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
 https://orcid.org/0000-0002-4717-9955

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 schema on this class.

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 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 [18].

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 several winning strategies 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 the optimal reachability problem on weighted timed automata 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 the results of [16] 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 schema 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 weights 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, we prove that 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 schema of [10] to our setting. At the very core of their schema 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 schema 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 4 and 6). 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 configurations may have value $-\infty$. While it is undecidable in general whether a configuration has value $-\infty$, we prove that it is decidable for almost-divergent weighted timed games (see Lemma 9).

- 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 5.

We also develop, in Section 7, a more symbolic approximation schema, 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 when using *value iteration* techniques instead of techniques based on the unfolding of the model into a finite tree which can contain many times the same location.

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 refine the granularity in the following). Guard $\overline{g}$ is the closed version of a satisfiable 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]: intuitively, a region is a set of valuations that are all time-abstract bisimilar. We will 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 $x \in X_i$ satisfy 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 shorthand for $\mathsf{Reg}_1(X, M)$. We recover the traditional notion of region for $N = 1$. E.g., the figure below depicts regions $\mathsf{Reg}(\{x, y\}, 2)$ as well as their refinement $\mathsf{Reg}_3(\{x, y\}, 2)$.
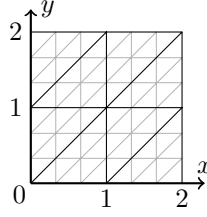
For any integer guard g, either all valuations of a given $1/N$-region satisfy g, or none of them do. A $1/N$-region r' is said to be a time successor of the $1/N$-region r if there exist $\nu \in r$, $\nu' \in r'$, and $d > 0$ such that $\nu' = \nu + d$. Moreover, for $Y \subseteq X$, we let $r[Y := 0]$ 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 $\mathsf{wt}_T \colon L_T \times \mathbb{R}_{\geqslant 0}^X \to \mathbb{R}_\infty$ is a function mapping each target location and valuation of the clocks 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 transitions start 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).
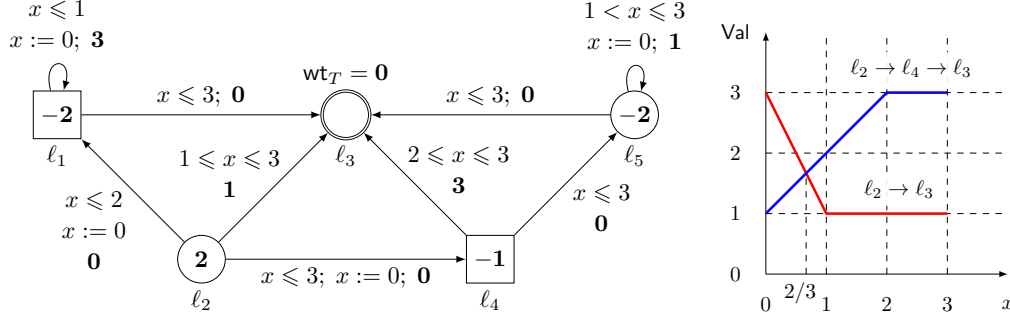
■ **Figure 1** On the left, a weighted timed game. Locations belonging to Min (resp. Max) are depicted by circles (resp. squares). The target location is ℓ_3, whose output weight function is null. It is easy to observe that location ℓ_1 (resp. ℓ_5) has value $+\infty$ (resp. $-\infty$). As a consequence, the value in ℓ_4 is determined by the edge to ℓ_3, and depicted in blue on the right. 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.

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 input 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. $w^\Delta_{\max} = \max_{\delta \in \Delta} |\mathsf{wt}(\delta)|$, $w^e_{\max} = M w^L_{\max} + w^\Delta_{\max}$). We also assume that the output weight functions are piecewise linear with a finite number of pieces and are continuous on each region. Notice that the zero output weight function satisfies this property. Moreover, the computations we will perform in the following maintain this property as an invariant, and use it to prove their correctness.

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'',

---

[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.

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 every edge $(\ell, \nu) \xrightarrow{d, \delta = (\ell, g, Y, \ell')} (\ell', \nu')$ by the transition $((\ell, r), g, Y, (\ell', r'))$, where r (resp. r') is the $1/N$-region containing ν (resp. ν'): 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 there exists $t = ((\ell, r), g, Y, (\ell', r'))$ 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 valuations $\nu \in r$, $\nu' \in r'$ such that $((\ell, r), \nu) \xrightarrow{d', t} ((\ell', r'), \nu')$ for some $d' \in \mathbb{R}_{\geqslant 0}$ (the latter condition ensures 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: a 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 above). 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. Figure 2 depicts a play, its projected path in the region game and one of its associated corner plays.
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 $\{\mathsf{wt}_\Sigma(\rho) \mid \rho \text{ finite play following } \pi\}$ is an interval bounded by the minimum and the maximum values of the set $\{\mathsf{wt}_\Sigma(\rho) \mid \rho \text{ finite corner play of } \mathcal{C}_N(\mathcal{G}) \text{ following } \pi\}$.*

---

[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.
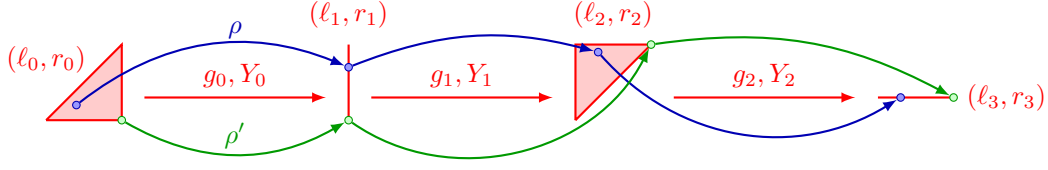
Figure 2 A play ρ (in blue), its projected path π in the region game (in red), and one of its associated corner plays ρ' (in green).

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 $L \times \mathbb{R}_{\geqslant 0}^X$ 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

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

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 targets 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 $\mathsf{Val}_\mathcal{G}^i$, which is intuitively what Min can guarantee when forced to reach the target in at most i steps.

More formally, we define $\mathsf{wt}_\mathcal{G}^i(\rho)$ 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 $\mathsf{Val}_\mathcal{G}^i(\ell, \nu) = \inf_{\sigma_{\mathsf{Min}}} \sup_{\sigma_{\mathsf{Max}}} \mathsf{wt}_\mathcal{G}^i(\mathsf{play}((\ell, \nu), \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}}))$. Then, if $\mathcal{G}$ is a tree of depth d, $V^i = \mathsf{Val}_\mathcal{G}$ if $i \geq d$.

The mappings V_ℓ^0 are piecewise linear for all ℓ, and $\mathcal{F}$ preserves piecewise linearity over regions, so all iterates V_ℓ^i are piecewise linear with a finite number of pieces. In [1], 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 $\mathsf{wt}_T \neq 0$.

3   Results

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 $\mathsf{Val}_\mathcal{G}(\ell, \mathbf{0})$.

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, 19, 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 authors 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.

In [16], we generalised the strictly non-Zeno cost property of [9, 16] to weighted timed games with both positive and negative weights: we called them divergent weighted timed games. This article relaxes the divergence property, to introduce almost-divergent weighted timed games. We first define formally these classes of games. 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 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 first 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 time doubly-exponential in the size of $\mathcal{G}$ and polynomial in $1/\varepsilon$. Moreover, deciding if a WTG is almost-divergent is PSPACE-complete.*

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] allow 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 3).

During the approximation process, we must think about the final weight functions as the previously computed approximations of the values of SCCs below the current one. 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 3) of $\mathcal{R}(\mathcal{G})$ from s_0 of finite depth, with certain nodes of the
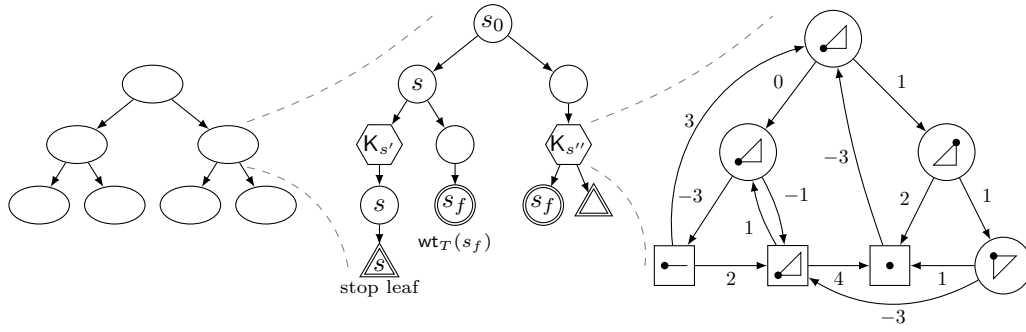
Figure 3 Static approximation schema: SCC decomposition of $\mathcal{R}(\mathcal{G})$, semi-unfolding of an SCC, corner-point abstraction for the kernels.

tree being *kernels*. These kernels are some parts of $\mathcal{R}(\mathcal{G})$ that contain all cycles of weight 0. The semi-unfolding is stopped either when reaching a final location, or when some location (or kernel) has been visited for a certain fixed number of times: such locations deep enough are called *stop leaves*.

Our second result is a more symbolic approximation schema based on the value iteration only. It is more symbolic in the sense that it does not require the SCC decomposition, the computation of kernels nor the semi-unfolding of the game in a tree.

▶ **Theorem 5.** *Let $\mathcal{G}$ be an almost-divergent WTG such that $\mathsf{Val}_\mathcal{G} > -\infty$ for all configurations. Then the sequence $(\mathsf{Val}_\mathcal{G}^k)_{k \geqslant 0}$ converges towards $\mathsf{Val}_\mathcal{G}$ and for every $\varepsilon \in \mathbb{Q}_{>0}$, we can compute an integer P such that $\mathsf{Val}_\mathcal{G}^P$ is an ε-approximation of $\mathsf{Val}_\mathcal{G}$ for all configurations.*

▶ **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 $+\infty$.

4 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 $t_1, \ldots, t_n$ 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 6.** *A cycle π is a 0-cycle iff there exists a corner play ρ following π with $\mathsf{wt}_\Sigma(\rho)=0$.*

Proof. 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.   ◀

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 7.** *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.*

Proof. 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.

First, since there are finitely many corners, 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. Formally, we 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 ρ'_k, ρ_{k+1}, $\rho'_{k+1}, \rho_{k+2}, \ldots, \rho'_{l-1}, \rho_l$. 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 6, we deduce that π_2 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 6.

▶ **Proposition 8.** *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 6 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.
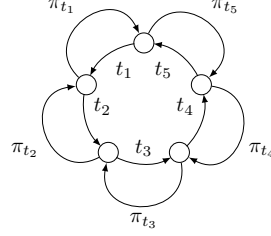
We now prove that every cycle in K is a 0-cycle. By construction, every transition $t \in T_\mathsf{K}$ is part of a simple 0-cycle. Thus, to every transition $t \in T_\mathsf{K}$, we can associate a path π_t such that $t\pi_t$ is a simple 0-cycle (rotate the simple cycle if necessary). We can prove (using both Lemmas 6 and 7) the following property by relying on another pumping argument on corners: 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. ◀

5 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 3). This crucially relies on the absence of states with value $-\infty$, so we explain how to deal with them first:

▶ **Lemma 9.** *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 $\mathcal{G}$ have 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 now explain how to build the semi-unfolding $\mathcal{T}(\mathcal{G})$. We only build the semi-unfolding $\mathcal{T}(\mathcal{G})$ of an SCC of $\mathcal{G}$ starting from some state $(\ell_0, r_0) \in S$ of the region game, 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 $|\mathcal{R}(\mathcal{G})|w_{\max}^e + \sup |\mathsf{wt}_T|$. As a consequence, we can find a bound γ linear in $|\mathcal{R}(\mathcal{G})|$, $w_{\max}^e$

and $\sup|\mathsf{wt}_T|$ such that a play that visits some state outside the kernel more than γ times has weight strictly above $|\mathcal{R}(\mathcal{G})|w_{\max}^e + \sup|\mathsf{wt}_T|$, hence is useless for the 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 3) such that each node not in the kernel is encountered at most γ times along a branch: the end of each branch is called a *stop leaf* of the semi-unfolding. In particular, the depth of $\mathcal{T}(\mathcal{G})$ is bounded by $|\mathcal{R}(\mathcal{G})|\gamma$, and thus is polynomial in $|\mathcal{R}(\mathcal{G})|$, $w_{\max}^e$ and $\sup|\mathsf{wt}_T|$. Leaves of the semi-unfolding are thus of two types: target leaves that are copies of target locations of $\mathcal{G}$ for which we set the target weight as in $\mathcal{G}$, and stop leaves for which we set their target weight as being constant to $+\infty$ if the SCC $\mathcal{G}$ is non-negative, and $-\infty$ if the SCC is non-positive.

▶ **Proposition 10.** *Let $\mathcal{G}$ be an almost-divergent WTG, and let $(\ell_0, r_0) \in S$ be some state of the region game. The semi-unfolding $\mathcal{T}(\mathcal{G})$ with initial state $(\tilde{\ell}_0, r_0)$ (a copy of state (ℓ_0, r_0)) is equivalent to $\mathcal{G}$, i.e. for all $\nu_0 \in r_0$, $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0) = \mathsf{Val}_{\mathcal{T}(\mathcal{G})}((\tilde{\ell}_0, r_0), \nu_0)$.*

6   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 to the right of Figure 3).

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

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:

▶ **Lemma 11.** *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 12.** *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

---

[3]   The function wt_T is said to be Λ-Lipschitz-continuous when $|\mathsf{wt}_T(s, \nu) - \mathsf{wt}_T(s, \nu')| \leqslant \Lambda \|\nu - \nu'\|_\infty$ 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 Λ-Lipschitz-continuous, for some Λ.

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 13.** *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 10.

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 13. For non-kernel nodes of $\mathcal{T}(\mathcal{G})$, using a careful analysis of the algorithm of [1], we obtain the following bound:

▶ **Lemma 14.** *If all the output weights of a WTG $\mathcal{G}$ are Λ-Lipschitz-continuous over regions (and piecewise linear, with finitely many pieces), then $\mathsf{Val}^i_\mathcal{G}$ is $\Lambda\Lambda'$-Lipschitz-continuous over regions, with Λ' 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 $1/\varepsilon$.

7 Symbolic approximation algorithm

The previous approximation result suffers from several drawbacks. It relies on the SCC decomposition of the region automaton. Each of these SCCs 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 7, is an approximation schema. This is stated in Theorem 5, for which we now sketch a proof in this section.

Notice that configurations with value $+\infty$ are stable through value iteration, and do not affect its other computations. Since Theorem 5 assumes the absence of configurations of value $-\infty$, we will therefore consider in the following that all configurations have finite value in $\mathcal{G}$.

---

[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.

Consider first a game $\mathcal{G}$ that is a kernel. By the results of Section 6, there exists 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 $P = |L||\mathsf{Reg}_N(X, M)|(|X| + 1)(2(|L||\mathsf{Reg}_N(X, M)|(|X| + 1) - 1)w_{\max}^e + 1)$, then $\mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}^P((\ell, r, v), v) = \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell, r, v), v)$. From this observation, we deduce the following property of P:

▶ **Lemma 15.** *If $\mathcal{G}$ is a kernel with no configurations of infinite value, then $|\mathsf{Val}_{\mathcal{G}}(\ell, \nu) - \mathsf{Val}_{\mathcal{G}}^P(\ell, \nu)| \leqslant \varepsilon$ for all configurations (ℓ, ν) of $\mathcal{G}$.*

Proof. We already know that $\mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}^P((\ell, r, v), v) = \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}((\ell, r, v), v)$ for all configurations $((\ell, r, v), v)$ of $\mathcal{C}_N(\mathcal{G})$. Moreover, Section 6 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 $|\mathsf{Val}_{\mathcal{G}}^P(\ell, \nu) - \mathsf{Val}_{\mathcal{C}_N(\mathcal{G})}^P((\ell, r, v), v)| \leqslant \varepsilon/2$ to conclude. This is done as for Lemma 12, since Lemma 11 (that we need to prove Lemma 12) 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 5 to prove that it also converges on non-negative SCCs when all values are finite.

▶ **Lemma 16.** *If $\mathcal{G}$ is a non-negative SCC with no configurations of infinite value, we can compute P_+ such that $|\mathsf{Val}_{\mathcal{G}}(\ell, \nu) - \mathsf{Val}_{\mathcal{G}}^{P_+}(\ell, \nu)| \leqslant \varepsilon$ for all configurations (ℓ, ν) of $\mathcal{G}$.*

The idea is to unfold every kernel of the semi-unfolding game $\mathcal{T}(\mathcal{G})$ according to its bound in Lemma 15. More precisely, let α be the maximum number of kernels along one of the branches of $\mathcal{T}(\mathcal{G})$. In a bottom-up fashion, we can find for each kernel K in $\mathcal{T}(\mathcal{G})$ a bound P_K such that, for all configurations (ℓ, ν), $|\mathsf{Val}_\mathsf{K}(\ell, \nu) - \mathsf{Val}_\mathsf{K}^{P_\mathsf{K}}(\ell, \nu)| \leqslant \varepsilon/\alpha$. We thus unfold K in $\mathcal{T}(\mathcal{G})$ with depth up to P_K. After each kernel has been replaced this way, $\mathcal{T}(\mathcal{G})$ is no longer a semi-unfolding, it is instead a (complete) unfolding of $\mathcal{R}(\mathcal{G})$, of a certain bounded depth P_+. This new bound P_+ is bounded by the former depth of $\mathcal{T}(\mathcal{G})$ to which is added α times the biggest bound P_K we need for the kernels. Now, $\mathcal{T}(\mathcal{G})$ is a tree of depth P_+ whose value at its root is ε-close to the value of $\mathcal{G}$. Finally, the value computed by $\mathsf{Val}_{\mathcal{G}}^{P_+}$ is bounded between $\mathsf{Val}_{\mathcal{G}}$ and $\mathsf{Val}_{\mathcal{T}(\mathcal{G})}$, which allows us to conclude.

The bound P_K for a kernel K depends linearly in Λ, the Lipschitz constant of value functions on locations of $\mathcal{T}(\mathcal{G})$ reachable from K. Once K has been replaced by its unfolding of depth P_K, the Lipschitz constant of the value function at the root of $\mathcal{T}(\mathcal{G})$ are thus bounded exponentially in Λ. This means that we ensure a bound for P_+ that is at most polynomial in $1/\varepsilon$, and that is of the order of a tower of α exponentials.

Proving the same property on non-positive SCCs requires more work, because the semi-unfolding gives output weight $-\infty$ to stop leaves, which doesn't integrate well with value iteration (initialisation at $+\infty$ on non-target states). However, by unfolding those SCCs slightly more (at most $|\mathcal{R}(\mathcal{G})|$ more steps), we can obtain the desired property with a similar bound P_-.

▶ **Lemma 17.** *If $\mathcal{G}$ is a non-positive SCC with no configurations of infinite value, we can compute P_- such that $|\mathsf{Val}_{\mathcal{G}}(\ell, \nu) - \mathsf{Val}_{\mathcal{G}}^{P_-}(\ell, \nu)| \leqslant \varepsilon$ for all configurations (ℓ, ν) of $\mathcal{G}$.*

Now, if we are given an almost-divergent game $\mathcal{G}$ and a precision ε, we can add the bounds for value iteration obtained from each SCC by Lemmas 16 and 17, and obtain a final bound P such that for all $k \geqslant P$, $\mathsf{Val}_{\mathcal{G}}^k$ is an ε-approximation of $\mathsf{Val}_{\mathcal{G}}$.

Discussion.   Overall, this leads to an upper bound complexity that is polynomial in $1/\varepsilon$ and of the order of a tower of n exponentials, with n polynomial in the size of the input WTG. However, we argue that this symbolic procedure is more amenable to implementation than the previous approximation schema. First, it avoids the three already mentioned drawbacks (SCC decomposition, sequential analysis of the SCCs, and refinement of the granularity of regions) of the previous approximation schema. Then, it allows one to directly launch the value iteration algorithm on the game $\mathcal{G}$, and we can stop the computation whenever we are satisfied enough by the approximation computed: however, there are no guarantees whatsoever on the quality of the approximation before the number of steps P given above. Finally, this schema allows one to easily obtain an almost-optimal strategy with respect to the computed value.

If $\mathcal{G}$ is not guaranteed to be free of configurations of value $-\infty$, then we must first 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 9. Then, we can apply the value iteration algorithm.

As a final remark, notice that our correctness proof strongly relies on Section 6, and thus would not hold with the approximation schema of [10] (which does not preserve the continuity on regions of the computed value functions, in turn needed to define output weights on $1/N$-corners).

8   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 correctness of a symbolic approximation schema, 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 of this work 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. `doi:10.4230/LIPIcs.FSTTCS.2015.278`.

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   Marcin Jurdziński and Ashutosh Trivedi. Reachability-Time Games on Timed Automata. In *Proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP'07)*, volume 4596 of *LNCS*, pages 838–849. Springer, 2007.

19   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.