Minimizing Regret in Discounted-Sum Games*

Paul Hunter[1], Guillermo A. Pérez[†2], and Jean-François Raskin[3]

1 **Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium**
phunter@ulb.ac.be
2 **Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium**
gperezme@ulb.ac.be
3 **Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium**
jraskin@ulb.ac.be

Abstract

In this paper, we study the problem of minimizing regret in discounted-sum games played on weighted game graphs. We give algorithms for the general problem of computing the minimal regret of the controller (Eve) as well as several variants depending on which strategies the environment (Adam) is permitted to use. We also consider the problem of synthesizing regret-free strategies for Eve in each of these scenarios.

1998 ACM Subject Classification F.1.1 Automata, D.2.4 Formal methods

Keywords and phrases Quantitative games, Regret, Verification, Synthesis, Game theory

Digital Object Identifier 10.4230/LIPIcs.CSL.2016.30

1 Introduction

Two-player games played by Eve and Adam on weighted graphs is a well accepted mathematical formalism for modelling quantitative aspects of a controller (Eve) interacting with its environment (Adam). The outcome of the interaction between the two players is an infinite path in the weighted graph and a value is associated to this infinite path using a *measure* such as *e.g.* the mean-payoff of the weights of edges traversed by the infinite path, or the discounted sum of those weights. In the classical model, the game is considered to be zero sum: the two players have antagonistic goals–one of the player want to maximize the value associated to the outcome while the other want to minimize this value. The main solution concept is then the notion of winning strategy and the main decision problem asks, given a threshold c, whether Eve has a strategy to ensure that, no matter how Adam plays, that the outcome has a value larger than or equal to c.

When the environment is not fully antagonistic, it is reasonable to study other solution concepts. One interesting concept to explore is the concept of *regret minimization* [3] which is as follows. When a strategy of Adam is fixed, we can identify the set of Eve's strategies that allow her to secure the *best possible outcome* against this strategy. This constitutes Eve's *best response*. Then we define the regret of a strategy σ of Eve as the difference between Eve's best response; and the payoff she secures thanks to her strategy σ. So, when trying to minimize the regret associated to a strategy, we use best responses as a *yardstick*. Let us now illustrate this with an example.

* This work was supported by the ERC inVEST (279499) project.
† Author supported by F.R.S.-FNRS fellowship.

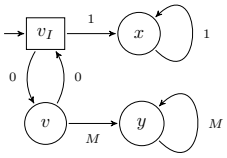

Figure 1 A game in which waiting is required to minimize regret.

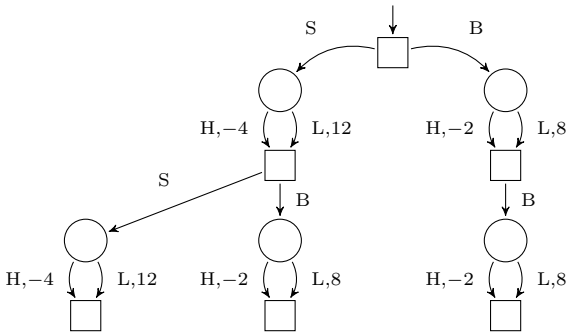

Figure 2 A game that models different investment strategies.

Table 1 The possible rate configurations for the rate of interests are given as the first four columns, then follows the worst-case performance and the regret associated to each strategy of Eve that are given in rows. Entries in bold are the values that are maximizing the worst case (strategy BB) and minimizing the regret (strategy SB).

	HH	HL	LH	LL	Worst-case	Regret
SS	−7.7616	7.6048	7.9784	23.2848	−7.7616	3.8808
SB	−5.8408	3.7632	9.8392	19.4432	−5.8408	**3.8416**
BB	−3.8808	5.7232	5.9192	15.5232	**−3.8808**	7.7616

▶ **Example 1** (Investment advice)**.** Consider the discounted sum game of Fig. 2. It models the rentability of different investment plans with a time horizon of two periods. In the first period, it can be decided to invest in treasure bonds (B) or to invest in the stock market (S). In the former case, treasure bonds (B) are chosen for two periods. In the latter case, after one period, there is again a choice for either treasure bonds (B) or stock market (S). The returns of the different investments depend on the fluctuation of the rate of interests. When the rate of interests is low (L) then the return for the stock market investments is equal to 12 and for the treasure bonds it is equal to 8. When the interest rate is high (H) then the returns for the stock market investments is equal to −4 and for the treasure bonds it is equal to −2. To model time and take into account the inflation rate, say equal to 2 percent, we consider a discount factor $\lambda = 0.98$ for the returns. In this example, we make the hypothesis that the fluctuation of the rate of interests is *not* a function of the behavior of the investor. It means that this fluctuation rate is either one of the following four possibilities: HH, HL, LH, LL. This corresponds to Adam playing a *word strategy* in our terminology. The discounted sum of returns obtained under the 12 different scenarios are given in Table 1.

Now, assume that you are a broker and you need to advise one of your customers regarding his next investment. There are several ways to advise your customer. First, if your customer is strongly *risk averse*, then you should be able to convince him that he has to go for the treasure bonds (B). Indeed, this is the choice that *maximizes the worst case*: if the interest rates stay high for two periods (HH) then the loss will be −3.8808 while it will be higher for any other choices. Second, and maybe more interestingly, if your customer tolerates some risks, then you may want to keep him happy so that he will continue to ask for your advice in the future! Then you should propose the following strategy: first invest in the stock market (S) then in treasure bonds (B) as this strategy *minimizes regret*. Indeed, at the end of the two investment periods, the actual interest rates will be known and so your customer will

evaluate your advices *ex-post*. So, after the two periods, the value of the choices made *ex ante* can be compared to the best strategy that could have been chosen knowing the evolution of the interest rates. The regret of SB is at most equal to 3.8416 in all cases and it is minimal: the regret of BB can be as high as 7.7616 if LL is observed, and the regret of SS can be as high as 3.8808.

Finally, let us remark that if the investments are done in financial markets that are subject to different interest rates, then instead of considering the minimization of regret against word strategies, then we could consider the regret against all strategies. We also study this case in this paper. ◀

Previous works. In [8], we studied regret minimization in the context of reactive synthesis for shortest path objectives. Recently in [13], we studied the notion of regret minimization when we assume different sets of strategies from which Adam chooses. We have considered three cases: when the Adam is allowed to play *any* strategy, when he is restricted to play a *memoryless* strategy, and when he plays *word* strategies. We refer the interested reader to [13] for motivations behind each of these definitions. In that paper, we studied the regret minimization problem for the following classical quantitative measures: inf, sup, $\liminf$, $\limsup$ and the *mean-payoff* measure. In this paper, we complete this picture by studying the regret minimization problem for the *discounted-sum* measure. Discounted-sum is a central measure in quantitative games but we did not consider it in [13] because it requires specific techniques which are more involved than the ones used for the other quantitative measures. For example, while for mean-payoff objectives, strategies that minimize regret are memoryless when Adam can play any strategy, we show in this paper that pseudo-polynomial memory is necessary (and sufficient) to minimize regret in discounted-sum games. The need for memory is illustrated by the following example.

▶ **Example 2.** Consider the example in Figure 1 where $M \gg 1$. Eve can play the following strategies in this game: let $i \in \mathbb{N} \cup \{\infty\}$, and note σ^i the strategy that first plays i rounds the edge (v_I, v) and then switches to (v_I, x). The regret values associated to those strategies are as follows.

The regret of σ^∞ is $\frac{1}{1-\lambda}$ and it is witnessed when Adam never plays the edge (v, y). Indeed, the discounted sum of the outcome in that case is 0, while if Eve had chosen to play (v_I, x) at the first step instead, then she would have gained $\frac{1}{1-\lambda}$. The regret of σ^i is equal to the maximum between $\frac{1}{1-\lambda} - \lambda^{2i}\frac{1}{1-\lambda}$ and $\lambda^{2i+1}\frac{M}{1-\lambda} - \lambda^{2i}\frac{1}{1-\lambda}$. The maximum is either witnessed when Adam never plays (v, y) or plays (v, y) if the edge (v_I, x) has been chosen $i+1$ times (one more time compared to σ^i).

So the strategy that minimizes regret is the strategy σ^N for $N > \frac{-\log M}{2\log\lambda} - \frac{1}{2}$ (so that $\lambda^{2N+1}M < 1$), *i.e.* the strategy needs to count up to N. ◀

Contributions. We describe algorithms to decide the regret threshold problem for games in three cases: when there is no restriction on the strategies that Adam can play, when Adam can only play memoryless strategies, and when Adam can only play word strategies. For this last case, our problem is closely related to open problems in the field of discounted-sum automata, and we also consider variants given as ε-gap promise problems.

We also study the complexity of the special case when the threshold is 0, *i.e.* when we ask for the existence of regret free strategies. We show that problem is sometimes easier to solve. Our results on the complexity of both the regret threshold and the regret-free problems are summarized in Table 2. All our results are for fixed discount factor λ.

Table 2 Complexity of deciding the regret threshold and regret-free problems for fixed λ.

	Any strategy	Memoryless strategies	Word strategies
regret threshold	NP (Thm. 8)	PSPACE (Thm. 20), coNP-h (Thm. 26)	PSPACE-c (ε-gap) (Thm. 29, Thm. 30)
regret-free	PTIME (Thm. 9)	PSPACE (Thm. 22), coNP-h (Thm. 26)	NP-c (Thm. 27)

Other related works. A Boolean version of our *regret-free strategies* has been described in [7]. In that paper, they are called *remorse-free strategies*. These correspond to strategies that ensure a regret value of 0 in games with ω-regular objectives. They do not establish lower bounds on the complexity of realizability or synthesis of remorse-free strategies and they only consider word strategies for Adam.

In [13], we established that regret minimization when Adam plays word strategies only is a generalization of the notion of *good-for-games automata* [10] and *determinization by pruning* (of a refinement) [1].

The notion of regret is closely related to the notion of competitive ratios used for the analysis of online algorithms [14]: the performance of an online algorithm facing uncertainty (*e.g.* about the future incoming requests or data) is compared to the performance of an offline algorithm (where uncertainty is resolved). According to this quality measure, an online algorithm is better if its performance is closer to the performance of an optimal offline solution.

Structure of the paper. In Sect. 2, we introduce the necessary definitions and notations. In Sect. 3, we study the minimization of regret when the second player plays any strategy. Finally, in Sect. 4, we study the minimization of regret when the second player plays a memoryless strategy and in Sect. 5 when he plays a word strategy.

For reasons of space, some claims presented in the sequel are only supported by a short proof sketch. We refer the interested reader to the technical report [12] for the full proofs.

2 Preliminaries

A *weighted arena* is a tuple $G = (V, V_\exists, E, w, v_I)$ where (V, E, w) is an edge-weighted graph (with rational weights), $V_\exists \subseteq V$, and $v_I \in V$ is the initial vertex. For a given $u \in V$ we denote by $\mathbf{succ}(u)$ the set of *successors of u in G*, that is the set $\{v \in V : (u, v) \in E\}$. We assume w.l.o.g. that no vertex is a sink, *i.e.* $\forall v \in V : |\mathbf{succ}(v)| > 0$, and that every Eve vertex has more than one successor, *i.e.* $\forall v \in V_\exists : |\mathbf{succ}(v)| > 1$. In the sequel, we depict vertices in $V_\exists$ with squares and vertices in $V \setminus V_\exists$ with circles. We denote the maximum absolute value of a weight in a weighted arena by $W := \max_{e \in E} |w(e)|$.

A *play* in a weighted arena is an infinite sequence of vertices $\pi = v_0 v_1 \dots$ where $(v_i, v_{i+1}) \in E$ for all $i \geq 0$. Given a play $\pi = v_0 v_1 \dots$ and integers $k \leq l$ we define $\pi[k..l] := v_k \dots v_l$, $\pi[..k] := \pi[0..k]$, and $\pi[l..] := v_l v_{l+1} \dots$. We refer to the latter two as *play prefixes* and *play suffixes*, respectively. To improve readability, we try to adhere to the following convention: use π to denote plays and ρ for play prefixes. The *length of a play* π, denoted $|\pi|$, is ∞, and the *length of a play prefix* $\rho = v_0 \dots v_n$, *i.e.* $|\rho|$, is $n + 1$.

A *strategy for* Eve (Adam) is a function σ that maps play prefixes ending with a vertex v from $V_\exists$ ($V \setminus V_\exists$) to a successor of v. A strategy has memory m if it can be realized as the output of a finite state machine with m states (see *e.g.* [11] for a formal definition). A

memoryless (or positional) strategy is a strategy with memory 1, that is, a function that only depends on the last element of the given partial play. A play $\pi = v_0 v_1 \dots$ is *consistent with a strategy* σ for Eve (Adam) if whenever $v_i \in V_\exists$ ($v_i \in V \setminus V_\exists$), then $\sigma(\pi[..i]) = v_{i+1}$. We denote by $\mathfrak{S}_\exists(G)$ ($\mathfrak{S}_\forall(G)$) the set of all strategies for Eve (Adam) and by $\Sigma^m_\exists(G)$ ($\Sigma^m_\forall(G)$) the set of all strategies for Eve (Adam) in G that require memory of size at most m, in particular $\Sigma^1_\exists(G)$ ($\Sigma^1_\forall(G)$) is the set of all memoryless strategies for Eve (Adam) in G. We omit G if the context is clear.

Given strategies σ, τ, for Eve and Adam respectively, and $v \in V$, we denote by $\pi^v_{\sigma\tau}$ the unique play starting from v that is consistent with σ and τ. If v is omitted, it is assumed to be v_I.

A *weighted automaton* is a tuple $\Gamma = (Q, q_I, A, \Delta, w)$ where A is a finite alphabet, Q is a finite set of states, q_I is the initial state, $\Delta \subseteq Q \times A \times Q$ is the transition relation, $w : \Delta \to \mathbb{Q}$ assigns weights to transitions. A *run* of Γ on a word $a_0 a_1 \dots \in A^\omega$ is a sequence $\psi = q_0 a_0 q_1 a_1 \dots \in (Q \times A)^\omega$ such that $(q_i, a_i, q_{i+1}) \in \Delta$, for all $i \geq 0$, and has *value* $\mathbf{Val}(\psi)$ determined by the sequence of weights of the transitions of the run and the payoff function $\mathbf{Val}$. The value Γ assigns to a word x, *i.e.* $\Gamma(x)$, is the supremum of the values of all runs on the word. We say the automaton is deterministic if Δ is functional (*i.e.* for all $q \in Q$ and all $\sigma \in \Sigma$ we have that $|\{q' \in Q : (q, \sigma, q') \in \Delta\}| = 1$.)

Safety games. A *safety game* is played on a weighted arena by Eve and Adam. The goal of Eve is to perpetually avoid traversing edges from a set of *bad edges*, while Adam attempts to force the play through any unsafe edge. More formally, a safety game is a tuple (G, B) where $G = (V, V_\exists, E, v_I)$ is a weighted arena and $B \subseteq E$ is the set of bad edges. A play $\pi = v_0 v_1 \dots$ is *winning for Eve* if $(v_i, v_{i+1}) \notin B$, for all $i \geq 0$, and it is *winning for Adam* otherwise. A strategy for Eve (Adam) is winning for her (him) in the safety game if all plays consistent with it are winning for her (him). A player wins the safety game if (s)he has a winning strategy.

▶ **Lemma 3** (from [2]). *Safety games are* positionally determined*: either Eve has a positional winning strategy or Adam has a positional winning strategy. Determining the winner in a safety game is decidable in linear time.*

Discounted-sum. A play in a weighted arena, or a run in a weighted automaton, induces an infinite sequence of weights. We define below the *discounted-sum* payoff function which maps finite and infinite sequences of rational weights to real numbers. In the sequel we refer to a weighted arena together with a payoff function as a *game*. Formally, given a sequence of weights $\chi = x_0 x_1 \dots$ of length $n \in \mathbb{N} \cup \{\infty\}$, the *discounted-sum* is defined for a rational discount factor $\lambda \in (0, 1)$ as follows: $\mathsf{DS}_\lambda(\chi) := \sum_{i=0}^{n} \lambda^i x_i$. For convenience, we apply payoff functions directly to plays, runs, and prefixes. For instance, given a play or play prefix $\pi = v_0 v_1 \dots$ we write $\mathsf{DS}_\lambda(\pi)$ instead $\mathsf{DS}_\lambda(w(v_0, v_1) w(v_1, v_2) \dots)$.

Consider a fixed weighted arena G, and a discounted-sum payoff function $\mathbf{Val} = \mathsf{DS}_\lambda$ for some $\lambda \in (0, 1)$. Given strategies σ, τ, for Eve and Adam respectively, and $v \in V$, we denote the value of $\pi^v_{\sigma\tau}$ by $\mathbf{Val}^v_G(\sigma, \tau) := \mathbf{Val}(\pi^v_{\sigma\tau})$. We omit G if it is clear from the context. If v is omitted, it is assumed to be v_I.

Antagonistic & co-operative values. Two values associated with a weighted arena that we will use throughout are the *antagonistic and co-operative values*, defined for plays from a vertex $v \in V$ as:

$$\mathbf{aVal}^v(G) := \sup_{\sigma \in \mathfrak{S}_\exists} \inf_{\tau \in \mathfrak{S}_\forall} \mathbf{Val}^v(\sigma, \tau) \qquad \mathbf{cVal}^v(G) := \sup_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \mathbf{Val}^v(\sigma, \tau).$$

Again, if G is clear from the context it will be omitted, and if v is omitted it is assumed to be v_I.

We note that, as memoryless strategies are sufficient in discounted-sum games [15], determining if $\mathbf{aVal}$ is larger (or smaller) than a given threshold is decidable in $\mathsf{NP} \cap \mathsf{coNP}$. Furthermore, the values $\mathbf{cVal}$ and $\mathbf{aVal}$ are representable using a polynomial number of bits. In [15] it is also shown that $\mathbf{aVal}$ can be computed in time polynomial (in $\frac{1}{1-\lambda}$, $|V|$, and $\log_2 W$). If λ is given in binary as part of the input, this becomes exponential (in the size of the input). Regardless of whether λ is part of the input, $\mathbf{cVal}$ is computable in polynomial time.

A useful observation used by Zwick and Paterson in [15], and which is implicitly used throughout this work, is the following.

▶ Remark. For all $u \in V$, $\mathbf{cVal}^u(G) = \max\{w(u,v) + \lambda\mathbf{cVal}^v(G) : (u,v) \in E\}$. For all $u \in V_\exists$, $\mathbf{aVal}^u(G) = \max\{w(u,v) + \lambda\mathbf{aVal}^v(G) : (u,v) \in E\}$. For all $u \in V \setminus V_\exists$, $\mathbf{aVal}^u(G) = \min\{w(u,v) + \lambda\mathbf{aVal}^v(G) : (u,v) \in E\}$.

We say a strategy σ for Eve is *worst-case optimal (maximizing)* from $v \in V$ if it holds that $\inf_{\tau \in \mathfrak{S}_\forall} \mathbf{Val}^v(\sigma,\tau) = \mathbf{aVal}^v(G)$. Similarly, a strategy τ for Adam is *worst-case optimal (minimizing)* from $v \in V$ if it holds that $\sup_{\sigma \in \mathfrak{S}_\exists} \mathbf{Val}^v(\sigma,\tau) = \mathbf{aVal}^v(G)$. Also, a pair of strategies σ, τ for Eve and Adam, respectively, is said to be *co-operative optimal* from $v \in V$ if $\mathbf{Val}^v(\sigma,\tau) = \mathbf{cVal}^v(G)$.

▶ **Lemma 4** (from [15]). *The following hold:*

- *there exists $\sigma \in \Sigma^1_\exists$ which is worst-case optimal maximizing from all $v \in V$,*
- *there exists $\tau \in \Sigma^1_\forall$ which is worst-case optimal minimizing from all $v \in V$,*
- *there are $\sigma \in \Sigma^1_\exists$ and $\tau \in \Sigma^1_\forall$ which are co-operative optimal from all $v \in V$.*

We now recall the definition of a *strongly co-operative optimal* strategy σ for Eve. Intuitively, such a strategy attempts to maximize the co-operative value. Formally, for any play prefix $\rho = v_0 \dots v_n$ consistent with σ, and such that $v_n \in V_\exists$ if $\sigma(\rho) = v'$, then $v' \in \mathbf{cOpt}(v_n)$; where $\mathbf{cOpt}(u) := \{v \in V : (u,v) \in E \text{ and } \mathbf{cVal}^u(G) = w(u,v) + \lambda\mathbf{cVal}^v(G)\}$. Finally, we define a new type of strategy for Eve: *co-operative worst-case optimal* strategies. A strategy is of this type if it attempts to maximize the co-operative value while achieving at least the antagonistic value. More formally, for any play prefix $\rho = v_0 \dots v_n$ consistent with σ, and such that $v_n \in V_\exists$, if $\sigma(\rho) = v'$ then $v' \in \mathbf{wOpt}(v_n)$ and

$$w(v_n, v') + \lambda\mathbf{cVal}^{v'}(G) = \max\{w(v_n, v'') + \lambda\mathbf{cVal}^{v''}(G) : v'' \in \mathbf{wOpt}(v_n)\},$$

where $\mathbf{wOpt}(u) := \{v \in V : (u,v) \in E \text{ and } \mathbf{aVal}^u(G) = w(u,v) + \lambda\mathbf{aVal}^v(G)\}$.

It is not hard to verify that strategies of the above types always exist for Eve.

▶ **Lemma 5.** *There exist strongly co-operative optimal strategies and co-operative worst-case optimal strategies for Eve.*

Regret. Let $\Sigma_\exists \subseteq \mathfrak{S}_\exists$ and $\Sigma_\forall \subseteq \mathfrak{S}_\forall$ be sets of strategies for Eve and Adam respectively. Given $\sigma \in \Sigma_\exists$ we define the *regret of σ in G w.r.t. $\Sigma_\exists$ and $\Sigma_\forall$* as:

$$\mathbf{reg}^\sigma_{\Sigma_\exists,\Sigma_\forall}(G) := \sup_{\tau \in \Sigma_\forall}(\sup_{\sigma' \in \Sigma_\exists} \mathbf{Val}(\sigma',\tau) - \mathbf{Val}(\sigma,\tau)).$$

A strategy σ for Eve is then said to be *regret-free* w.r.t. $\Sigma_\exists$ and $\Sigma_\forall$ if $\mathbf{reg}^\sigma_{\Sigma_\exists,\Sigma_\forall}(G) = 0$. We define the *regret of G w.r.t. $\Sigma_\exists$ and $\Sigma_\forall$* as:

$$\mathbf{Reg}_{\Sigma_\exists,\Sigma_\forall}(G) := \inf_{\sigma \in \Sigma_\exists} \mathbf{reg}^\sigma_{\Sigma_\exists,\Sigma_\forall}(G).$$

When $\Sigma_\exists$ or $\Sigma_\forall$ are omitted from $\mathbf{reg}(\cdot)$ and $\mathbf{Reg}(\cdot)$ they are assumed to be the set of all strategies for Eve and Adam.

In the unfolded definition of the regret of a game, *i.e.*

$$\mathbf{Reg}_{\Sigma_\exists,\Sigma_\forall}(G) := \inf_{\sigma\in\Sigma_\exists} \sup_{\tau\in\Sigma_\forall}(\sup_{\sigma'\in\Sigma_\exists} \mathbf{Val}(\sigma',\tau) - \mathbf{Val}(\sigma,\tau)),$$

let us refer to the witnesses σ and σ' as the *primary strategy* and the *alternative strategy* respectively. Observe that for any primary strategy for Eve and any one strategy for Adam, we can assume Adam plays to maximize the payoff (*i.e.* co-operates) together with the alternative strategy once it deviates (necessarily at an Eve vertex) or to minimize against the primary strategy – again, once it deviates. Indeed, since the deviation yields different histories, the two strategies for Adam can be combined without conflict. More formally,

▶ **Lemma 6.** *Consider any $\sigma \in \mathfrak{S}_\exists$, $\tau \in \mathfrak{S}_\forall$, and corresponding play $\pi_{\sigma\tau} = v_0v_1\dots$. For all $i \geq 0$ such that $v_i \in V_\exists$, for all $v' \in \mathbf{succ}(v_i) \setminus \{v_{i+1}\}$ there exist $\sigma' \in \mathfrak{S}_\exists$, $\tau' \in \mathfrak{S}_\forall$ for which*

(i) $\pi_{\sigma'\tau}[..i+1] = \pi_{\sigma\tau}[..i]\cdot v'$,
(ii) $\mathbf{Val}(\pi_{\sigma'\tau'}[i+1..]) = \mathbf{cVal}^{v'}(G)$, *and*
(iii) $\pi_{\sigma\tau} = \pi_{\sigma\tau'}$.

Furthermore, from any vertex $v \in V$, Eve has a strategy to ensure a payoff of at least $\mathbf{aVal}^v(G)$ and Adam has a strategy to ensure a payoff of at most $\mathbf{aVal}^v(G)$. Thus, one could further assume that Adam plays to minimize against the primary strategy while maximizing against the alternative one.

▶ **Lemma 7.** *Consider any $\sigma \in \mathfrak{S}_\exists$, $\tau \in \mathfrak{S}_\forall$, and corresponding play $\pi_{\sigma\tau} = v_0v_1\dots$. For all $i \geq 0$ such that $v_i \in V_\exists$, for all $v' \in \mathbf{succ}(v_i) \setminus \{v_{i+1}\}$ there exist $\sigma' \in \mathfrak{S}_\exists$, $\tau' \in \mathfrak{S}_\forall$ for which*

(i) $\pi_{\sigma'\tau}[..i+1] = \pi_{\sigma\tau}[..i]\cdot v' = \pi_{\sigma\tau'}[..i]\cdot v'$,
(ii) $\mathbf{Val}(\pi_{\sigma'\tau'}[i+1..]) = \mathbf{cVal}^{v'}(G)$, *and*
(iii) $\mathbf{Val}(\pi_{\sigma\tau'}[i+1..]) \leq \mathbf{aVal}^{v_{i+1}}(G)$.

Both claims follow from the definitions of strategies for Eve and Adam and from Lemma 4.

In the remaining of this work, we will assume that λ is **not** given as part of the input.

3 Regret against all strategies of Adam

In this section we describe an algorithm to compute the (minimal) regret of a discounted-sum game when there are no restrictions placed on the strategies of Adam. The algorithm can be implemented by an alternating machine guaranteed to halt in polynomial time. We show that the regret *value* of any game is achieved by a strategy for Eve which consists of two strategies, the first choosing edges which lead to the optimal co-operative value, the second choosing edges which ensure the antagonistic value. The switch from the former to the latter is done based on the "local regret" of the vertex (this is formalized in the sequel). The latter allows us to claim NP-membership of the regret threshold problem. The following theorem summarizes the bounds we obtain:

▶ **Theorem 8.** *Deciding if the regret value is less than a given threshold (strictly or non-strictly), playing against all strategies of Adam, is in* NP.

Let us start by formalizing the concept of *local regret*. Given a play or play prefix $\pi = v_0 \dots$ and integer $0 \leq i < |\pi|$ such that $v_i \in V_\exists$, define $\mathbf{locreg}(\pi, i)$ as follows:

$$\begin{cases} \lambda^i \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(G) - \mathbf{Val}(\pi[i..]) \right) & \text{if } \pi \text{ is a play,} \\ \lambda^i \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(G) - \mathbf{Val}(\pi[i..j]) \right) - \lambda^j \mathbf{aVal}^{v_j}(G) & \text{if } \pi \text{ is a prefix of length } j+1 > i+1, \\ \lambda^i \left(\mathbf{cVal}^{v_i}(G) - \mathbf{aVal}^{v_i}(G) \right) & \text{if } \pi \text{ is a prefix of length } i+1, \end{cases}$$

where $\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(G) = \max\{w(v_i, v) + \lambda \mathbf{cVal}^v(G) : (v_i, v) \in E \text{ and } v \neq v_{i+1}\}$. Intuitively, for π a play, $\mathbf{locreg}(\pi, i)$ corresponds to the difference between the value of the best *deviation* from position i and the value of π. For π a play prefix, $\mathbf{locreg}(\pi, i)$ assumes that after position $j = |\pi| - 1$ Eve will play a worst-case optimal strategy.

3.1 Deciding 0-regret

We will now argue that the problem of determining whether Eve has a regret-free strategy can be decided in polynomial time. Furthermore, if no such strategy for Eve exists, we will extract a strategy for Adam which, against any strategy of Eve, ensures non-zero regret. To do so, we will reduce the problem to that of deciding whether Eve wins a safety game. The unsafe edges are determined by a function of the antagonistic and co-operative values of the original game. Critically, the game is played on the same arena as the original regret game.

▶ **Theorem 9.** *Deciding if the regret value is* 0*, playing against all strategies of Adam, is in* PTIME*.*

Proof. We define a partition of the edges leaving vertices from $V_\exists$ into good and bad for Eve. A bad edge is one which witnesses non-zero local regret. We then show that Eve can ensure a regret value of 0 if and only if she has a strategy to avoid ever traversing bad edges. More formally, let us assume a given weighted arena $G = (V, V_\exists, v_I, E, w)$ and a discount factor $\lambda \in (0, 1)$. We define the set of bad edges $\mathcal{B} := \{(u, v) \in E : u \in V_\exists \text{ and } w(u, v) + \lambda \mathbf{aVal}^v(G) < \mathbf{cVal}^u_{\neg v}(G)\}$.

Note that strategies for either player in the newly defined safety game are also strategies for them in the original game (and vice versa as well). We now claim that winning strategies for Adam in the safety game $\hat{G} = (V, V_\exists, v_I, E, \mathcal{B})$ ensure that, regardless of the strategy of Eve, its regret will be strictly positive. The idea behind the claim is that, Adam can force to traverse a bad edge and from there, play adversarially against the primary strategy and co-operatively with an alternative strategy.

▶ **Claim 10.** *If $\tau \in \mathfrak{S}_\forall$ is a winning strategy for Adam in $\hat{G}$, then there exist $\tau' \in \mathfrak{S}_\forall$ and $\sigma' \in \mathfrak{S}_\exists$ such that $\forall \sigma \in \mathfrak{S}_\exists : \mathbf{Val}(\sigma', \tau') - \mathbf{Val}(\sigma, \tau') \geq \lambda^{|V|} \min\{\mathbf{cVal}^u_{\neg v}(G) - w(u, v) - \lambda \mathbf{aVal}^v(G) : (u, v) \in \mathcal{B} \text{ and } u \in V_\exists\} > 0$.*

The claim follows from the definitions and Lemma 7. Conversely, winning strategies for Eve in $\hat{G}$ are actually regret-free.

▶ **Claim 11.** *If $\sigma \in \mathfrak{S}_\exists$ is a winning strategy for Eve in $\hat{G}$, then $\mathbf{reg}^\sigma(G) = 0$.*

Our argument to prove this claim requires we first show that a winning strategy for Eve ensures the antagonistic value of G from v_I.

The desired result then follows from Lemma 3 and from the fact that membership of an edge in $\mathcal{B}$ can be decided by computing **cVal** and a threshold query regarding **aVal**, thus in polynomial time. ◀

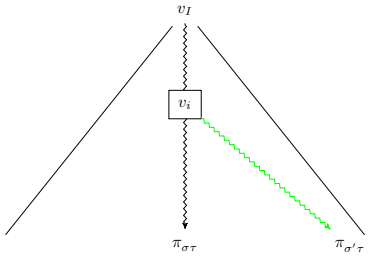

Figure 3 Depiction of a play and a "better alternative play".

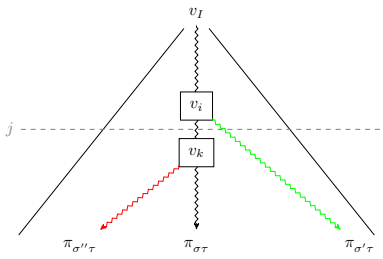

Figure 4 A deviation from v_k cannot be a better alternative to $\pi_{\sigma\tau}$ if $j \geq N(\mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau))$.

We observe that the proof of Theorem 9 – more precisely, Claim 10 – implies that, if there is no regret-free strategy for Eve in a game, then the regret of the game is at least $\lambda^{|V|}$ times the smallest local regret labelling the bad edge from $\mathcal{B}$ which Adam can force. More formally:

▶ **Corollary 12.** *If no regret-free strategy for Eve exists in G, then $\mathbf{Reg}(G) \geq a_G$ where $a_G := \lambda^{|V|} \min\{\mathbf{locreg}(uv, 0) : u \in V_\exists \text{ and } (u, v) \in \mathcal{B}\}$.*

3.2 Deciding r-regret

It will be useful in the sequel to define the *regret of a play* and the *regret of a play prefix*. Given a play $\pi = v_0 v_1 \dots$, we define the regret of π as:

$$\mathbf{reg}(\pi) := \sup\left(\{\mathbf{locreg}(\pi, i) : v_i \in V_\exists\} \cup \{0\}\right).$$

Intuitively, the local regrets give lower bounds for the overall regret of a play. We will also let the *regret of a play prefix* $\rho = v_0 \dots v_j$ be equal to

$$\max\left(\{\lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(G) - \mathbf{Val}(\rho[i..j])) : 0 \leq i < j \text{ and } v_i \in V_\exists\} \cup \{0\}\right).$$

Let us give some more intuition regarding the regret of a play. Consider a pair of strategies σ and τ for Eve and Adam, respectively. Suppose there is an alternative strategy σ' for Eve, such that, against τ, the obtained payoff is greater than that of $\pi_{\sigma\tau}$. It should be clear that this implies there is some position i such that, from vertex $v_i \in V_\exists$ σ' and τ result in a different play from $\pi_{\sigma\tau}$ (see Figure 3). We will sometimes refer to this deviation, *i.e.* the play $\pi_{\sigma'\tau}$, as a *better alternative* to $\pi_{\sigma\tau}$.

We can now show the regret of a strategy for Eve in fact corresponds to the supremum of the regret of plays consistent with the strategy.

▶ **Lemma 13.** *For any strategy σ of Eve, $\mathbf{reg}^\sigma(G) = \sup\{\mathbf{reg}(\pi) : \pi \text{ is consistent with } \sigma\}$.*

We note that for any play π, the sequence $\langle \lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(G) - \mathbf{Val}(\pi[i..]))\rangle_{i \geq 0}$ converges to 0 because $(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(G) - \mathbf{Val}(\pi[i..]))$ is bounded by $\frac{2W}{(1-\lambda)}$. It follows that if we have a non-zero lower bound for the regret of π, then there is some index N such that the witness for the regret occurs before N. Moreover, we can place a polynomial upper bound on N. More precisely:

▶ **Lemma 14.** *Let π be a play in G and suppose $0 < r \leq \mathbf{reg}(\pi)$. Let*

$$N(r) := \lfloor (\log r + \log(1 - \lambda) - \log(2W)) / \log \lambda \rfloor + 1.$$

Then $\mathbf{reg}(\pi) = \mathbf{reg}(\pi[..N(r)]) - \lambda^{N(r)} \mathbf{Val}(\pi[N(r)..])$.

The above result gives us a bound on how far we have to unfold a game after having witnessed a non-zero lower bound, r, for the regret. If we consider the example from Figure 3, this translates into a bound on how many turns after v_i a deviation can still yield bigger local regret (see Figure 4).

Corollary 12 then gives us the required lower bound to be able to use Lemma 14.

▶ **Lemma 15.** *If* $\mathbf{Reg}(G) \geq a_G$ *then* $\mathbf{Reg}(G)$ *is equal to*

$$\inf_{\sigma \in \mathfrak{S}_\exists} \sup\{\mathbf{reg}(\pi[..N(a_G)]) - \lambda^{N(a_G)}\mathbf{aVal}^{v_{N(a_G)}}(G) : \pi = v_0 v_1 \dots \text{ is consistent with } \sigma\}.$$

This already implies we can compute the regret value in alternating polynomial time (or equivalently, deterministic polynomial space [5]).

▶ **Proposition 16.** *The regret value is computable using only polynomial space.*

Proof. We first label the arena with the antagonistic and co-operative values and solve the safety game described for Theorem 9. The latter can be done in polynomial time. If the Eve wins the safety game, the regret value is 0. Otherwise, we know $a_G > 0$ is a lower bound for the regret value. We now simulate G using an alternating Turing machine which halts in at most $N(a_G)$ steps. That is, a polynomial number of steps. The simulated play prefix is then assigned a regret value as per Lemma 15 (recall we have already pre-computed the antagonistic value of every vertex). ◀

As a side-product of the algorithm described in the above proof we get that finite memory strategies suffice for Eve to minimize her regret in a discounted-sum game.

▶ **Corollary 17.** *Let* $\mu := |\Delta|^{N(a_G)}$*, with* $N(0) = 0$*. It holds that*

$$\mathbf{Reg}_{\Sigma^\mu_\exists, \mathfrak{S}_\forall}(G) = \mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{S}_\forall}(G).$$

3.3 Simple regret-minimizing behaviours

We will now argue that Eve has a simple strategy which ensures regret of at most $\mathbf{Reg}(G)$. Her strategy will consist of "playing co-operatively" for some turns (until a high local regret has already been witnessed) and then switch to a co-operative worst-case optimal strategy.

We will now define a family of strategies which switch from co-operative behaviour to antagonistic, after a specific number of turns have elapsed (in fact, enough for the discounted local regret to be less than the desired regret). Denote by σ^{co} a strongly co-operative strategy for Eve in G and by σ^{cw} a co-operative worst-case optimal strategy for Eve in G. Recall that, by Lemma 5, such strategies for her always exist. Finally, given a co-operative strategy σ^{co}, a co-operative worst-case optimal strategy σ^{cw}, and $t \in \mathbb{Q}$ let us define an *optimistic-then-pessimistic strategy for Eve* $[\sigma^{\mathsf{co}} \xrightarrow{\mathrm{t}} \sigma^{\mathsf{cw}}]$. The strategy is such that, for any play prefix $\rho = v_0 \dots v_n$ such that $v_n \in V_\exists$

$$[\sigma^{\mathsf{co}} \xrightarrow{\mathrm{t}} \sigma^{\mathsf{cw}}](\rho) = \begin{cases} \sigma^{\mathsf{co}}(\rho) & \text{if } |\mathbf{cOpt}(v_n)| = 1 \text{ and } \mathbf{locreg}(\rho \cdot \sigma^{\mathsf{cw}}(\rho), n+1) > t \\ \sigma^{\mathsf{cw}}(\rho) & \text{otherwise.} \end{cases}$$

We claim that, when we set $t = \mathbf{Reg}(G)$, an optimistic-then-pessimistic strategy for Eve ensures minimal regret. That is

▶ **Proposition 18.** *Let* σ^{co} *be a strongly co-operative strategy for Eve,* σ^{cw} *be a Eve and a co-operative worst-case optimal strategy for Eve, and* $t = \mathbf{Reg}(G)$*. The strategy* $\sigma = [\sigma^{\mathsf{co}} \xrightarrow{\mathrm{t}} \sigma^{\mathsf{cw}}]$ *has the property that* $\mathbf{reg}^\sigma(G) = \mathbf{Reg}(G)$*.*

This is a refinement of the strategy one can obtain from applying the algorithm used to prove Proposition 16.[1] The latter tells us that a regret-minimizing strategy of Eve eventually switches to a worst-case optimal behaviour. For vertices where, before this switch, another edge was chosen by Eve, we argue that she must have been playing a co-operative strategy. Otherwise, she could have switched sooner.

We have shown the regret value can be computed using an algorithm which requires polynomial space only. This algorithm is based on a polynomial-length unfolding of the game and from it we can deduce that the regret value is representable using a polynomial number of bits. (Indeed, all exponents occurring in the formula from Lemma 15 will be polynomial according to Lemma 14.) Also, we have argued that Eve has a "simple" strategy σ to ensure minimal regret. Such a strategy is defined by two polynomial-time constructible sub-strategies and the regret value of the game. Hence, it can be encoded into a polynomial number of bits itself. Furthermore, σ is guaranteed to be playing as its co-operative worst-case optimal component after $N(\mathbf{Reg}(G))$ turns (see, again, Lemma 14), which is a polynomial number of turns. Given a regret threshold r, we claim we can verify that σ ensures regret at most r in polynomial time. This can be achieved by allowing Adam to play in G, and against σ, with the objective of reaching an edge with high local regret before $N(\mathbf{Reg}(G))$ turns. A possible formalization of this idea follows. Consider the product of G with a counter ranging from 1 to $N(\mathbf{Reg}(G))$ where we make all vertices belong to Adam. In this game H, we make edges leaving vertices previously belonging to Eve go to a sink and define a new weight function w' which assigns to these edges their negative non-discounted local regret: going from u to v when σ dictates to go to v' yields $w(u,v')+\lambda\mathbf{aVal}^{v'}(H\times\sigma)-w(u,v)+\lambda\mathbf{cVal}^{v}(H)$. Lemma 15 allows us to show that σ ensures regret at most r in G if and only if the antagonistic value of a discounted-sum game played on H with weight function w' is at most $-r$.

It follows that the regret threshold problem is in NP, as stated in Theorem 8.

▶ **Example 19.** We revisit the discounted-sum game from Figure 1. Let us instantiate the values $M = 100$ and $\lambda = \frac{9}{10}$. According to our previous remarks on this arena, after i visits to v without Adam choosing (v, y), Eve could achieve $\left(\frac{1}{1-\frac{9}{10}}\right)\left(\frac{9}{10}\right)^{2i} = 10\left(\frac{9}{10}\right)^{2i}$, by going to x, or hope for $\left(\frac{100}{1-\frac{9}{10}}\right)\left(\frac{9}{10}\right)^{2i+1} = 1000\left(\frac{9}{10}\right)^{2i+1}$ by going to v again. Her best regret minimizing strategy corresponds to σ^{22} which ensures regret of at most $10 - 10(\frac{9}{10})^{44} \approx 9.9030$. (Note that, since $1000(\frac{9}{10})^{2j+1} < 10$ for all $j \geq 22$, the best alternative strategy indeed achieves a payoff of 10 only. This alternative strategy corresponds to having gone to x from the beginning.)

It is easy to see that Eve cannot win the safety game $\hat{G}$ constructed from this arena. Indeed, from the initial vertex v_I both (v_I, x) and (v_I, v) are bad edges for Eve since there are always alternative strategies to obtain higher payoffs (that is, if Adam does not play to y). More formally, we note that $\mathbf{cVal}^{v_I}_{\neg x} = 1000\lambda$, $\mathbf{cVal}^{v_I}_{\neg v} = 10$, $\mathbf{aVal}^{x} = 10$, and $\mathbf{aVal}^{v} = 10\lambda^2$. Thus, the lower bound a_G one can obtain from $\hat{G}$ is then equal to

$$\min\left\{1000\left(\frac{9}{10}\right) - 10, 10 - 10\left(\frac{9}{10}\right)^2\right\}\left(\frac{9}{10}\right)^4 = 10\left(\frac{9}{10}\right)^4 - 10\left(\frac{9}{10}\right)^6 \approx 1.2466.$$

As expected, when Eve plays her optimal regret-minimizing (optimistic-then-pessimistic) strategy any better alternative must deviate before $N(a_G) = 71$ turns. In general, against σ^i, for $i < 22$ a regret bigger than 9.9030 is obtained by Adam choosing the edge (v, y) to help

[1] In fact, our proof of Prop. 18 relies on Eve requiring finite memory, to minimize her regret.

any strategy of Eve going to v more than i times, for $i \geq 22$ choices of Adam are no longer relevant and the best alternative strategy for Eve is to have gone to x from the first step. ◀

4 Regret against positional strategies of Adam

In this section we consider the problem of computing the (minimal) regret when Adam is restricted to playing positional strategies.

▶ **Theorem 20.** *Deciding if the regret value is less than a given threshold (strictly or non-strictly), playing against positional strategies of Adam, is in* PSPACE.

Playing against an Adam, when he is restricted to playing memoryless strategies gives Eve the opportunity to learn some of Adam's strategic choices. However, due to its decaying nature, with the discounted-sum payoff function Eve must find a balance between exploring too quickly, thereby presenting lightly discounted alternatives; and learning too slowly, thereby heavily discounting her eventual payoff.

A similar approach to the one we have adopted in Section 3 can be used to obtain an algorithm for this setting. The claimed lower bound follows from Theorem 26.

4.1 Deciding 0-regret

As in the previous section, we will reduce the problem of deciding if the game has regret value 0 to that of determining the winner of a safety game. It will be obvious that if no regret-free strategy for Eve exists in the original game, then we can construct, for any strategy of hers, a positional strategy of Adam which ensures non-zero regret. Hence, we will also obtain a lower bound on the regret of the game in the case Adam wins the safety game.

Let us fix some notation. For a set of edges $D \subseteq E$, we denote by $G\lfloor D$ the weighted arena $(V, V_\exists, v_I, D, w)$. Also, for a positional strategy $\tau : (V \setminus V_\exists) \to E$ for Adam in G, we denote by $G \times \tau$ the weighted arena resulting from removing all edges not consistent with τ. Next, for an edge $(s,t) \in E$ we define $E_\forall(st) := \{(u,v) \in E : \text{if } u = s \text{ then } v = t \text{ or } u \in V_\exists\}$. We extend the latter to play prefixes $\rho = v_0 \dots v_n$ by (recursively) defining $E_\forall(\rho) := E_\forall(\rho[..n-1]) \cap E_\forall(v_{n-1}v_n)$. If π is a play, then $E \supseteq E_\forall(\pi[..i]) \supseteq E_\forall(\pi[..j])$ for all $0 \leq i \leq j$. Hence, since E is finite, the value $E_\forall(\pi) := \lim_{i \geq 0} E_\forall(\pi[..i])$ is well-defined. Remark that $E_\forall(\pi)$ does not restrict edges leaving vertices of Eve. The following properties directly follow from our definitions.

▶ **Lemma 21.** *Let π be a play or play prefix consistent with a positional strategy for Adam. It then holds that:*

(i) *for every $v \in V \setminus V_\exists$ there is some edge $(v, \cdot) \in E_\forall(\pi)$,*
(ii) *π is consistent with a strategy $\tau \in \Sigma^1_\forall(G)$ if and only if $\tau \in \Sigma^1_\forall(G\lfloor E_\forall(\pi))$, and*
(iii) *every strategy $\tau \in \Sigma^1_\forall(G\lfloor E_\forall(\pi))$ is also an element from $\Sigma^1_\forall(G)$.*

To be able to decide whether regret-free strategies for Eve exist, we define a new safety game. The arena we consider is $\hat{G} := (\hat{V}, \hat{V_\exists}, \hat{v_I}, \hat{E})$ where $\hat{V} := V \times \mathcal{P}(E)$, $\hat{V_\exists} := V_\exists \times \mathcal{P}(E)$, $\hat{v_I} := (v_I, E)$, and $\hat{E}$ contains the edge $((u,C),(v,D))$ if and only if $(u,v) \in E$ and $D = C \cap E_\forall(uv)$.

▶ **Theorem 22.** *Deciding if the regret value is* 0*, playing against positional strategies of Adam, is in* PSPACE.

Proof. A safety game is constructed as in the proof of Theorem 9. Here, we consider $\tilde{G}$ and the set of bad edges $\tilde{\mathcal{B}} := \{((u,C),(v,D)) \in \hat{E} : u \in V_\exists \text{ and } \exists \tau \in \Sigma^1_\forall(G\lfloor C), w(u,v) +$

$\lambda\mathbf{cVal}^v(G \times \tau) < \mathbf{cVal}^u_{\neg v}(G \times \tau)\}$. We then have the safety game $\tilde{G} = (\hat{V}, \hat{V}_\exists, \hat{v_I}, \hat{E}, \tilde{\mathcal{B}})$. Note that there is an obvious bijective mapping from plays (and play prefixes) in $\tilde{G}$ to plays (prefixes) in G which are consistent with a positional strategy for Adam. One can then show the following properties hold:

▶ **Claim 23.** *If $\tau \in \mathfrak{S}_\forall(\tilde{G})$ is a winning strategy for Adam in $\tilde{G}$, then for all $\sigma \in \mathfrak{S}_\exists(G)$, there exist $t_{\tau\sigma} \in \Sigma^1_\forall(G)$ and $s_{\tau\sigma} \in \mathfrak{S}_\exists(G)$ such that $\mathbf{Val}(s_{\tau\sigma}, t_{\tau\sigma}) - \mathbf{Val}(\sigma, t_{\tau\sigma}) \geq \lambda^{|V|(|E|+1)}$ $\min\{\mathbf{cVal}^u_{\neg v}(G \times \tau) - w(u,v) - \lambda\mathbf{cVal}^v(G \times \tau) : ((u,C),(v,D)) \in \tilde{\mathcal{B}}, \tau \in \Sigma^1_\forall(G\lfloor C)\}$.*

The claim follows from *positional determinacy* of safety games and Lemma 21.

▶ **Claim 24.** *If $\sigma \in \mathfrak{S}_\exists(\tilde{G})$ is a winning strategy for Eve in $\tilde{G}$, then there is $s_\sigma \in \mathfrak{S}_\exists(G)$ such that $\mathbf{reg}^{s_\sigma}_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G) = 0$.*

It then follows from the determinacy of safety games that Eve wins the safety game $\tilde{G}$ if and only if she has a regret-free strategy.

We observe that simple cycles in $\tilde{G}$ have length at most $|V|(|E|+1)$. Thus, we can simulate the safety game until we complete a cycle and check that all traversed edges are good, all in alternating polynomial time. Indeed, an alternating Turing machine can simulate the cycle and then (universally) check that for all edges, for all positional strategies of the Adam, the inequality holds. ◀

▶ **Corollary 25.** *If no regret-free strategy for Eve exists in G, then $\mathbf{Reg}_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G) \geq b_G$ where $b_G := \lambda^{|V|(|E|+1)} \min\{\mathbf{cVal}^u_{\neg v}(G \times \tau) - w(u,v) - \lambda\mathbf{cVal}^v(G \times \tau) : ((u,C),(v,D)) \in \tilde{\mathcal{B}}$ and $\tau \in \Sigma^1_\forall(G\lfloor C)\}$.*

4.2 Lower bounds

We claim that both 0-regret and r-regret are coNP-hard. This can be shown by adapting the reduction from 2-disjoint-paths given in [13] to the regret threshold problem against memoryless adversaries.

▶ **Theorem 26.** *Let $\lambda \in (0,1)$ and $r \in \mathbb{Q}$ be fixed. Deciding if the regret value is less than r (strictly or non-strictly), playing against positional strategies of Adam, is coNP-hard.*

5 Playing against word strategies of Adam

In this section, we consider the case where Adam is restricted to playing *word strategies*. First, we show that the regret threshold problem can be solved whenever the discounted sum automata associated to the game structure can be made deterministic. As the determinization problem for discounted sum automata has been solved in the literature for only sub-classes of discount factors, and left open in the general case, we complement this result by two other results. First, we show how to solve an *ε-gap promise* variant of the regret threshold problem, and second, we give an algorithm to solve the 0 regret problem. In the two cases, we obtain completeness results on the computational complexities of the problems.

5.1 Preliminaries

The formal definition of the ε-gap promise problem is given below. We first define here the necessary vocabulary. We say that a strategy of Adam is a *word strategy* if his strategy can be expressed as a function $\tau : \mathbb{N} \to [\max\{\mathbf{deg}^+(v) : v \in V\}]$, where $[n] = \{i : 1 \leq i \leq n\}$ and $\mathbf{deg}^+(v)$ is the *outdegree* of v (i.e. the number of edges leaving v). Intuitively, we consider an

order on the successors of each Adam vertex. On the i-th turn, the strategy τ of Adam will tell him to move to the $\tau(i)$-th successor of the vertex according to the fixed order. We denote by $\mathfrak{W}_\forall$ the set of all such strategies for Adam. A game in which Adam plays word strategies can be reformulated as a game played on a weighted automaton $\Gamma = (Q, q_I, A, \Delta, w)$ and strategies of Adam– of the form $\tau : \mathbb{N} \to A$ – determine a sequence of input symbols, i.e. an omega word, to which Eve has to react by choosing Δ-successor states starting from q_I. In this setting a strategy of Eve which minimizes regret defines a run by resolving the non-determinism of Δ in Γ, and ensures the difference of value given by the constructed run is minimal w.r.t. to the value of the best run on the word spelled out by Adam.

5.2 Deciding 0-regret

We will now show that if the regret of an arena (or automaton) is 0, then we can construct a memoryless strategy for Eve which ensures no regret is incurred. More specifically, assuming the regret is 0, we have the existence of a family of strategies of Eve which ensure decreasing regret (with limit 0). We use this fact to choose a small enough ε and the corresponding strategy of hers from the aforementioned family to construct a memoryless strategy for Eve with nice properties which allow us to conclude that its regret is 0. Hence, it follows that an automaton has zero regret if and only if a memoryless strategy of Eve ensures regret 0. As we can guess such a strategy and easily check if it is indeed regret-free (using the obvious reduction to non-emptiness of discounted-sum automata or one-player discounted-sum games), the problem is in NP. A matching lower bound follows from a reduction from SAT which was first described in [1].

▶ **Theorem 27.** *Deciding if the regret value is* 0*, playing against word strategies of Adam, is* NP*-complete.*

5.3 Deciding r-regret: determinizable cases

When the weighted automaton Γ associated to the game structure can be made deterministic, we can solve the regret threshold problem with the following algorithm. In [13] we established that, against eloquent adversaries, computing the regret reduced to computing the value of a quantitative simulation game as defined in [6]. The game is obtained by taking the product of the original automaton and a deterministic version of it. The new weight function is the difference of the weights of both components (for each pair of transitions). In [4], it is shown how to determinize discounted-sum automata when the discount factor is of the form $\frac{1}{n}$, for $n \in \mathbb{N}$. So, for this class of discount factor, we can state the following theorem:

▶ **Theorem 28.** *Deciding if the regret value is less than a given threshold (strictly or non-strictly), playing against word strategies of Adam, is in* EXP *for λ of the form $\frac{1}{n}$.*

5.4 The ε-gap promise problem

Given a discounted-sum automaton $\mathcal{A}$, $r \in \mathbb{Q}$, and $\varepsilon > 0$, the ε-gap promise problem adds to the regret threshold problem the hypothesis that $\mathcal{A}$ will either have regret $\leq r$ or $> r + \varepsilon$. We observe that an algorithm that satisfies that:

- a YES answer implies that $\mathbf{Reg}_{\Sigma_\exists, \mathfrak{W}_\forall}(\mathcal{A}) \leq r + \varepsilon$,
- whereas a NO answer implies $\mathbf{Reg}_{\Sigma_\exists, \mathfrak{W}_\forall}(\mathcal{A}) > r$.

will decide the ε-gap promise problem.

In [4], it is shown that there are discounted-sum automata which define functions that cannot be realized with deterministic-sum automata. Nevertheless, it is also shown in that

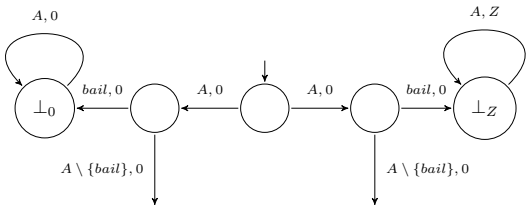

Figure 5 Initial gadget used in reduction from QBF.

paper that given a discounted-sum automaton it is always possible to construct a deterministic one that is ε-close in the following formal sense. A *discounted-sum automaton $\mathcal{A}$ is ε-close to another discounted sum automaton $\mathcal{B}$*, if for all words x the absolute value of the difference between the values assign by $\mathcal{A}$ and $\mathcal{B}$ to x is at most ε. So, it should be clear that we can apply the algorithm underlying Theorem 28 to Γ and a determinized version $\mathcal{D}_\Gamma$ of it (which is ε-close to Γ) and solve the ε-gap promise problem. We can then prove the following result.

▶ **Theorem 29.** *Deciding the ε-gap regret problem is in* PSPACE.

The complexity of the algorithm follows from the fact that the value of a (quantitative simulation) game, played on the product of Γ and $\mathcal{D}_\Gamma$ we described above, can be determined by simulating the game for a polynomial number of turns. Thus, although the automaton constructed using the techniques of Boker and Henzinger [4] is of size exponential, we can construct it "on-the-fly" for the required number of steps and then stop.

Proof Sketch. We reduce the problem to determining the winner of a reachability game on an exponentially larger arena. Although the arena is exponentially larger, all paths are only polynomial in length, so the winner can be determined in alternating polynomial time, or equivalently, polynomial space.

The idea of the construction is as follows. Given a discounted-sum automaton $\mathcal{A}$, we determinize its transitions via a subset construction, to obtain a deterministic, multi-valued discounted-sum automaton $D_\mathcal{A}$. Then we decide if Eve is able to simulate, within the regret bound, the $D_\mathcal{A}$ on $\mathcal{A}$ for all *finite* words up to a length (polynomially) dependent on ε. If we simulate the automaton for a sufficient number of steps, then any significant gap between the automata will be unrecoverable regardless of future inputs, and we can give a satisfactory answer for the ε-GAP REGRET PROBLEM. More specifically, we only have to simulate this determinization process for N steps, where $N := \lfloor(\log\varepsilon + \log(1-\lambda) - \log(4W))/\log\lambda\rfloor + 1$. ◀

5.5 Lower bounds

We claim the ε-gap promise problem is PSPACE-hard even if both λ and ε are not part of the input. To establish the result, we give a reduction from QSAT which uses the gadgets depicted in Figures 5 and 6.

▶ **Theorem 30.** *Let $\lambda \in (0,1)$ and $\varepsilon \in (0,1)$ be fixed. As input, assume we are given $r \in \mathbb{Q}$ and weighted arena $\mathcal{A}$ such that $\mathbf{Reg}_{\Sigma_\exists,\mathfrak{W}_\forall}(\mathcal{A}) \leq r$ or $\mathbf{Reg}_{\Sigma_\exists,\mathfrak{W}_\forall}(\mathcal{A}) > r + \varepsilon$. Deciding if the regret value is less than a given threshold, playing against word strategies of Adam, is* PSPACE*-hard.*

Proof Sketch. We will reduce the QSAT PROBLEM to the present one.

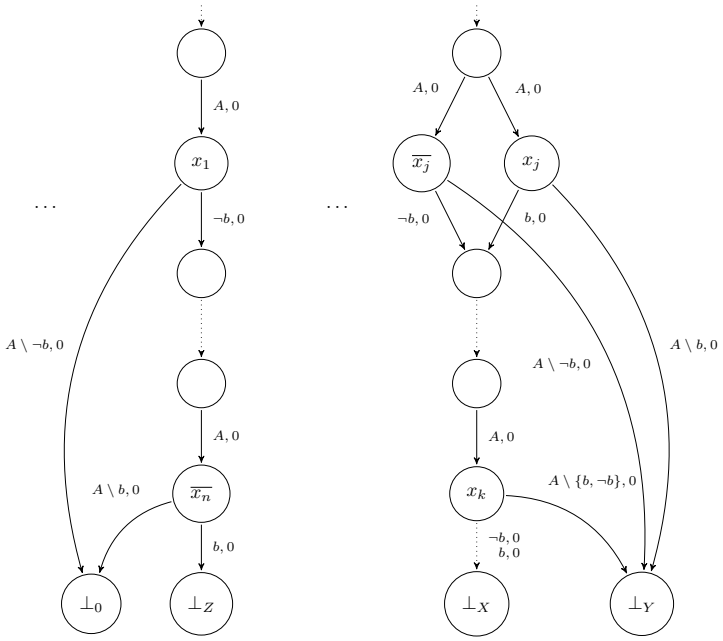

Figure 6 Left and right sub-arenas of the reduction from QBF. Clause i shown on the left; existential and universal gadgets for variables x_j and x_k, respectively, on the right.

The QSAT Problem asks whether a given fully quantified Boolean formula (QBF) is satisfiable. The problem is known to be PSPACE-complete [9]. It is known the result holds even if the formula is assumed to be in conjunctive normal form with three literals per clause (also known as 3-CNF). Therefore, w.l.o.g., we consider an instance of the QSAT Problem to be given in the form:

$$\exists x_0 \forall x_1 \exists x_2 \ldots \Phi(x_0, x_1, \ldots, x_n)$$

where Φ is in 3-CNF.

Our reduction works for values of r, X, Y, and Z such that

(i) $\lambda^2 Z > (r + \varepsilon)(1 - \lambda)$,
(ii) $\lambda^{2n}(Z - X) > (r + \varepsilon)(1 - \lambda)$,
(iii) $\lambda^{2n}(Z - Y) \leq r(1 - \lambda)$,
(iv) $\lambda^3 Y - \lambda^{2n} X \leq r(1 - \lambda)$.

The alphabet of the new weighted arena is $A = \{bail, b, \neg b\}$.

Intuitively, the construction works as follows. The initial gadget shown in Figure 5, ensures that Eve always plays to the rightmost sub-arena. In this part of the game, she has, for every existentially quantified variable, a choice which corresponds to setting the variable to a value of true or false. Depending on her choice, Adam can either play a specific symbol corresponding to him accepting the valuation or stop the simulation and go to $\bot_Y$ where Eve wins the regret game. For universally quantified variables, Eve has no choice (see right part of Fig. 6). The crucial idea is that Adam accepts or rejects a valuation of a variable by playing letter b or $\neg b$. These letters can only be read by a clause gadget if the literal does not make the clause true.

If the QBF is true, Eve will be able to make sure no alternative play in the left side of the game leads to $\bot_Z$. This is the case since the construction ensures that every clause gadget (see left part of Fig. 6) with a literal set to true cannot be used to reach $\bot_Z$. ◀

It follows that the general problem is also PSPACE-hard (even if ε is set to 0).

▶ **Corollary 31.** *Let $\lambda \in (0,1)$. For $r \in \mathbb{Q}$, weighted arena G, determining whether $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(G) \lhd r$, for $\lhd \in \{<,\leq\}$, is* PSPACE*-hard.*

References

1. Benjamin Aminof, Orna Kupferman, and Robby Lampert. Reasoning about online algorithms with weighted automata. *ACM Transactions on Algorithms*, 2010. doi:10.1145/1721837.1721844.
2. Krzysztof R. Apt and Erich Grädel. *Lectures in game theory for computer scientists*. Cambridge University Press, 2011.
3. David E. Bell. Regret in decision making under uncertainty. *Operations Research*, 30(5):961–981, 1982. doi:10.1287/opre.30.5.961.
4. Udi Boker and Thomas A. Henzinger. Exact and approximate determinization of discounted-sum automata. *LMCS*, 10(1), 2014. doi:10.2168/LMCS-10(1:10)2014.
5. Ashok K. Chandra, Dexter Kozen, and Larry J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981. doi:10.1145/322234.322243.
6. Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Transactions on Computational Logic*, 11(4), 2010. doi:10.1145/1805950.1805953.
7. Werner Damm and Bernd Finkbeiner. Does it pay to extend the perimeter of a world model? In *FM*, volume 6664 of *LNCS*, pages 12–26. Springer, 2011. doi:10.1007/978-3-642-21437-0_4.
8. Emmanuel Filiot, Tristan Le Gall, and Jean-François Raskin. Iterated regret minimization in game graphs. In *MFCS*, volume 6281 of *LNCS*, pages 342–354. Springer, 2010.
9. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
10. Thomas A. Henzinger and Nir Piterman. Solving games without determinization. In *CSL*, pages 395–410, 2006. doi:10.1007/11874683_26.
11. Paul Hunter, Guillermo A. Pérez, and Jean-François Raskin. Mean-payoff games with partial-observation – (extended abstract). In *RP*, pages 163–175, 2014. doi:10.1007/978-3-319-11439-2_13.
12. Paul Hunter, Guillermo A. Pérez, and Jean-François Raskin. Minimizing regret in discounted-sum games. *CoRR*, abs/1511.00523, 2015. URL: http://arxiv.org/abs/1511.00523.
13. Paul Hunter, Guillermo A Pérez, and Jean-François Raskin. Reactive synthesis without regret. *Acta Informatica*, pages 1–37, 2015.
14. Daniel Dominic Sleator and Robert Endre Tarjan. Amortized efficiency of list update rules. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 – May 2, 1984, Washington, DC, USA*, pages 488–492. ACM, 1984.
15. Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *TCS*, 158(1):343–359, 1996.