathsf{wt}_T| + 1$, and by Lemma 20 this play does not reach a stopped leaf when mimicked in $\mathcal{T}(\mathcal{G})$, which is absurd. ◀

If $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde\ell_0, r_0), \nu_0) = -\infty$, we are necessarily in the non-positive case, and by construction this implies having Min ensuring stopped leaves reachability in $\mathcal{T}(\mathcal{G})$.

Claim 2. If $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde\ell_0, r_0), \nu_0) = -\infty$, then there are no winning strategies in $\mathcal{R}(\mathcal{G})$ for Max ensuring weight above $-|\mathcal{R}(\mathcal{G})|w^e_{\max} - \sup|\mathsf{wt}_T| - 1$ from (ℓ_0, r_0).

Thus, we can obtain the contradiction $\mathsf{Val}_{\mathcal{R}(\mathcal{G})}((\ell_0, r_0), \nu_0) < -|\mathcal{R}(\mathcal{G})|w^e_{\max} - \sup|\mathsf{wt}_T|$.

Proof of Claim 2. By contradiction, consider a strategy σ_{Max} of Max ensuring weight $A \geqslant -|\mathcal{R}(\mathcal{G})|w^e_{\max} - \sup|\mathsf{wt}_T| - 1$ in $\mathcal{R}(\mathcal{G})$. Then, for all σ_{Min}, the cumulated weight of $\mathsf{play}_{\mathcal{R}(\mathcal{G})}(((\tilde{\ell}_0, r_0), \nu_0), \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})$ (reaching target configuration (ℓ, ν)) is at least $A - \mathsf{wt}_T(\ell, \nu) \geqslant -|\mathcal{R}(\mathcal{G})|w^e_{\max} - 2\sup|\mathsf{wt}_T| - 1$, and by Lemma 20 this play does not reach a stopped leaf when mimicked in $\mathcal{T}(\mathcal{G})$, which is absurd. ◀

Then, strategies and plays of $\mathcal{T}(\mathcal{G})$ starting from $(\tilde{\ell}_0, r_0)$ can be mimicked in $\mathcal{R}(\mathcal{G})$, therefore $\mathsf{Val}_{\mathcal{R}(\mathcal{G})}((\ell_0, r_0), \nu_0) \leqslant \mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\tilde{s}_0, \nu_0)$: If $\mathcal{R}(\mathcal{G})$ is non-negative, for all $\varepsilon > 0$ we can fix an ε-optimal strategy σ_{Min} for Min in $\mathcal{T}(\mathcal{G})$. It is a winning strategy, so every play derived from σ_{Min} in $\mathcal{T}(\mathcal{G})$ reaches a target leaf, and can be mimicked in $\mathcal{R}(\mathcal{G})$ by Lemma 20. Therefore, σ_{Min} can be mimicked in $\mathcal{R}(\mathcal{G})$, where it is also winning, with the same weight. From this we deduce $\mathsf{Val}_{\mathcal{R}(\mathcal{G})}((\ell_0, r_0), \nu_0) \leqslant \mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\tilde{s}_0, \nu_0)$. If $\mathcal{R}(\mathcal{G})$ is non-positive, the same reasoning applies by considering an ε-optimal strategy for Max in $\mathcal{T}(\mathcal{G})$.

Let us now show that $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0) \leqslant \mathsf{Val}_{\mathcal{R}(\mathcal{G})}((\ell_0, r_0), \nu_0)$. If $\mathcal{R}(\mathcal{G})$ is non-negative, let us fix $0 < \varepsilon < 1$, an ε-optimal strategy σ_{Min} for Min in $\mathcal{R}(\mathcal{G})$, and a strategy σ_{Max} of Max in $\mathcal{R}(\mathcal{G})$. Let ρ be their outcome $\mathsf{play}_{\mathcal{R}(\mathcal{G})}(((\ell_0, r_0), \nu_0), \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}}))$, ρ_k be the finite prefix of ρ defining its cumulative weight and (ℓ_k, ν_k) be the configuration defining its output weight, such that $\mathsf{wt}_{\mathcal{R}(\mathcal{G})}(\rho) = \mathsf{wt}_\Sigma(\rho_k) + \mathsf{wt}_T(\ell_k, \nu_k)$. Then, $\mathsf{wt}_{\mathcal{R}(\mathcal{G})}(\rho) \leqslant \mathsf{Val}_{\mathcal{R}(\mathcal{G})}((\ell_0, r_0), \nu_0) + \varepsilon < |\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T| + 1$, therefore $\mathsf{wt}_\Sigma(\rho_k) < |\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T| + 1 - \mathsf{wt}_T(\ell_k, \nu_k) \leqslant |\mathcal{R}(\mathcal{G})|w^e_{\max} + 2\sup|\mathsf{wt}_T| + 1$ and by Lemma 20 all such plays ρ can be mimicked in $\mathcal{T}(\mathcal{G})$, and $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0) \leqslant \mathsf{Val}_{\mathcal{R}(\mathcal{G})}((\ell_0, r_0), \nu_0)$. Once again, if $\mathcal{R}(\mathcal{G})$ is non-positive, the same reasoning applies by considering an ε-optimal strategy for Max in $\mathcal{R}(\mathcal{G})$. ◀

This proof not only holds on an SCC, but also on full almost-divergent WTGs, by simply stacking the semi-unfoldings of each SCC on top of each others.

Note that the semi-unfolding procedure of an SCC depends on $\sup|\mathsf{wt}_T|$, where wt_T can be the value function of an SCCs under the current one. Assuming all configurations have finite value, we can extend the reasoning of Lemma 18 and bound all values in the full game by $|\mathcal{R}(\mathcal{G})|w^e_{\max} + \sup|\mathsf{wt}_T|$, which let us bound uniformly the unfolding depth of each SCC and gives us a bound on the depth of the complete semi-unfolding tree: $|\mathcal{R}(\mathcal{G})|(5|\mathcal{R}(\mathcal{G})|w^e_{\max} + 2\sup|\mathsf{wt}_T| + 2) + 1$

D Proofs of the approximation scheme (Section 5)

D.1 Proofs of the approximation of kernels

Proof of Lemma 10. Since ρ and ρ' follow the same locations ℓ of $\mathcal{G}$, one reaches a target location if and only if the other does. In the case where they do not reach a target location, both weights are infinite, and thus equal. We now look at the case where both plays reach a target location, moreover in the same step.

Consider the region path π of the run ρ: π can be decomposed into a simple path with maximal cycles in it. The number of such maximal cycles is bounded by $|L \times \mathsf{Reg}(X, M)|$ and the remaining simple path has length at most $|L \times \mathsf{Reg}(X, M)|$. Since all cycles of a kernel are 0-cycles, the parts of ρ that follow the maximal cycles have weight exactly 0.

Consider the same decomposition for the play ρ'. Cycles of π do not necessarily map to cycles over locations of $\mathcal{C}_N(\mathcal{G})$, since the $1/N$-regions could be distinct. However, Lemma 2 shows that, for all those cycles of π, there exists a sequence of finite plays of $\mathcal{G}$ whose weight tends to the weight of ρ'. Since all those finite plays follow a cycle of the region game $\mathcal{R}(\mathcal{G})$ (with $\mathcal{G}$ being a kernel), they all have weight 0. Hence, the parts of ρ' that follow the maximal cycles of π have also weight exactly 0.

Therefore, the difference $|\mathsf{wt}_{\mathcal{G}}(\rho) - \mathsf{wt}_{\mathcal{C}_N(\mathcal{G})}(\rho')|$ is concentrated on the remaining simple path of π: on each transition of this path, the maximal weight difference is $1/N \times w^L_{\max}$ since $1/N$ is the largest difference possible in time delays between plays that stay $1/N$-close (since they stay in the same $1/N$-regions). Moreover, the difference between the output weight functions is bounded by K/N, since the output weight function wt_T is K-Lipschitz-continuous and the output weight function of $\mathcal{C}_N(\mathcal{G})$ is obtained as limit of wt_T. Summing the two contributions, we obtain as upper bound the constant $\mathbf{B}/N$. ◀

Proof of Lemma 11. Let us prove that both $\mathsf{Val}_{\mathcal{G}}(\ell,\nu) \leqslant \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v) + \alpha$ and $\mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v) \leqslant \mathsf{Val}_{\mathcal{G}}(\ell,\nu) + \alpha$, with $\alpha = \mathbf{B}/N$. By definition and determinacy of turn based WTG, this is equivalent to proving these two inequalities:

$$\inf_{\sigma_{\mathsf{Min}}} \sup_{\sigma_{\mathsf{Max}}} \mathsf{wt}_{\mathcal{G}}(\mathsf{play}((\ell,\nu),\sigma_{\mathsf{Max}},\sigma_{\mathsf{Min}})) \leqslant \inf_{\sigma'_{\mathsf{Min}}} \sup_{\sigma'_{\mathsf{Max}}} \mathsf{wt}_{\mathcal{C}_N(\mathcal{G})}(\mathsf{play}(((\ell,r,v),v),\sigma'_{\mathsf{Max}},\sigma'_{\mathsf{Min}})) + \alpha$$

$$\sup_{\sigma'_{\mathsf{Max}}} \inf_{\sigma'_{\mathsf{Min}}} \mathsf{wt}_{\mathcal{C}_N(\mathcal{G})}(\mathsf{play}(((\ell,r,v),v),\sigma'_{\mathsf{Max}},\sigma'_{\mathsf{Min}})) \leqslant \sup_{\sigma_{\mathsf{Max}}} \inf_{\sigma_{\mathsf{Min}}} \mathsf{wt}_{\mathcal{G}}(\mathsf{play}((\ell,\nu),\sigma_{\mathsf{Max}},\sigma_{\mathsf{Min}})) + \alpha$$

Let (β) denote $|\mathsf{wt}_{\mathcal{G}}(\mathsf{play}((\ell,\nu),\sigma_{\mathsf{Max}},\sigma_{\mathsf{Min}})) - \mathsf{wt}_{\mathcal{C}_N(\mathcal{G})}(\mathsf{play}(((\ell,r,v),v),\sigma'_{\mathsf{Max}},\sigma'_{\mathsf{Min}}))| \leqslant \alpha$. To show the first inequality, it suffices to show that for all σ'_{Min}, there exists σ_{Min} such that for all σ_{Max}, there is σ'_{Max} verifying (β). For the second, it suffices to show that for all σ'_{Max}, there exists σ_{Max} such that for all σ_{Min}, there is σ'_{Min} verifying (β). We will detail the proof for the first, the second being syntactically the same, with both players swapped.

Equation (β) can be obtained from Lemma 10, under the condition that the plays $\mathsf{play}((\ell,\nu),\sigma_{\mathsf{Max}},\sigma_{\mathsf{Min}})$ and $\mathsf{play}(((\ell,r,v),v),\sigma'_{\mathsf{Max}},\sigma'_{\mathsf{Min}})$ are $1/N$-close. Therefore, we fix a strategy σ'_{Min} of Min in the game $\mathcal{C}_N(\mathcal{G})$, and we construct a strategy σ_{Min} of Min in $\mathcal{G}$, as well as two mappings $f\colon \mathsf{FPlays}^{\mathsf{Min}}_{\mathcal{G}} \to \mathsf{FPlays}^{\mathsf{Min}}_{\mathcal{C}_N(\mathcal{G})}$ and $g\colon \mathsf{FPlays}^{\mathsf{Max}}_{\mathcal{C}_N(\mathcal{G})} \to \mathsf{FPlays}^{\mathsf{Max}}_{\mathcal{G}}$ such that:

- for all $\rho \in \mathsf{FPlays}^{\mathsf{Min}}_{\mathcal{G}}$, ρ and $f(\rho)$ are $1/N$-close, and if ρ is consistent with σ_{Min} and starts in (ℓ,ν), then $f(\rho)$ is consistent with σ'_{Min} and starts in $((\ell,r,v),v)$;
- for all $\rho' \in \mathsf{FPlays}^{\mathsf{Max}}_{\mathcal{C}_N(\mathcal{G})}$, $g(\rho')$ and ρ' are $1/N$-close, and if ρ' is consistent with σ'_{Min} and starts in $((\ell,r,v),v)$, then $g(\rho')$ is consistent with σ_{Min} and starts in (ℓ,ν).

We build σ_{Min}, f, and g by induction on the length n of plays, over prefixes of plays of length $n-1$, n and n, respectively. For $n=0$ (plays of length 0 are those restricted to a single configuration), we let $f(\ell,\nu) = ((\ell,r,v),v)$ and $g((\ell,r,v),v) = (\ell,\nu)$, leaving the other values arbitrary (since we will not use them).

Then, we suppose σ_{Min}, f, and g built until length $n-1$, n and n, respectively (if $n=0$, σ_{Min} has not been build yet), and we define them on plays of length n, $n+1$ and $n+1$, respectively. For every $\rho \in \mathsf{FPlays}^{\mathsf{Min}}_{\mathcal{G}}$ of length n, we note $\rho' = f(\rho)$. Consider the decision $(d',\delta') = \sigma'_{\mathsf{Min}}(\rho')$ and ρ'_+ the prefix ρ' extended with the decision (d',δ'). By timed bisimulation, there exists (d,δ) such that the prefix ρ_+ composed of ρ extended with the decision (d,δ) builds $1/N$-close plays ρ_+ and ρ'_+. We let $\sigma_{\mathsf{Min}}(\rho) = (d,\delta)$. If $\rho_+ \in \mathsf{FPlays}^{\mathsf{Min}}_{\mathcal{G}}$, we also let $f(\rho_+) = \rho'_+$, and otherwise we let $g(\rho'_+) = \rho_+$. Symmetrically,consider $\rho' \in \mathsf{FPlays}^{\mathsf{Max}}_{\mathcal{C}_N(\mathcal{G})}$ of length n, and $\rho = g(\rho')$. For all possible decisions (d',δ'), by timed bisimulation, there exists a decision (d,δ) in the prefix ρ such that the respective extended plays ρ'_+ and ρ_+ are $1/N$-close. We then let $g(\rho'_+) = \rho_+$ if $\rho_+ \in \mathsf{FPlays}^{\mathsf{Max}}_{\mathcal{G}}$ and $f(\rho_+) = \rho'_+$ otherwise. We extend the definition of f and g arbitrarily for other prefixes of plays. The properties above are then trivially verified.

We then fix a strategy σ_{Max} of Max in the game $\mathcal{G}$, which determines a unique play $\mathsf{play}((\ell,\nu),\sigma_{\mathsf{Max}},\sigma_{\mathsf{Min}})$. We construct a strategy σ'_{Max} of Max in the game $\mathcal{C}_N(\mathcal{G})$ by building the unique play $\mathsf{play}(((\ell,r,v),v),\sigma'_{\mathsf{Max}},\sigma'_{\mathsf{Min}})$ we will be interested in, such that each of its prefixes is in relation, via f or g, to the associated prefix of $\mathsf{play}((\ell,\nu),\sigma_{\mathsf{Max}},\sigma_{\mathsf{Min}})$. Thus, we only need to consider a prefix of play $\rho' \in \mathsf{FPlays}^{\mathsf{Max}}_{\mathcal{C}_N(\mathcal{G})}$ that starts in $((\ell,r,v),v)$ and is consistent with σ'_{Min}, and σ'_{Max} built so far. Consider the play $\rho = g(\rho')$, starting in (ℓ,ν) and consistent with σ_{Min}, and σ_{Max} (by assumption). For the decision $(d,\delta) = \sigma_{\mathsf{Max}}(\rho)$ (letting ρ_+ be the extended prefix), the definition of f and g ensures that there exists a decision (d',δ') after ρ' that results in an extended play ρ'_+ that is $1/N$-close, via f or g, with ρ_+. We thus can choose $\sigma'_{\mathsf{Max}}(\rho') = (d',\delta')$.

We finally have built two plays $\mathsf{play}((\ell,\nu),\sigma_{\mathsf{Max}},\sigma_{\mathsf{Min}})$ and $\mathsf{play}((\ell',\nu'),\sigma'_{\mathsf{Max}},\sigma'_{\mathsf{Min}})$ that are $1/N$-close, as needed, which concludes this proof. ◀

Proof of Lemma 12. By construction, the approximated value is piecewise linear with one piece per $1/N$-region. To prove the Lipschitz constant, it is then sufficient to bound the difference between $\mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v)$ and $\mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v'),v')$, for v and v' two corners of a $1/N$-region r. We can pick any valuation ν in r and apply Lemma 11 twice, between ν and v, and between ν and v'. We obtain $|\mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v) - \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v'),v')| \leqslant 2\mathbf{B}/N = 2\|v-v'\|_\infty \mathbf{B}$. ◀

D.2 Computing the value of an acyclic WTG

Note that for a piecewise linear functions with finitely many pieces, being K-Lipschitz-continuous over regions is equivalent to being continuous over regions and having all partial derivatives bounded by K in absolute value.

▶ **Lemma 22.** *If for all $\ell \in L$, V_ℓ is piecewise linear with finitely many pieces that have all their partial derivatives bounded by K in absolute value, then for all $\ell \in L$, $\mathcal{F}(V)_\ell$ is continuous over regions and piecewise linear with finitely many pieces that have all their partial derivatives bounded by $\max(K, |\mathsf{wt}(\ell)| + (n-1)K)$ in absolute value.*

Proof. We will show that for every region r, $\mathcal{F}(V)$ restricted to r has those properties. Note that they are transmitted over finite min and max operations. The continuity over regions is easy to prove because it is stable by inf and sup. We now use the notations and definitions of [1] to bound the partial derivatives.There exists a partition cost function (P,F) that represents V, with P an n-dimensional nested tube partition and F a mapping from the leaf nodes of P to linear expressions over variables in X. Intuitively, P defines a finite arborescence of convex spaces, defined by linear inequalities, whose root is the whole region r and whose leaves partition r into *cells*. A crucial property of those cells ([1, Theorem 4]) is that, for a given valuation ν, the delays t that need to be considered in the sup or inf operation of $\mathcal{F}(V)_{(\ell,\nu)}$ correspond to the intersection points of the diagonal half line containing the time successors of ν and borders of cells (if ν^b is such a valuation, $t = \|\nu^b - \nu\|_\infty$ is the associated delay). In particular, there is a finite number of such borders, and the final $\mathcal{F}(V)_\ell$ function can be written as a finite nesting of finite min and max operations over linear terms, each corresponding to a choice of delay and a transition to take. Formally, there are several cases to consider to define those terms, depending on delay and transition choices. For each available transition δ, those terms can either be:

1. If a delay 0 is taken and all clocks in $Y \subseteq X$ are reset by δ, then $\mathsf{wt}_\Sigma((\ell,\nu) \xrightarrow{0} (\ell,\nu) \xrightarrow{\delta} (\ell',\nu[Y:=0])) = \mathsf{wt}_\Sigma(\delta) + V_{(\ell',\nu[Y:=0])}$
2. If a delay $t > 0$ (leading to valuation ν^b on border B) is taken and the clocks in Y are reset by δ, then $\mathsf{wt}_\Sigma((\ell,\nu) \xrightarrow{t} (\ell,\nu^b) \xrightarrow{\delta} (\ell',\nu^b[Y:=0])) = \mathsf{wt}_\Sigma(\ell) \times t + \mathsf{wt}_\Sigma(\delta) + V_{(\ell',\nu^b[Y:=0])}$

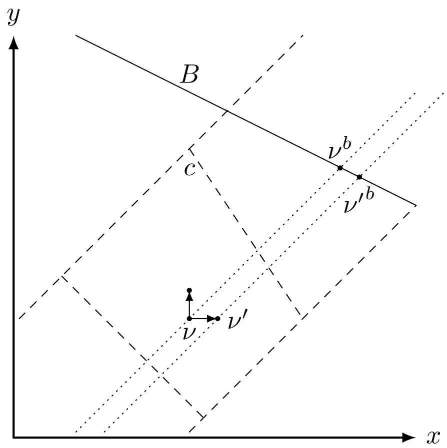


Figure 3 A tubular cell c as described in the proof of Lemma 22. Dashed lines bound the cell c, dotted lines are proof constructions.

In the first case, the resulting partial derivatives are 0 for clocks in Y, and the same as the partial derivatives in $V_{\ell'}$ for all other clocks, which allows us to conclude that they are bounded by K. We now consider the second case. We argue that the second case could be decomposed as a delay followed by a transition of the first case, meaning that we can assume $Y = \emptyset$ without loss of generality.

There are again two cases: the border B being inside a region or on the frontier of a region.

If the border is not the frontier of a region, it is the intersection points of two affine pieces of $V_{\ell'}$ whose equations (in the space $\mathbb{R}^{n+1}$ whose n first coordinates are the clocks $(x_1, \dots, x_n)$ and the last coordinate correspond to the value $V_{\ell'}(x_1, \dots, x_n)$) can be written $y = \sum_{i=1}^{n} a_i x_i + b$ (before the border) and $y = \sum_{i=1}^{n} a'_i x_i + b'$ (after the border). Therefore, valuations of the borders all fulfil the equation

$$\sum_{i=1}^{n} (a'_i - a_i) x_i + b - b' = 0 \tag{2}$$

We let $A = \sum_{i=1}^{n} (a'_i - a_i)$. Consider that ℓ is a location of Min (the very same reasoning applies to the case of a location of Max). Since $\mathcal{F}$ computes an infimum, we know that the function mapping the delay t to the weight obtained from reaching $\nu + t$ is decreasing before the border and increasing after. These functions are locally affine which implies that their slopes verify:

$$\mathsf{wt}(\ell) + \sum_{i=1}^{n} a_i \leqslant 0 \quad \text{and} \quad \mathsf{wt}(\ell) + \sum_{i=1}^{n} a'_i \geqslant 0\,. \tag{3}$$

We deduce from these two inequalities that $A \geqslant 0$. The case where $A = 0$ would correspond to the case where the border contains a diagonal line, which is forbidden, and $A > 0$. Consider now a valuation of coordinates $\nu = (x_1, \dots, x_n)$ and another valuation of coordinates $\nu' = (x_1, \dots, x_{k-1}, x_k + \lambda, x_{k+1}, \dots, x_n)$. The delays t and t' needed to arrive to the border starting from these two valuations are such that $\nu + t$ and $\nu' + t'$ both verify (2). We can then deduce that $t' - t = \lambda \frac{a_k - a'_k}{A}$. It is now possible to compute the partial derivative of

$\mathcal{F}(V)_\ell$ in the k-th coordinate using

$$\frac{\mathcal{F}(V)_{\ell,\nu'} - \mathcal{F}(V)_{\ell,\nu}}{\lambda} = \frac{\mathsf{wt}(\ell)(t'-t') + V_{\ell',\nu'+t'} - V_{\ell',\nu+t}}{\lambda}.$$

We may compute it by using the equations of the affine pieces before or after the border. We thus obtain

$$\frac{\mathcal{F}(V)_{\ell,\nu'} - \mathcal{F}(V)_{\ell,\nu}}{\lambda} = \frac{a_k - a'_k}{A}(\mathsf{wt}(\ell) + \sum_{i=1}^{n} a_i) + a_k$$

$$\frac{\mathcal{F}(V)_{\ell,\nu'} - \mathcal{F}(V)_{\ell,\nu}}{\lambda} = \frac{a_k - a'_k}{A}(\mathsf{wt}(\ell) + \sum_{i=1}^{n} a'_i) + a'_k$$

In the case where $a_k \geqslant a'_k$, the first equation, with (3), allows us to obtain that the partial derivative is at most a_k. We may then lower $\mathsf{wt}(\ell)$ by $-\sum_{i=1}^n a'_i$ to obtain that the partial derivative is at least a'_k. Since a_k and a'_k are bounded in absolute value by K, so is the partial derivative. We get the same result by reasoning on the second equation if $a'_k \geqslant a_k$.

We now come back to the case where the border is on the frontier of a region. Then, it is a segment of a line of equation $x_k = c$ for some k and c. $V_{\ell'}$ contains at most three values for points of B: The limit coming from before the border, the value at the border, and the limit coming from after the border. The computation of $\mathcal{F}(V)$ computes values obtained from all three and takes the min (or the max).

Now, let $y = \sum_{i=1}^n a_i x_i + b$ be the equation defining the linear piece of $V_{\ell'}$ before the border (resp. at the border, after the border). Consider now a valuation of coordinates $\nu = (x_1, \ldots, x_n)$ and another valuation of coordinates $\nu' = (x_1, \ldots, x_{j-1}, x_j + \lambda, x_{j+1}, \ldots, x_n)$. The delays t and t' needed to arrive to the border starting from these two valuations are such that $\nu + t$ and $\nu' + t'$ both verify $x_k = c$. We can then deduce that $t' - t = 0$ if $j \neq k$ and $t' - t = -\lambda$ if $j = k$. It is now possible to compute the partial derivative of $\mathcal{F}(V)_\ell$ in the j-th coordinate using

$$\frac{\mathcal{F}(V)_{\ell,\nu'} - \mathcal{F}(V)_{\ell,\nu}}{\lambda} = \frac{\mathsf{wt}(\ell)(t'-t') + V_{\ell',\nu'+t'} - V_{\ell',\nu+t}}{\lambda}.$$

We may compute it by using the equations of the linear piece before the border (resp. at the border, after the border). Then, $V_{\ell',\nu+t} = \sum_{i=1}^n a_i(x_i+t)+b = (\sum_{i=1 \neq k}^n a_i(x_i+t))+a_k c+b+$ and $V_{\ell',\nu'+t'} = (\sum_{i=1\neq k}^n a_i(x_i+t')) + a_k c + b$. We thus obtain

$$\frac{\mathcal{F}(V)_{\ell,\nu'} - \mathcal{F}(V)_{\ell,\nu}}{\lambda} = a_j \text{ if } j \neq k$$

$$\frac{\mathcal{F}(V)_{\ell,\nu'} - \mathcal{F}(V)_{\ell,\nu}}{\lambda} = -\mathsf{wt}(\ell) - \sum_{i=1, i\neq k}^{n} a_i \text{ otherwise}$$

Then, the partial derivatives are bounded, in absolute value, by $|\mathsf{wt}(\ell)| + (n-1)K$. ◀

As a corollary, we can now obtain Lemma 13, or more precisely:

▶ **Lemma 23.** *Consider an acyclic WTG $\mathcal{G}$ of depth i with all the output weights being K-Lipschitz-continuous over each region (and piecewise linear, with finitely many pieces). Then,*

- *if $|X| = 1$, $\mathsf{Val}_\mathcal{G} = \mathsf{Val}^i_\mathcal{G}$ is $\max(K, w^L_{\max})$-Lipschitz-continuous over regions;*
- *if $|X| = 2$, $\mathsf{Val}_\mathcal{G} = \mathsf{Val}^i_\mathcal{G}$ is $(i * w^L_{\max} + K)$-Lipschitz-continuous over regions;*
- *otherwise, $\mathsf{Val}_\mathcal{G} = \mathsf{Val}^i_\mathcal{G}$ is $(w^L_{\max}\frac{(|X|-1)^i-1}{|X|-2} + (|X|-1)^i K)$-Lipschitz-continuous over regions.*

D.3 Example of an execution of the approximation scheme

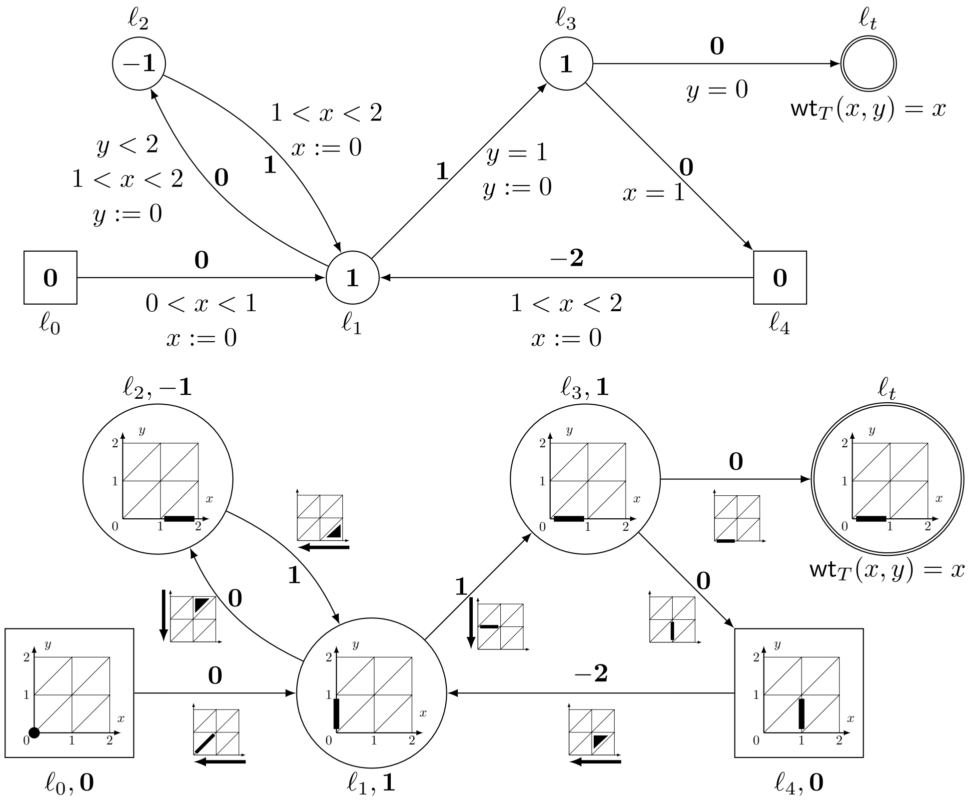


Figure 4 A weighted timed game $\mathcal{G}$ with two clocks x and y, and the portion of its region game $\mathcal{R}(\mathcal{G})$ accessible from configuration $(\ell_0, (0, 0))$. Locations of Min (resp. Max) are depicted as circles (resp. squares). The states of $\mathcal{R}(\mathcal{G})$ are labeled by their associated region, location and weight, and transitions are labeled by a representation of their guards and resets. Since each location ℓ of $\mathcal{G}$ leads to a unique states (ℓ, r) of $\mathcal{R}(\mathcal{G})$, we will refer to states by their associated location label.

We are given the WTG $\mathcal{G}$ in Figure 4 and $\varepsilon \in \mathbb{Q}_{>0}$, and want to compute an ε-approximation of its value in location ℓ_0 for the valuation $(x{=}0, y{=}0)$, denoted $\mathsf{Val}_{\mathcal{G}}(\ell_0, (0, 0))$. In this example, we will use $\varepsilon{=}15$ because the computations would not be readable with a smaller precision. $\mathcal{R}(\mathcal{G})$ contains one SCC $\{\ell_1, \ell_2, \ell_3, \ell_4\}$, made of two simple cycles. $\pi_1 = \ell_1 \to \ell_2 \to \ell_1$ is a positive cycle (all plays following π_1 have cumulated weight in the interval $(1, 3)$) and $\pi_2 = \ell_1 \to \ell_3 \to \ell_4 \to \ell_1$ is a 0-cycle (all plays following π_2 have cumulated weight 0). This can be checked by Lemma 2.

Therefore, $\mathcal{R}(\mathcal{G})$ only contains non-negative SCCs and is almost-divergent. Since all states are in the attractor of Min towards L_T, all cycles are non-negative and the output weight function is bounded (on all reachable regions), there are no configurations in $\mathcal{R}(\mathcal{G})$ with value $+\infty$ or $-\infty$.

We let the kernel K be the sub-game of $\mathcal{R}(\mathcal{G})$ defined by π_2, and we construct a semi-unfolding $\mathcal{T}(\mathcal{G})$ of $\mathcal{R}(\mathcal{G})$ of equivalent value. Following Appendix C, we should unfold the game until every stopped branch contains a state seen at least $3|\mathcal{R}(\mathcal{G})|w^e_{\max}{+}2\sup|\mathsf{wt}_T|{+}2 = 3 * 3 * 4 + 2 * 1 = 38$ times. We will unfold with bound 4 instead of 38 for readability (it is enough on this example). Thus the infinite branch $(\ell_1\ell_2)^\omega$ is stopped when ℓ_1 is reached for the fourth time, as depicted in Figure 5.

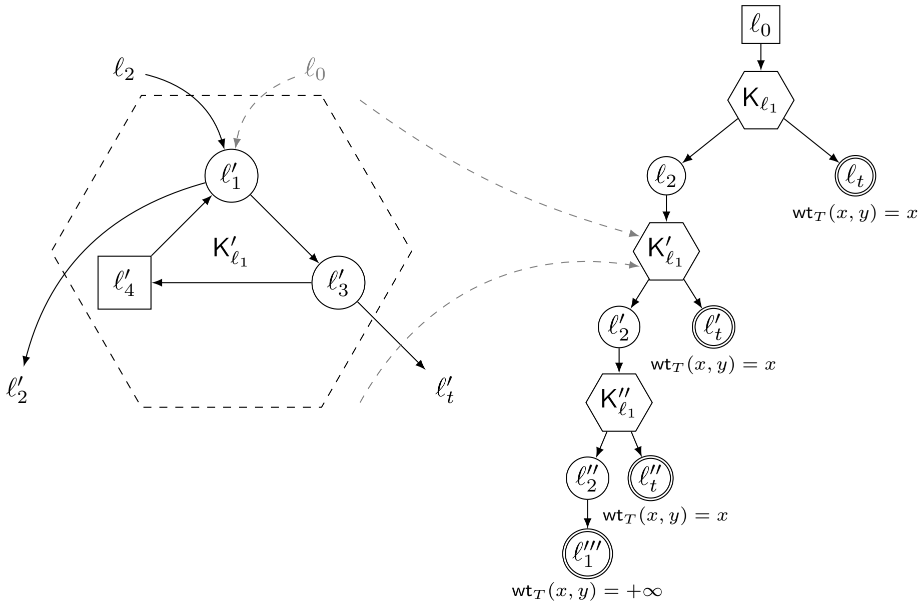


Figure 5 The kernel K (with input state ℓ_1), and a semi-unfolding $\mathcal{T}(\mathcal{G})$ such that $\mathsf{Val}_{\mathcal{G}}(\ell_0,(0,0)) = \mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell_0,(0,0))$. We denote ℓ_i, ℓ'_i and ℓ''_i the locations in K, K' and K''.

Let us now compute an approximation of $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}$. Let us first remove the states of value $+\infty$: ℓ'''_1 and ℓ''_2. Then, we start at the bottom and compute an $(\varepsilon/3)$-approximation of the value of ℓ''_1 in the game defined by K''_{ℓ_1} and its output transition to ℓ''_t. Following Section 5, we should use $N \geqslant 3(4+1)/\varepsilon$ and compute values in the $1/N$-corners game $\mathcal{C}_N(\mathsf{K}''_{\ell_1})$ in order to obtain an $(\varepsilon/3)$-approximation of the value function. For $\varepsilon = 15$ we will use $N = 1$ (in this case the computation happens to be exact and would also hold with a small ε) We construct this corner game, and obtain the finite (untimed) weighted game in Figure 6.

We can compute the values in this game to obtain $\mathsf{Val}(c'_1) = 1$ and $\mathsf{Val}(c_1) = 3$. We then define a value for every configuration in state ℓ''_1 by linear interpolation, obtaining $(x,y) \to 3-2y$ (which happens to be exactly $(x,y) \to \mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell''_1,(x,y))$ in this case, but would only be an $\varepsilon/3$-approximation of it in general). Now, we can compute an $\varepsilon/3$-approximation of $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell'_2)$ with one step of value iteration, obtaining $(x,y) \to \inf_{0<d<2-x}(-1)*d+1+3-2(0+d) = 3x-2$.

The next step is computing an $\varepsilon/3$-approximation of the value of ℓ'_1 in the game defined by K'_{ℓ_1} and its output transitions to ℓ'_t and ℓ'_2, of respective output weight functions $(x,y) \to x$ and $(x,y) \to 3x-2$. This will give us an $2\varepsilon/3$-approximation of $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell'_1)$.

Following Section 5 once again, we should use $N \geqslant 3(5+3)/\varepsilon$ and compute values in the $1/N$-corners game $\mathcal{C}_N(\mathsf{K}'_{\ell_1})$. For $\varepsilon = 15$ this gives $N = 2$ (which will once again keep the computation exact). We can construct a finite (untimed) weighted game as in Figure 6, and obtain a value for each $1/2$-corner of state ℓ'_1: On the $1/2$-region $(0<y<1/2, x=0)$, corner $(0,0)$ has value 2 and corner $(0,1/2)$ has value 2. On the $1/2$-region $(y=1/2, x=0)$, corner $(0,1/2)$ has value 2. On the $1/2$-region $(1/2<y<1, x=0)$, corner $(0,1/2)$ has value 2 and corner $(0,1)$ has value 1. From these results, we define a piecewise-linear function by interpolating the values of corners on each $1/2$-region, and

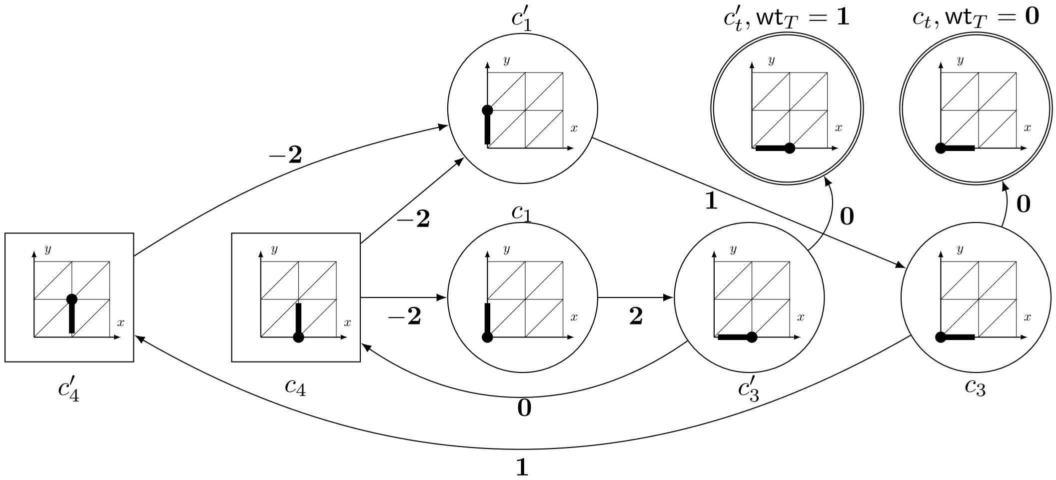


Figure 6 The finite weighted game obtained from $\mathcal{C}_1(\mathsf{K}''_{\ell_1})$, where c_i and c'_i are the corners of ℓ''_i in $\mathcal{T}(\mathcal{G})$.

obtain $(x, y) \to \begin{cases} 2 & \text{if } y \leqslant 1/2 \\ 3 - 2y & \text{otherwise} \end{cases}$, as depicted in Figure 7.

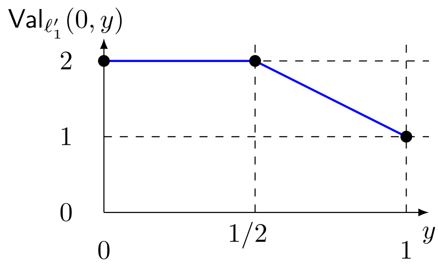


Figure 7 The value function $(x, y) \to \mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell'_1, (x, y))$, projected on $x = 0$. Black dots represent the values obtained for 1/2-corners using the corner-points abstraction.

This gives us an $2\varepsilon/3$-approximation of $(x, y) \to \mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell'_1, (x, y))$ (in fact exactly $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell'_1)$). Now, we can compute an $2\varepsilon/3$-approximation of $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell_2)$ on region $(1 < x < 2, y = 0)$ with one step of value iteration, obtaining :

$$(x, y) \to \inf_{0<d<2-x} \begin{cases} 3 - d & \text{if } d \leqslant 1/2 \\ 4 - 3d & \text{otherwise} \end{cases} = \begin{cases} 3x - 2 & \text{if } x \leqslant 3/2 \\ x + 1 & \text{otherwise} \end{cases}$$

Then, we need to compute an $\varepsilon/3$-approximation of the value of ℓ_1 in the game defined by K_{ℓ_1} and its output transitions to ℓ_t and ℓ_2, of respective output weight functions $(x, y) \to x$ and $(x, y) \to 3x - 2$ if $x \leqslant 3/2, x + 1$ otherwise. This will give us an ε-approximation of $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell_1)$.

Following Section 5 one last time, we should use $N \geqslant 3(5 + 3)/\varepsilon$ and compute values in the $1/N$-corners game $\mathcal{C}_N(\mathsf{K}_{\ell_1})$. This time, let us use $N = 3$ to showcase an example where the computed value is not exact. We can construct a finite (untimed) weighted game as in Figure 6, and obtain a value for each 1/3-corner of state ℓ'_1. From these results, we define a piecewise-linear function by interpolation, as depicted in Figure 8.

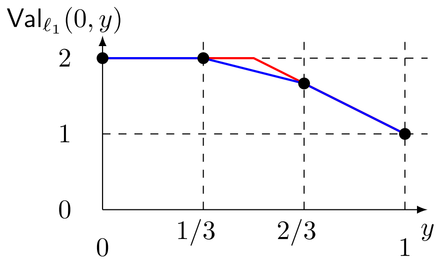


Figure 8 The value function $(x, y) \to \mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell_1, (x, y))$, projected on $x = 0$, is depicted in red. Black dots represent the values obtained for 1/3-corners using the corner-points abstraction, and the derived approximation of the value function is depicted in blue

Finally, from this ε-approximation of $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell_1)$, we can compute an ε-approximation of $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell_0)$ using one step of value iteration, and conclude. On our example this ensures $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell_0, (0, 0)) = \sup_{0<d<1} \mathsf{Val}_{\mathcal{T}(\mathcal{G})}(\ell_1, (0, d)) \in [2 - \varepsilon, 2 + \varepsilon]$.

D.4 Complexity analysis

We will express complexities according to several parameters: $|L|$, $|X|$, greatest guard constant M, greatest location and transition weight constants $w^L_{\max}$ and $w^\Delta_{\max}$. We also need to keep track of the output weight functions' characteristics. Recall that the output weight functions must be piecewise linear with finitely many pieces and Lipschitz-continuous over regions. We define three parameters, its Lipschitz constant K, its number of linear pieces J and a bound U (that we call additive bound) on its additive constant, such that if $(x_1, \dots, x_{|X|}) \to \sum_{i=1}^{|X|} a_i x_i + b$ defines one of those linear pieces, then $|b| \leqslant U$ and $\forall 1 \leqslant i \leqslant |X|, |a_i| \leqslant K$.

Note that $|L|$, $|X|$ and J are all polynomial in the size of the input, but M, $w^L_{\max}$, $w^\Delta_{\max}$, K and U are exponential in the size of the input if constants are encoded in binary.

We start with simple estimates:

- Number of regions $|\mathsf{Reg}(X, M)|$: Polynomial in M, exponential in $|X|$.
- Number of $1/N$-regions $|\mathsf{Reg}_N(X, M)|$: Polynomial in M and N, exponential in $|X|$.
- Number of $1/N$-corners: Polynomial in M and N, exponential in $|X|$.
- Maximum weight of a timed transition $w^e_{\max}$: Polynomial in M, $w^L_{\max}$ and $w^\Delta_{\max}$.
- Maximum output weight $\sup |\mathsf{wt}_T|$: Polynomial in M, U, $|X|$ and K.

D.4.1 Tree

Let us recall the complexity of the value iteration algorithm, used to compute the exact value of an acyclic WTG:
Input: An acyclic game of depth i.
Algorithm scheme: Computes $\mathcal{F}^i(V^0) = \mathsf{Val}^i = \mathsf{Val}$.
Output: A K'-Lipschitz-continuous function with J' pieces and additive bound U' that is the game's value.

- K' is of the form $K\,K''$ with K'' polynomial in $w^L_{\max}$ and $|X|$ and exponential in i.
- J' is of the form $J^{|X|}\,J''$ with J'' polynomial in M and $|L|$ and exponential in $|X|$ and i.
- U' is of the form $U + U''$ with U'' polynomial in M, $w^L_{\max}$, $w^\Delta_{\max}$ and i.

Complexity: exponential in i and the size of the input.

D.4.2 Kernel

Input: A kernel WTG, a precision $\varepsilon > 0$.
Algorithm scheme: Solves optimal reachability on the finite $1/N$-corner game with N polynomial in $1/\varepsilon$, $w^L_{\max}$, $|L|$, M and K and exponential in $|X|$.
Output: A K'-Lipschitz-continuous value function with J' pieces and additive bound U' that is an ε-approximation of the game's value.

- K' is of the form $K\,K''$ with K'' polynomial in $|L|$, $w^L_{\max}$ and M and exponential in $|X|$.
- J' is polynomial in $1/\varepsilon$, $w^L_{\max}$, $|L|$, M and K and exponential in $|X|$ (in particular, it is independent in J).
- U' of the form $U + U''$ with U'' polynomial in $1/\varepsilon$, $w^L_{\max}$, $w^\Delta_{\max}$, $|L|$, M and K and exponential in $|X|$.

Complexity: polynomial in $1/\varepsilon$, $w^L_{\max}$, $w^\Delta_{\max}$, $|L|$, M and K and exponential in $|X|$.

D.4.3 Semi-unfolding

We now stack several kernel and tree parts to form a semi-unfolding of a region game.
Input: A semi-unfolding of branch depth D, a precision $\varepsilon > 0$.
Algorithm scheme: value iteration for the trees and region-based for the kernels (on $1/N$ corners), with precision ε/D. In order to bound N, we need to bound the Lipschitz constants along the whole computation. We can recursively show that along this computation the Lipschitz constants, additive constants and number of pieces do not grow too much, and obtain global bounds:

- we can bound all Lipschitz constants by $K\,K''$ with K'' polynomial in $|L|$, $w^L_{\max}$, M and exponential in $|X|$ and D.
- we can bound all number of pieces by $J^{|X|}\,J''$ with J'' polynomial in $1/\varepsilon$, M, $|L|$, $w^L_{\max}$, and K and exponential in $|X|$ and D.
- we can bound all additive constants by $U + U''$ with U'' polynomial in $1/\varepsilon$, $w^L_{\max}$, $w^\Delta_{\max}$, $|L|$, M and K and exponential in $|X|$ and D.

Therefore, N can be chosen polynomial in $1/\varepsilon$, $w^L_{\max}$, $|L|$, M and K and exponential in $|X|$ and D.
Output: A K'-Lipschitz-continuous value function with J' pieces and additive bound U' that is an ε-approximation of the game's value. K', J', U' are bounded by their respective global bound.
Complexity: polynomial in $1/\varepsilon$ and exponential in the size of the input and D.

D.4.4 Almost divergent game

Input: An almost divergent game, a precision $\varepsilon > 0$.
Algorithm scheme: First, compute the region game's SCCs, and remove $+-\infty$ locations. Then, perform the semi-unfolding of the game, of depth D whose value is equivalent to that of the original game, with D polynomial in M, $|L|$, $w^L_{\max}$, $w^\Delta_{\max}$, K, U and exponential in $|X|$.
Output: A K'-Lipschitz-continuous value function with J' pieces and additive bound U' that is an ε-approximation of the game's value.

- K' is exponential in M, $|L|$, $w^L_{\max}$, $w^\Delta_{\max}$, K, U and doubly-exponential in $|X|$.
- J' is polynomial in J, $1/\varepsilon$, exponential in M, $|L|$, $w^L_{\max}$, $w^\Delta_{\max}$, K, U and doubly-exponential in $|X|$.

- U' is polynomial in $1/\varepsilon$ and exponential in M, $|L|$, $w^L_{\max}$, $w^\Delta_{\max}$, K, U and doubly-exponential in $|X|$

Complexity: polynomial in $1/\varepsilon$, exponential in the size of the input and M, K, U, $w^L_{\max}$ and $w^\Delta_{\max}$ and doubly-exponential in $|X|$.

E Proofs of the symbolic approximation scheme (Section 6)

This section is devoted to the proof of Theorem 14.

Notice that configurations with value $+\infty$ are stable through value iteration, and do not affect its other computations. Since we assumed the absence of value $-\infty$, we will therefore consider in the following that all configurations have finite value in $\mathcal{G}$.

Consider a game $\mathcal{G}$ that is a kernel. Following Section 5, we can define an integer N such that solving the untimed weighted game $\mathcal{C}_N(\mathcal{G})$ computes an $\varepsilon/2$-approximation of the value of $1/N$ corners.

Using the results of [14] for untimed weighted games, we know that those values are obtained after a finite number of steps of (the untimed version of) the value iteration operator. More precisely, if one considers a number of iterations $P = |L||\mathsf{Reg}_N(X,M)|(|X|+1)(2(|L||\mathsf{Reg}_N(X,M)|(|X|+1)-1)w^e_{\max}+1)$, then $\mathsf{Val}^P_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v) = \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v)$.

From this observation, we deduce the following property of P:

▶ **Lemma 24.** *If $\mathcal{G}$ is a kernel with no infinite value, $|\mathsf{Val}_\mathcal{G}(\ell,\nu) - \mathsf{Val}^P_\mathcal{G}(\ell,\nu)| \leqslant \varepsilon$ for all configurations (ℓ,ν) of $\mathcal{G}$.*

Proof. We already know that $\mathsf{Val}^P_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v) = \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v)$ for all configurations $((\ell,r,v),v)$ of $\mathcal{C}_N(\mathcal{G})$. Moreover, Section 5 ensures $|\mathsf{Val}_\mathcal{G}(\ell,\nu) - \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v)| \leqslant \varepsilon/2$ whenever ν is in the $1/N$-region r. Therefore, we only need to prove that $|\mathsf{Val}^P_\mathcal{G}(\ell,\nu) - \mathsf{Val}^P_{\mathcal{C}_N(\mathcal{G})}((\ell,r,v),v)| \leqslant \varepsilon/2$ to conclude. This is a simple rewriting of Lemma 11 that holds with exactly the same proof, since Lemma 10 does not depend on the length of the plays ρ and ρ', and both runs reach the target state in the same step, i.e. both before or after the horizon of P steps. ◀

Once we know that value iteration converges on kernels, we can use the semi-unfolding of Section 4 to prove that it also converges on non-negative SCCs when all values are finite.

▶ **Lemma 25.** *If $\mathcal{G}$ is a non-negative SCC with no infinite value, we can compute P such that $|\mathsf{Val}_\mathcal{G}(\ell,\nu) - \mathsf{Val}^P_\mathcal{G}(\ell,\nu)| \leqslant \varepsilon$ for all configurations (ℓ,ν) of $\mathcal{G}$.*

Proof. Consider a non-negative SCC's $\mathcal{G}$, a precision ε, and an initial configuration (ℓ_0,ν_0). Let $\mathcal{T}(\mathcal{G})$ be its finite semi-unfolding (obtained from the labelled tree T, as in Appendix C), such that $\mathsf{Val}_\mathcal{G}(\ell_0,\nu_0) = \mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0,r_0),\nu_0)$. Let α be the maximum number of kernels along a branch of T. Let P' be an integer such that for all kernels K in $\mathcal{T}(\mathcal{G})$, $|\mathsf{Val}_\mathsf{K}(\ell,\nu) - \mathsf{Val}^{P'}_\mathsf{K}(\ell,\nu)| \leqslant \varepsilon/\alpha$ for all configurations (ℓ,ν) of $\mathcal{G}$. We can find such a P' by using Lemma 24.

Create $\mathcal{T}'(\mathcal{G})$ from T by applying the method used to create $\mathcal{T}(\mathcal{G})$ but replace every kernel by its complete P'-unfolding instead. This implies that $\mathcal{T}'(\mathcal{G})$ is a tree, of bounded depth P (at most the depth of T times P'). Then $|\mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0,r_0),\nu_0) - \mathsf{Val}_{\mathcal{T}'(\mathcal{G})}((\tilde{\ell}_0,r_0),\nu_0)| \leqslant \varepsilon$. This holds because the value function is 1-Lipschitz-continuous with regards to the output weight function, so imprecision builds up additively.

Consider now $\mathcal{T}''(\mathcal{G})$ the (complete) unfolding of $\mathcal{R}(\mathcal{G})$ with unfolding depth P, where kernels are also unfolded. By construction, $\mathsf{Val}_{\mathcal{T}''(\mathcal{G})}((\tilde{\ell}_0,r_0),\nu_0) = \mathsf{Val}^P_{\mathcal{T}''(\mathcal{G})}((\tilde{\ell}_0,r_0),\nu_0)$. Then, we can prove that $\mathsf{Val}^P_{\mathcal{T}''(\mathcal{G})}((\tilde{\ell}_0,r_0),\nu_0) = \mathsf{Val}^P_\mathcal{G}(\ell_0,\nu_0)$ (same strategies at bounded horizon

P), which implies $\mathsf{Val}_{\mathcal{R}(\mathcal{G}))}((\ell_0, r_0), \nu_0) \leqslant \mathsf{Val}_{\mathcal{T}''(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$ (monotonicity of Val^k). By another monotonicity argument (because $\mathcal{T}''$ contains $\mathcal{T}'$ as a prefix), we can also prove $\mathsf{Val}_{\mathcal{T}''(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0) \leqslant \mathsf{Val}_{\mathcal{T}'(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$.

Bringing everything together we obtain $|\mathsf{Val}^P_{\mathcal{G}}(\ell_0, \nu_0) - \mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0)| \leqslant \varepsilon$. ◀

Proving the same property on non-positive SCCs requires more work, because the semi-unfolding gives stopped leaves $-\infty$ as output weight (for symmetry reasons), which doesn't integrate well with value iteration (initialisation at $+\infty$ on non-target states).

▶ **Lemma 26.** *If $\mathcal{G}$ is a non-positive SCC with no infinite value, there exists P such that $|\mathsf{Val}_{\mathcal{G}}(\ell, \nu) - \mathsf{Val}^P_{\mathcal{G}}(\ell, \nu)| \leqslant \varepsilon$ for all configurations (ℓ, ν) of $\mathcal{G}$.*

Proof. Consider a non-positive SCC $\mathcal{G}$, a precision ε, and an initial configuration (ℓ_0, ν_0). Let $\mathcal{T}(\mathcal{G})$ be its finite semi-unfolding (obtained from the labelled tree T, as in Appendix C), such that $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0) = \mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$.

We now change T, by adding a subtree under each stopped leaf: the complete unfolding of $\mathcal{R}(\mathcal{G})$, starting from the stopped leaf, of depth $|\mathcal{R}(\mathcal{G})|$. Let us name T^+ this unfolding tree. We then construct $\mathcal{T}^+(\mathcal{G})$ as before, based on T^+. Since we are in a non-positive SCC, $\mathcal{T}^+(\mathcal{G})$ must have output weight $-\infty$ on its stopped leaves. It is easy to see that $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0) = \mathsf{Val}_{\mathcal{T}^+(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$ still holds (the proof was based on branches being long enough, and we increased the lengths). We now perform a small but crucial change: the output weight of stopped leaves in $\mathcal{T}^+(\mathcal{G})$ is set to $+\infty$ instead of $-\infty$. Trivially $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0) \leqslant \mathsf{Val}_{\mathcal{T}^+(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$ (we increased the output weight function). Let us prove that $\mathsf{Val}_{\mathcal{T}^+(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0) \leqslant \mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$.

For a fixed $\eta > 0$, consider σ_{Min} a η-optimal strategy for player Min in $\mathcal{T}(\mathcal{G})$. Let us define σ^+_{Min}, a strategy for Min in $\mathcal{T}^+(\mathcal{G})$, by making the same choice as σ_{Min} on the common prefix tree, and once a node that is a stopped leaf in $\mathcal{T}(\mathcal{G})$ is reached, we switch to a memoryless attractor strategy of Min towards target states. Consider any strategy σ^+_{Max} of Max in $\mathcal{T}^+(\mathcal{G})$, and let σ_{Max} be its projection in $\mathcal{T}(\mathcal{G})$. Let ρ^+ denote the (maximal) play $\mathsf{play}_{\mathcal{T}^+(\mathcal{G})}(((\ell_0, r_0), \nu_0), \sigma^+_{\mathsf{Min}}, \sigma^+_{\mathsf{Max}}))$, and ρ be $\mathsf{play}_{\mathcal{T}(\mathcal{G})}(((\ell_0, r_0), \nu_0), \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}}))$. By construction, ρ^+ does not reach a stopped leaf in $\mathcal{T}^+(\mathcal{G})$. If the play ρ^+ stays in the common prefix tree of T and T^+, then $\rho = \rho^+$, and $\mathsf{wt}_{\mathcal{T}^+(\mathcal{G})}(\rho^+) \leqslant \mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0) + \eta$. If it doesn't, then ρ^+ has a prefix that reaches a stopped leaf in $\mathcal{T}(\mathcal{G})$: this must be ρ. This implies that $\mathsf{wt}_{\mathcal{T}^+(\mathcal{G})}(\rho^+) < -|\mathcal{R}(\mathcal{G})| w^e_{\max} - \sup |\mathsf{wt}_T| \leqslant \mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$ (see Lemma 20). Since this holds for all $\eta > 0$, we proved $\mathsf{Val}_{\mathcal{T}^+(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0) \leqslant \mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$, which finally implies that the two values are equal.

Then, we can follow the proof of Lemma 25 (with T^+ and $\mathcal{T}^+(\mathcal{G})$) in order to conclude. ◀

Now, if we are given an almost-divergent game $\mathcal{G}$ and a precision ε, we can glue together the semi-unfoldings of each SCC (non-positive SCCs have to get the same treatment as in Lemma 26 and get slightly more unfolded than the non-negative ones), and follow once again the proof of Lemma 25 in order to conclude. Therefore, by adding the convergence time of value iteration obtained from each SCC, we can obtain an integer P such that for all $k \geqslant P$, $\mathsf{Val}^k_{\mathcal{G}}$ is an ε-approximation of $\mathsf{Val}_{\mathcal{G}}$.