Acta Informatica (2017) 54:3–39
DOI 10.1007/s00236-016-0268-z

ORIGINAL ARTICLE

Reactive synthesis without regret

Paul Hunter[1] · **Guillermo A. Pérez**[1] ·
Jean-François Raskin[1]

Received: 29 November 2015 / Accepted: 13 April 2016 / Published online: 27 April 2016

Abstract Two-player zero-sum games of infinite duration and their quantitative versions are used in verification to model the interaction between a controller (Eve) and its environment (Adam). The question usually addressed is that of the existence (and computability) of a strategy for Eve that can maximize her payoff against any strategy of Adam. In this work, we are interested in strategies of Eve that minimize her regret, i.e. strategies that minimize the difference between her actual payoff and the payoff she could have achieved if she had known the strategy of Adam in advance. We give algorithms to compute the strategies of Eve that ensure minimal regret against an adversary whose choice of strategy is (1) unrestricted, (2) limited to positional strategies, or (3) limited to word strategies, and show that the two last cases have natural modelling applications. These results apply for quantitative games defined with the classical payoff functions Inf, Sup, LimInf, LimSup, and mean-payoff. We also show that our notion of regret minimization in which Adam is limited to word strategies generalizes the notion of good for games introduced by Henzinger and Piterman, and is related to the notion of determinization by pruning due to Aminof, Kupferman and Lampert.

1 Introduction

The model of two player games played on graphs is an adequate mathematical tool to solve important problems in computer science, and in particular the reactive-system synthesis problem [26]. In that context, the game models the non-terminating interaction between the system to synthesize and its environment. Games with quantitative objectives are useful to formalize important quantitative aspects such as mean-response time or energy consumption. They have attracted large attention recently, see e.g. [6,10]. Most of the contributions in

Work partially supported by the ERC inVEST (279499) project.
G. A. Pérez supported by F.R.S.-FNRS fellowship.

✉ Guillermo A. Pérez
gperezme@ulb.ac.be

[1] Département d'Informatique, Université Libre de Bruxelles (ULB), Brussels, Belgium

this context are for zero-sum games: the objective of Eve (that models the system) is to maximize the value of the game while the objective of Adam (that models the environment) is to minimize this value. This is a worst-case assumption: because the cooperation of the environment cannot be assumed, we postulate that it is *antagonistic*.

In this antagonistic approach, the main solution concept is that of a *winning strategy*. Given a threshold value, a winning strategy for Eve ensures a minimal value greater than the threshold against any strategy of Adam. However, sometimes there are no winning strategies. What should the behaviour of the system be in such cases? There are several possible answers to this question. One is to consider *non-zero sum* extensions of those games: the environment (Adam) is not completely antagonistic, rather it has its own specification. In such games, a strategy for Eve must be winning only when the outcome satisfies the objectives of Adam, see e.g. [8]. Another option for Eve is to play a strategy which minimizes her *regret*. The regret is informally defined as the difference between what a player actually wins and what she could have won if she had known the strategy chosen by the other player. Minimization of regret is a central concept in decision theory [3]. This notion is important because it usually leads to solutions that agree with common sense.

Let us illustrate the notion of regret minimization on the example of Fig. 1. In this example, Eve owns the squares and Adam owns the circles (we do not use the letters labelling edges for the moment). The game is played for infinitely many rounds and the value of a play for Eve is the long-run average of the values of edges traversed during the play (the so-called *mean-payoff*). In this game, Eve is only able to secure a mean-payoff of $\frac{1}{2}$ when Adam is fully antagonistic. Indeed, if Eve (from v_1) plays to v_2 then Adam can force a mean-payoff value of 0, and if she plays to v_3 then the mean-payoff value is at least $\frac{1}{2}$. Note also that if Adam is not fully antagonistic, then the mean-payoff could be as high as 2. Now, assume that Eve does not try to force the highest value in the worst-case but tries to minimize her regret. If she plays $v_1 \mapsto v_2$ then the regret is equal to 1. This is because Adam can play the following strategy: if Eve plays to v_2 (from v_1) then he plays $v_2 \mapsto v_1$ (giving a mean-payoff of 0), and if Eve plays to v_3 then he plays to v_5 (giving a mean-payoff of 1). If she plays $v_1 \mapsto v_3$ then her regret is $\frac{3}{2}$ since Adam can play the symmetric strategy. It should thus be clear that the strategy of Eve which always chooses $v_1 \mapsto v_2$ is indeed minimizing her regret.

In this paper, we will study three variants of *regret minimization*, each corresponding to a different set of strategies we allow Adam to choose from. The first variant is when Adam can play any possible strategy (as in the example above), the second variant is when Adam is restricted to playing *memoryless strategies*, and the third variant is when Adam is restricted to playing *word strategies*. To illustrate the last two variants, let us consider again the example of Fig. 1. Assume now that Adam is playing memoryless strategies only. Then in this case, we

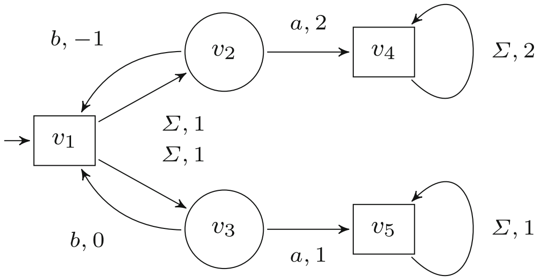

Fig. 1 Example weighted arena G_0

claim that there is a strategy of Eve that ensures regret 0. The strategy is as follows: first play to v_2, if Adam chooses to go back to v_1, then Eve should henceforth play $v_1 \mapsto v_3$. We claim that this strategy has regret 0. Indeed, when v_2 is visited, either Adam chooses $v_2 \mapsto v_4$, and then Eve secures a mean-payoff of 2 (which is the maximal possible value), or Adam chooses $v_2 \mapsto v_1$ and then we know that $v_1 \mapsto v_2$ is not a good option for Eve as cycling between v_1 and v_2 yields a payoff of only 0. In this case, the mean-payoff is either 1, if Adam plays $v_3 \mapsto v_5$, or $\frac{1}{2}$, if he plays $v_3 \mapsto v_1$. In all the cases, the regret is 0. Let us now turn to the restriction to word strategies for Adam. When considering this restriction, we use the letters that label the edges of the graph. A word strategy for Adam is a function $w : \mathbb{N} \to \{a, b\}$. In this setting Adam plays a sequence of letters and this sequence is independent of the current state of the game. It is more convenient to view the latter as a game played on a weighted automata—assumed to be total and with at least one transition for every action from every state—in which Adam plays letters and Eve responds by resolving non-determinism. When Adam plays word strategies, the strategy that minimizes regret for Eve is to always play $v_1 \mapsto v_2$. Indeed, for any word in which the letter a appears, the mean-payoff is equal to 2, and the regret is 0, and for any word in which the letter a does not appear, the mean-payoff is 0 while it would have been equal to $\frac{1}{2}$ when playing $v_1 \mapsto v_3$. So the regret of this strategy is $\frac{1}{2}$ and it is the minimal regret that Eve can secure. Note that the three different strategies give three different values in our example. This is in contrast with the worst-case analysis of the same problem (memoryless strategies suffice for both players).

We claim that at least the two last variants are useful for modelling purposes. For example, the memoryless restriction is useful when designing a system that needs to perform well in an environment which is only partially known. In practical situations, a controller may discover the environment with which it is interacting at run time. Such a situation can be modelled by an arena in which choices in nodes of the environment model an entire family of environments and each memoryless strategy models a specific environment of the family. In such cases, if we want to design a controller that performs reasonably well against all the possible environments, we can consider a controller that minimizes regret: the strategy of the controller will be as close as possible to an optimal strategy if we had known the environment beforehand. This is, for example, the modelling choice done in the famous Canadian traveller's problem [23]: a driver is attempting to reach a specific location while ensuring the traversed distance is *not too far* from the shortest feasible path. The partial knowledge is due to some roads being closed because of snow. The Canadian traveller, when planning his itinerary, is in fact searching for a strategy to minimize his regret for the shortest path measure against a memoryless adversary who determines the roads that are closed. Similar situations naturally arise when synthesizing controllers for *robot motion planning* [27]. We now illustrate the usefulness of the variant in which Adam is restricted to play word strategies. Assume that we need to design a system embedded into an environment that produces disturbances: if the sequence of disturbances produced by the environment is independent of the behavior of the system, then it is natural to model this sequence not as a function of the state of the system but as a temporal sequence of events, i.e. a *word* on the alphabet of the disturbances. Clearly, if the sequences are not the result of an antagonistic process, then minimizing the regret against all disturbance sequences is an adequate solution concept to obtain a reasonable system and may be preferable to a system obtained from a strategy that is optimal under the antagonistic hypothesis.

Table 1 Complexity of deciding the regret threshold problem

Payoff type	Any strategy	Memoryless strategies	Word strategies
Sup, Inf, LimSup	PTIME-c (Theorem 1)	PSPACE (Lemma 4), coNP-h (Lemma 11)	EXPTIME-c (Theorem 3)
LimInf	PTIME-c (Theorem 1)	PSPACE-c (Theorem 2)	EXPTIME-c (Theorem 3)
$\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$	MP equiv. (Theorem 1)	PSPACE-c (Theorem 2)	Undecidable (Lemma 14)

1.1 Contributions

In this paper, we provide algorithms to solve the *regret threshold problem* (strict and non-strict) in the three variants explained above, i.e. given a game and a threshold, does there exist a strategy for Eve with a regret that is (strictly) less than the threshold against all (resp. all memoryless, resp. all word) strategies for Adam. It is worth mentioning that, in the first two cases we consider, we actually provide algorithms to solve the following search problem: find the controller which ensures the minimal possible regret. Indeed, our algorithms are reductions to well-known games and are such that a winning strategy for Eve in the resulting game corresponds to a regret-minimizing strategy in the original one. Conversely, in games played against word strategies for Adam, we only explicitly solve the regret threshold problem. However, since the set of possible regret values of the considered games is finite and easy to describe, it will be obvious that one can implement a binary search to find the regret value and a corresponding optimal regret-minimizing strategy for Eve.

We study this problem for six common quantitative measures: Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$. For all measures, but MP, the strict and non-strict threshold problems are equivalent. We state our results for both cases for consistency. In almost all the cases, we provide matching lower bounds showing the worst-case optimality of our algorithms. Our results are summarized in the table of Fig. 1. For the variant in which Adam plays word strategies only, we show that we can recover decidability of mean-payoff objectives when the memory of Eve is fixed in advance: in this case, the problem is NP-complete (Theorems 4 and 5). Table 1 summarizes our results.

1.2 Related works

The notion of regret minimization is a central one in game theory, see e.g. [28] and references therein. Also, *iterated* regret minimization has been recently proposed by Halpern et al. as a concept for *non-zero* sum games [18]. There, it is applied to matrix games and not to game graphs. In a previous contribution, we have applied the iterated regret minimization concept to non-zero sum games played on weighted graphs for the shortest path problem [16]. Restrictions on how Adam is allowed to play were not considered there. As we do not consider an explicit objective for Adam, we do not consider iteration of the regret minimization here.

The disturbance-handling embedded system example was first given in [11]. In that work, the authors introduce *remorsefree strategies*, which correspond to strategies which minimize regret in games with ω-regular objectives. They do not establish lower bounds on the complexity of realizability or synthesis of remorsefree strategies and they focus on word strategies of Adam only.

In [19], Henzinger and Piterman introduce the notion of *good for games automata*. A non-deterministic automaton is good for solving games if it fairly simulates the equivalent deterministic automaton. We show that our notion of regret minimization for word strategies extends this notion to the quantitative setting (Proposition 3). Our definitions give rise to a natural notion of approximate determinization for weighted automata on infinite words.

In [1], Aminof et al. introduce the notion of *approximate determinization by pruning* for weighted sum automata over finite words. For $\alpha \in (0, 1]$, a weighted sum automaton is *α-determinizable by pruning* if there exists a finite state strategy to resolve non-determinism and that constructs a run whose value is at least α times the value of the maximal run of the given word. So, they consider a notion of approximation which is a *ratio*. We will show that our concept of regret, when Adam plays word strategies only, defines instead a notion of approximation with respect to the *difference* metric for weighted automata (Proposition 2). There are other differences with their work. First, we consider infinite words while they consider finite words. Second, we study a general notion of regret minimization problem in which Eve can use any strategy while they restrict their study to fixed memory strategies only and leave the problem open when the memory is not fixed a priori.

Finally, the main difference between these related works and this paper is that we study the Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$ measures while they consider the total sum measure or qualitative objectives.

2 Preliminaries

A *weighted arena* is a tuple $G = (V, V_\exists, E, w, v_I)$ where (V, E, w) is a finite edge-weighted graph[1] with integer weights, $V_\exists \subseteq V$, and $v_I \in V$ is the initial vertex. In the sequel we depict vertices owned by Eve (i.e. $V_\exists$) with squares and vertices owned by Adam (i.e. $V \setminus V_\exists$) with circles. We denote the maximum absolute value of a weight in a weighted arena by W.

A *play* in a weighted arena is an infinite sequence of vertices $\pi = v_0 v_1 \ldots$ where $v_0 = v_I$ and $(v_i, v_{i+1}) \in E$ for all i. We extend the weight function to partial plays by setting $w(\langle v_i \rangle_{i=k}^{l}) = \sum_{i=k}^{l-1} w(v_i, v_{i+1})$.

A *strategy for* Eve (Adam) is a function σ that maps partial plays ending with a vertex v in $V_\exists$ $(V \setminus V_\exists)$ 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. [20] 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 $\pi = v_0 v_1 \ldots$ is *consistent with a strategy* σ for Eve (Adam) if whenever $v_i \in V_\exists$ $(v_i \in V \setminus V_\exists)$, $\sigma(\langle v_j \rangle_{j \le i}) = v_{i+1}$. We denote by $\mathfrak{S}_\exists(G)$ $(\mathfrak{S}_\forall(G))$ the set of all strategies for Eve (Adam) and by $\Sigma_\exists^m(G)$ $(\Sigma_\forall^m(G))$ the set of all strategies for Eve (Adam) in G that require memory of size at most m, in particular $\Sigma_\exists^1(G)$ $(\Sigma_\forall^1(G))$ is the set of all memoryless strategies of Eve (Adam) in G. We omit G if the context is clear.

2.1 Payoff functions

A play in a weighted arena defines an infinite sequence of weights. We define below several classical *payoff functions* that map such sequences to real numbers.[2] Formally, for a play $\pi = v_0 v_1 \ldots$ we define:

– the Inf (Sup) payoff, is the minimum (maximum) weight seen along a play: $\mathsf{Inf}(\pi) = \inf\{w(v_i, v_{i+1}) \mid i \ge 0\}$ and $\mathsf{Sup}(\pi) = \sup\{w(v_i, v_{i+1}) \mid i \ge 0\}$;

[1] W.l.o.g. G is assumed to be total: for each $v \in V$, there exists $v' \in V$ such that $(v, v') \in E$.

[2] The values of all functions are not infinite, and therefore in $\mathbb{R}$ since we deal with finite graphs only.

- the LimInf (LimSup) payoff, is the minimum (maximum) weight seen infinitely often: $\mathsf{LimInf}(\pi) = \liminf_{i\to\infty} w(v_i, v_{i+1})$ and, respectively, we have that $\mathsf{LimSup}(\pi) = \limsup_{i\to\infty} w(v_i, v_{i+1})$;
- the *mean-payoff* value of a play, i.e. the limiting average weight, defined using lim inf or lim sup since the running averages might not converge: $\underline{\mathsf{MP}}(\pi) = \liminf_{k\to\infty} \frac{1}{k} w(\langle v_i\rangle_{i<k})$ and $\overline{\mathsf{MP}}(\pi) = \limsup_{k\to\infty} \frac{1}{k} w(\langle v_i\rangle_{i<k})$. In words, $\underline{\mathsf{MP}}$ corresponds to the *limit inferior* of the average weight of increasingly longer prefixes of the play while $\overline{\mathsf{MP}}$ is defined as the *limit superior* of that same sequence.

A payoff function **Val** is *prefix-independent* if for all plays $\pi = v_0 v_1 \ldots$, for all $i \geq 0$, $\mathbf{Val}(\pi) = \mathbf{Val}(\langle v_j\rangle_{j\geq i})$. It is well-known that LimInf, LimSup, $\underline{\mathsf{MP}}$, and $\overline{\mathsf{MP}}$ are prefix-independent. Often, the arguments that we develop work uniformly for these four measures because of their prefix-independent property. Inf and Sup are not prefix-independent but often in the sequel we apply a simple transformation to the game and encode Inf into a LimInf objective, and Sup into a LimSup objective. The transformation consists of encoding in the vertices of the arena the minimal (maximal) weight that has been witnessed by a play, and label the edges of the new graph with this same recorded weight. When this simple transformation does not suffice, we mention it explicitly.

2.2 Regret

Consider a fixed weighted arena G, and payoff function **Val**. 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 and denote its value 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.

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:

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

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.

Remark 1 (Ratio vs. difference) Let G be a weighted arena and $\Sigma_\exists \subseteq \mathfrak{S}_\exists$ and $\Sigma_\forall \subseteq \mathfrak{S}_\forall$. Consider the regret of G defined using the ratio measure, instead of difference. For Inf, Sup, LimInf, and LimSup, deciding if the regret of G is (strictly) less than a given threshold r reduces (in polynomial time) to deciding the same problem in $G_{\log}$ – which is obtained by replacing every weight x in G with $\log_2 x$ – for threshold $\log_2 r$ with the difference measure.

We will make use of two other values associated with the vertices of an arena: the *antagonistic* and *cooperative* values, defined for plays from a vertex $v \in V$ as

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

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

Remark 2 It is well-known that **cVal** and **aVal** can be computed in polynomial time, w.r.t. the underlying graph of the given arena, for all payoff functions but MP [7,9]. For MP, **cVal** is known to be computable in polynomial time, for **aVal** it can be done in $\mathsf{UP} \cap \mathsf{coUP}$ [21] and in *pseudo-polynomial time* [6,29].

3 Variant I: Adam plays any strategy

For this variant, we establish that for all the payoff functions that we consider, the problem of computing the antagonistic value and the problem of computing the regret value are *inter-reducible* in polynomial time. As a direct consequence, we obtain the following theorem:

Theorem 1 *Deciding if the regret value is less than a given threshold (strictly or non-strictly) is* PTIME*-complete (under log-space reductions) for* Inf*,* Sup*,* LimInf*, and* LimSup*, and equivalent to mean-payoff games (under polynomial-time reductions) for* $\underline{\mathsf{MP}}$ *and* $\overline{\mathsf{MP}}$.

3.1 Upper bounds

We now describe an algorithm to compute regret for all payoff functions. To do so, we will use the fact that all payoff functions we consider, can be assumed to be prefix-independent. Thus, let us first convince the reader that one can, in polynomial time, modify Inf and Sup games so that they become prefix-independent.

Lemma 1 *For a given weighted arena G, and payoff function* Sup*:* $\mathbf{Reg}(G) = \mathbf{Reg}(G_{\max})$*; for payoff function* Inf*:* $\mathbf{Reg}(G) = \mathbf{Reg}(G_{\min})$.

Consider a weighted arena $G = (V, V_\exists, v_I, E, w)$. We describe how to construct $G_{\min}$ from G so that there is a clear bijection between plays in both games defined with the Inf payoff function. The arena $G_{\min}$ consists of the following components:

- $V' = V \times \{w(e) \mid e \in E\}$;
- $V'_\exists = \{(v, n) \in V' \mid v \in V_\exists\}$;
- $v'_I = (v_I, W)$;
- $E' \ni \big((u, n), (v, m)\big)$ if and only if $(u, v) \in E$ and $m = \min\{n, w(u, v)\}$;
- $w'\big((u, n), (v, m)\big) = m$.

Intuitively, the construction keeps track of the minimal weight witnessed by a play by encoding it into the vertices themselves. It is not hard to see that plays in $G_{\min}$ indeed have a one-to-one correspondence with plays in G. Furthermore, the LimInf and LimSup values of a play in $G_{\min}$ are easily seen to be equivalent to the Inf value of the play in $G_{\min}$ and the corresponding play in G.

A similar idea can be used to construct weighted arena $G_{\max}$ from a Sup game such that the maximal weight is recorded (instead of the minimal).

Lemma 2 *For payoff functions* Inf*,* Sup*,* LimInf*,* LimSup*,* $\underline{\mathsf{MP}}$*, and* $\overline{\mathsf{MP}}$ *computing the regret of a game is at most as hard as computing the antagonistic value of a (polynomial-size) game with the same payoff function.*

Consider a weighted arena $G = (V, V_\exists, v_I, E, w)$. We describe how to construct $G_{\min}$ from G so that there is a clear bijection between plays in both games defined with the Inf payoff function. The arena $G_{\min}$ consists of the following components:

- $V' = V \times \{w(e) \mid e \in E\}$;
- $V'_\exists = \{(v, n) \in V' \mid v \in V_\exists\}$;
- $v'_I = (v_I, W)$;
- $E' \ni \big((u, n), (v, m)\big)$ if and only if $(u, v) \in E$ and $m = \min\{n, w(u, v)\}$;
- $w'\big((u, n), (v, m)\big) = m$.

Intuitively, the construction keeps track of the minimal weight witnessed by a play by encoding it into the vertices themselves. It is not hard to see that plays in $G_{\min}$ indeed have a one-to-one correspondence with plays in G. Furthermore, the LimInf and LimSup values of a play in $G_{\min}$ are easily seen to be equivalent to the Inf value of the play in $G_{\min}$ and the corresponding play in G.

A similar idea can be used to construct weighted arena $G_{\max}$ from a Sup game such that the maximal weight is recorded (instead of the minimal).

We now describe the construction used to prove Lemma 2.

Let us fix a weighted arena G. We define a new weight function w' as follows. For any edge $e = (u, v)$ let $w'(e) = -\infty$ if $u \in V \setminus V_\exists$, and if $u \in V_\exists$ then $w'(e) = \max\{\mathbf{cVal}^{v'} \mid (u, v') \in E \setminus \{e\}\}$. Intuitively, w' represents the best value obtainable for a strategy of Eve that differs at the given edge. It is not difficult to see that in order to minimize regret, Eve is trying to minimize the difference between the value given by the original weight function w and the value given by w'. Let $\text{Range}(w')$ be the set of values $\{w'(e) \mid e \in E\}$. For $b \in \text{Range}(w')$ we define G^b to be the graph obtained by restricting G—the original weighted arena with weight function w—to edges e with $w'(e) \leq b$.

Next, we will construct a new weighted arena $\hat{G}$ such that the regret of G is a function of the *antagonistic* value of $\hat{G}$. Figure 2 depicts the general form of the arena we construct. We have three vertices $v_0 \in \hat{V} \setminus \hat{V}_\exists$ and $v_1, v_n \in \hat{V}_\exists$ and a "copy" of G as G^b for each $b \in \text{Range}(w') \setminus \{-\infty\}$. We have a self-loop of weight 0 on v_0 which is the initial vertex of $\hat{G}$, a self-loop of weight $-2W - 1$ on $v_\perp$, and weight-0 edges from v_0 to v_1 and from v_1 to the initial vertices of G^b for all b. Recall that G^b might not be total. To fix this we add, for all vertices without a successor, a weight-0 edge to $v_\perp$. The remainder of the weight function $\hat{w}$, is defined for each edge e^b in G^b as $\hat{w}(e^b) = w(e) - b$.

Intuitively, in $\hat{G}$ Adam first decides whether he can ensure a non-zero regret. If this is the case, then he moves to v_1. Next, Eve chooses a maximal value she will allow for strategies which differ from the one she will play (this is the choice of b). The play then moves to the corresponding copy of G, i.e. G^b. She can now play to maximize her mean-payoff value. However, if her choice of b was not correct then the play will end in $v_\perp$.

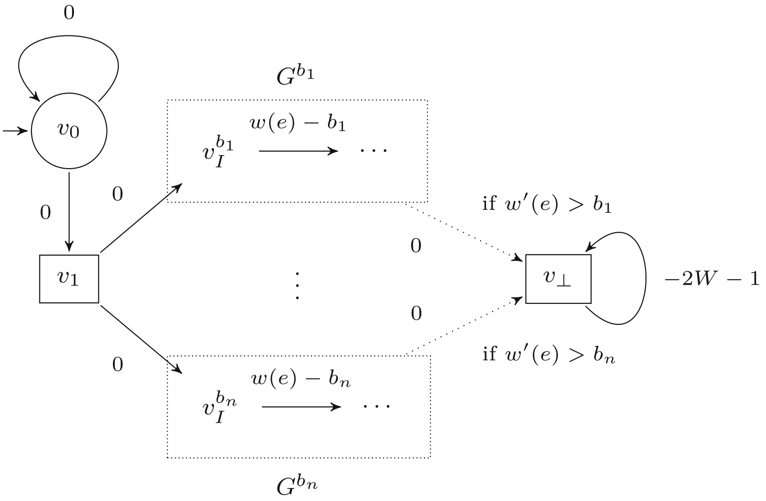

Fig. 2 Weighted arena $\hat{G}$, constructed from G. *Dotted lines* represent several edges added when the condition labelling it is met

We show that, for all prefix-independent payoff functions we consider, the following holds:

Claim For all prefix-independent payoff functions considered in this work $\mathbf{Reg}(G) = -\mathbf{aVal}(\hat{G})$.

This implies Lemma 2 for all prefix-independent payoff functions. Together with Lemma 1, we get the same result for Inf and Sup.

Proof (of the Claim) Let us start by arguing that the following equality holds.

$$\mathbf{Reg}(G) = \inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\}} \{0, \mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau)\}. \tag{1}$$

Indeed, it follows from the definition of regret that if $\sigma' = \sigma$ then the regret of the game is 0. Thus, Adam can always ensure the regret of a game is at least 0. Now, for $b \in \mathrm{Range}(w')$, define $\Sigma_\exists(b) \subseteq \mathfrak{S}_\exists(G)$ as:

$$\Sigma_\exists(b) := \{\sigma \mid \sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\}} \mathbf{Val}(\sigma', \tau) \leq b\}.$$

It is clear from the definitions that $\sigma \in \Sigma_\exists(b)$ if and only if σ is a strategy for Eve in G^b which avoids ever reaching $v_\perp$. Now, if we let

$$b_\sigma = \sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\}} \mathbf{Val}(\sigma', \tau),$$

then $\sigma \in \Sigma_\exists(b)$ if and only $b_\sigma \leq b$. It follows that for all σ:

$$\sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\}} \mathbf{Val}(\sigma', \tau) = \inf\{b \mid \sigma \in \Sigma_\exists(b)\}. \tag{2}$$

We now turn to the mean-payoff game played on $\hat{G}$, and make some observations about the strategies we need to consider. It is well known that memoryless strategies suffice for either player to ensure an antagonistic value of at least (resp. at most) $\mathbf{aVal}(\hat{G})$, for all quantitative games considered in this work, so we can assume that Adam and Eve play positionally. It follows that all plays either remain in v_0, or move to G^b for some b, and Adam can ensure a non-positive payoff. Note that for $b_{\max} = \max(\mathrm{Range}(w') \setminus \{-\infty\})$ we have $G^{b_{\max}} = G$. So the copy of $G^{b_{\max}}$ in $\hat{G}$ has no edge to $v_\perp$, and by playing to this sub-graph Eve can ensure a payoff of at least $-|b_{\max} - W| \geq -2W$. As any play that reaches $v_\perp$ will have a payoff of $-2W - 1$, we can restrict Eve to strategies which avoid $v_\perp$, and hence all plays either remain in v_0 or (eventually) in the copy of G^b for some b. Now G^b contains no restrictions for Adam, so we can assume that he plays the same strategy in all the copies of G^b (where he cannot force the play to $v_\perp$), and these strategies have a one-to-one correspondence with strategies in G. Likewise, as Eve chooses a unique G^b to play in, we have a one-to-one correspondence with strategies of Eve in $\hat{G}$ and strategies in G. More precisely, if $\hat{\sigma} \in \mathfrak{S}_\exists(\hat{G})$ is such that $\hat{\sigma}(v_1) = v_I^b$ and $\hat{\sigma}$ avoids $v_\perp$, then the corresponding strategy $\sigma \in \mathfrak{S}_\exists(G)$ is a valid strategy in G^b, and hence:

$$\hat{\sigma}(v_1) = v_I^b \implies \sigma \in \Sigma(b). \tag{3}$$

Now suppose $\hat{\sigma} \in \mathfrak{S}_\exists(\hat{G})$ is a strategy such that $\hat{\sigma}(v_1) = v_I^b$ and $\hat{\sigma}$ avoids $v_\perp$, and $\hat{\tau} \in \mathfrak{S}_\forall(\hat{G})$ is a strategy such that $\hat{\tau}(v_0) = v_1$. Let $\sigma \in \Sigma(b)$ and $\tau \in \mathfrak{S}_\forall(G)$ be the strategies in G corresponding to $\hat{\sigma}$ and $\hat{\tau}$ respectively. It is easy to show that:

$$-\mathbf{Val}_{\hat{G}}(\hat{\sigma}, \hat{\tau}) = b - \mathbf{Val}_G(\sigma, \tau). \tag{4}$$

Putting together Eqs. (1)–(4) gives:

$$\begin{aligned}
-\mathbf{aVal}(\hat{G}) &= -\sup_{\hat{\sigma}} \inf_{\hat{\tau}} \mathbf{Val}_{\hat{G}}(\hat{\sigma}, \hat{\tau}) \\
&= \inf_{\hat{\sigma}} \sup(\{-\mathbf{Val}_{\hat{G}}(\hat{\sigma}, \hat{\tau}) \mid \hat{\tau}(v_0) = v_1\} \cup \{0\}) \\
&= \inf\{\sup(\{-\mathbf{Val}_{\hat{G}}(\hat{\sigma}, \hat{\tau}) \mid \hat{\tau}(v_0) = v_1\} \cup \{0\}) \mid \hat{\sigma}(v_1) = v_I^b\} \\
&= \inf\{\sup_{\tau \in \mathfrak{S}_\forall}(\{b - \mathbf{Val}_G(\sigma, \tau)\} \cup \{0\}) \mid \sigma \in \Sigma(b)\} \\
&= \inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall}(\{\inf\{b \mid \sigma \in \Sigma(b)\} - \mathbf{Val}_G(\sigma, \tau)\} \cup \{0\}) \\
&= \inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists}\{0, \mathbf{Val}_G(\sigma', \tau) - \mathbf{Val}_G(\sigma, \tau)\} \\
&= \mathbf{Reg}(G) \text{ as required.}
\end{aligned}$$

□

3.2 Lower bounds

For all the payoff functions, from G we can construct in logarithmic space G' such that the antagonistic value of G is a function of the regret value of G', and so we have:

Lemma 3 *For payoff functions* Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, *and* $\overline{\mathsf{MP}}$ *computing the regret of a game is at least as hard as computing the antagonistic value of a (polynomial-size) game with the same payoff function.*

Proof Suppose G is a weighted arena with initial vertex v_I. Consider the weighted arena G' obtained by adding to G the gadget of Fig. 3. The initial vertex of G' is set to be v'_I. In G' from v'_I Eve can either progress to the original game or to the new gadget, both with weight L. We claim that the right choice of values for the parameters L, M_1, M_2, N_1, N_2 makes it so that the antagonistic value of G is a function of the regret of the game G'.

Let us first give the values of L, M_1, M_2, N_1, and N_2 for each of the payoff functions considered. For all our payoff functions we have $M_1 = M_2 = L$; $N_1 = W + 1$; and $N_2 = -3W - 2$. For Inf we have $L = W$, for Sup we have $L = -W$ and for the remaining payoff functions we have $L = 0$. □

The following result shows that computing the regret of G is at least as hard as computing the (antagonistic) value of G'.

Claim For all payoff functions:

$$\mathbf{aVal}(G) = W + 1 - \mathbf{Reg}(G').$$

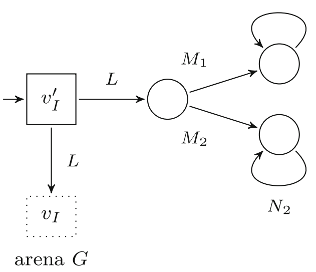

Fig. 3 Gadget to reduce a game to its regret game

We first observe that for all payoff functions we consider we have that $\mathbf{aVal}(G)$ and $\mathbf{cVal}(G)$ both lie in $[-W, W]$.

At v'_I Eve has a choice: she can choose to remain in the gadget or she can move to the original game G. If she chooses to remain in the gadget, her payoff will be $-3W - 2$, meanwhile Adam could choose a strategy that would have achieved a payoff of $\mathbf{cVal}(G)$ if she had chosen to play to G. Hence her regret in this case is $\mathbf{cVal}(G) + 3W + 2 \geq 2W + 2$. Otherwise, if she chooses to play to G she can achieve a payoff of at most $\mathbf{aVal}(G)$. As $\mathbf{cVal}(G) \leq W$ is the maximum possible payoff achievable in G, the strategy which now maximizes Eve's regret is the one which remains in the gadget – giving a payoff of $W + 1$. Her regret in this case is $W + 1 - \mathbf{aVal}(G) \leq 2W + 1$. Therefore, to minimize her regret she will play this strategy, and $\mathbf{Reg}(G') = W + 1 - \mathbf{aVal}(G)$. □

3.3 Memory requirements for Eve and Adam

It follows from the reductions underlying the proof of Lemma 2 that Eve only requires positional strategies to minimize regret when there is no restriction on Adam's strategies. On the other hand, for any given strategy σ for Eve, the strategy τ for Adam which witnesses the maximal regret against it consists of a combination of three positional strategies: first he moves to the optimal vertex for deviating (it is from this vertex that the alternative strategy σ' of Eve will achieve a better payoff against τ), then he plays his optimal (positional) strategy in the antagonistic game (i.e. against σ). His strategy for the alternative scenario, i.e. against σ', is his optimal strategy in the co-operative game which is also positional. This combined strategy is clearly realizable as a strategy with three memory states, giving us:

Corollary 1 *For payoff functions* LimInf, LimSup, $\underline{\mathsf{MP}}$ *and* $\overline{\mathsf{MP}}$*:* $\mathbf{Reg}(G) = \mathbf{Reg}_{\Sigma^1_\exists, \Sigma^3_\forall}(G)$.

The algorithm we give relies on the prefix-independence of the payoff function. As the transformation from Inf and Sup to equivalent prefix-independent ones is polynomial it follows that polynomial memory (w.r.t. the size of the underlying graph of the arena) suffices for both players.

4 Variant II: Adam plays memoryless strategies

For this variant, we provide a polynomial space algorithm to solve the problem for all the payoff functions, we then provide lower bounds.

Theorem 2 *Deciding if the regret value is less than a given threshold (strictly or non-strictly) playing against memoryless strategies of Adam is* PSPACE*-complete for* LimInf, $\underline{\mathsf{MP}}$ *and* $\overline{\mathsf{MP}}$*; in* PSPACE *and* coNP*-hard for* Inf, Sup *and* LimSup.

4.1 Upper bounds

Let us now show how to compute regret against positional adversaries.

Lemma 4 *For payoff functions* Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$ *and* $\overline{\mathsf{MP}}$*, the regret of a game played against a positional adversary can be computed in polynomial space.*

Given a weighted arena G, we construct a new weighted arena $\hat{G}$ such that we have that $-\mathbf{aVal}(\hat{G})$ is equivalent to the regret of G.

The vertices of $\hat{G}$ encode the choices made by Adam. For a subset of edges $D \subseteq E$, let $G \upharpoonright D$ denote the weighted arena $(V, V_\exists, D, w, v_I)$. The new weighted arena $\hat{G}$ is the

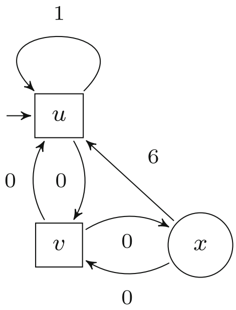

Fig. 4 Example weighted arena G_1

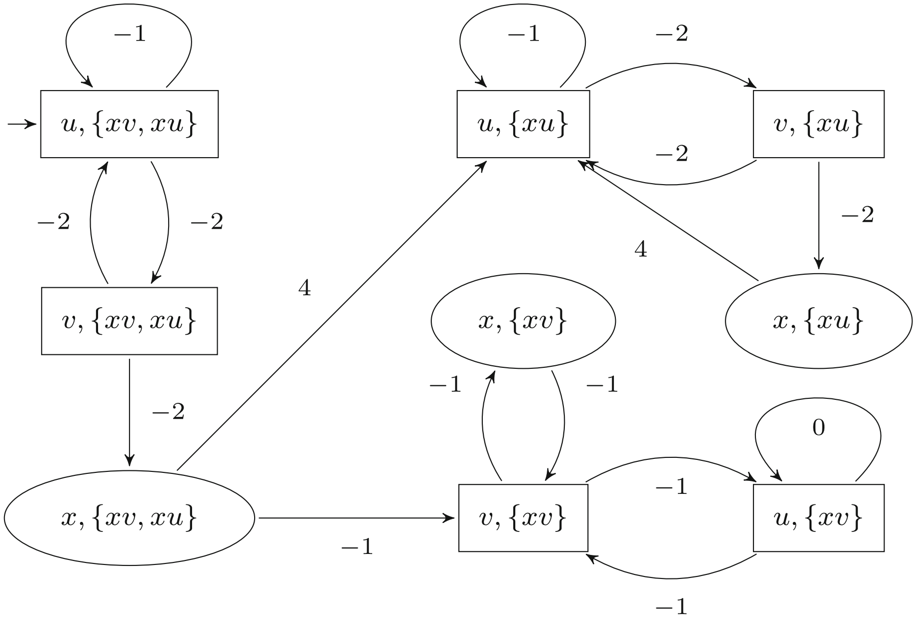

Fig. 5 Weighted arena $\hat{G}_1$, constructed from G_1 w.r.t the $\underline{\mathsf{MP}}$ payoff function. In the edge set component only edges leaving Adam nodes are depicted

tuple $(\hat{V}, \hat{V}_\exists, \hat{E}, \hat{w}, \hat{v}_I)$ where (i) $\hat{V} = V \times \mathcal{P}(E)$; (ii) $\hat{V}_\exists = \{(v, D) \in \hat{V} \mid v \in V_\exists\}$; (iii) $\hat{v}_I = (v_I, E)$; (iv) $\hat{E}$ contains the edge $\big((u, C), (v, D)\big)$ if and only if $(u, v) \in C$ and, either $u \in V_\exists$ and $D = C$, or $u \in V \backslash V_\exists$ and $D = C \backslash \{(u, x) \in E \mid x \neq v\}$; (v) $\hat{w}\big((u, C), (v, D)\big) = w(u, v) - \mathbf{cVal}(G \restriction D)$. The application of this transformation for the graph of Fig. 4 w.r.t. to the $\underline{\mathsf{MP}}$ payoff function is given in Fig. 5.

Consider a play $\hat{\pi} = (v_0, C_0)(v_1, C_1)\ldots$ in $\hat{G}$. We denote by $[\hat{\pi}]_{\mathbf{k}}$, for $k \in \{1, 2\}$, the sequence $\langle c_{k,i}\rangle_{i\geq 0}$, where $c_{k,i}$ is the k-th component of the i-th pair from $\hat{\pi}$. Observe that $[\hat{\pi}]_{\mathbf{1}}$ is a valid play in G. Also observe that $E \supseteq C_j \supseteq C_{j+1}$ for all j. Hence $[\hat{\pi}]_{\mathbf{2}}$ is an infinite descending chain of finite subsets, and therefore $\lim\, [\hat{\pi}]_{\mathbf{2}}$ is well-defined. Finally, we define $\mathbf{c}(\hat{\pi}) := \mathbf{cVal}(G \restriction \lim\, [\hat{\pi}]_{\mathbf{2}})$. The following result relates the value of a play in $\hat{G}$ to the value of the corresponding play in G.

Lemma 5 *For payoff functions* LimInf, LimSup, $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$ *and for any play* $\hat{\pi}$ *in* $\hat{G}$ *we have that* $\mathbf{Val}(\hat{\pi}) = \mathbf{Val}([\hat{\pi}]_{\mathbf{1}}) - \mathbf{c}(\hat{\pi})$.

Proof We first establish the following intermediate result. It follows from the existence of $\lim\,[\hat{\pi}]_{\mathbf{2}}$ and the definition of $\mathbf{c}(\cdot)$ that:

$$\limsup_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{cVal}(G \restriction C_i) = \liminf_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{cVal}(G \restriction C_i) = \mathbf{c}(\hat{\pi}). \tag{5}$$

We now show that the result holds for $\underline{\mathsf{MP}}$.

$$\begin{aligned}
\mathbf{Val}(\hat{\pi}) &= \liminf_{n\to\infty}\left(\frac{1}{n}\sum_{i=0}^{n-1}\big(w(v_i, v_{i+1}) - \mathbf{cVal}(G \restriction C_j)\big)\right) && \text{defs. of } \mathbf{Val}(\cdot),\ \hat{w}\\
&= \mathbf{Val}([\hat{\pi}]_{\mathbf{1}}) - \limsup_{n\to\infty}\frac{1}{n}\sum_{j=0}^{n-1}\mathbf{cVal}(G \restriction C_j) && \text{def. of } \mathbf{Val}(\cdot)\\
&= \mathbf{Val}([\hat{\pi}]_{\mathbf{1}}) - \mathbf{c}(\hat{\pi}) && \text{from Eq. (5)}
\end{aligned}$$

The proofs for the other payoff functions are almost identical (for LimInf and LimSup replace the use of Eqs. (5) by (6)).

$$\limsup_{i\to\infty} \mathbf{cVal}(G \restriction C_i) = \liminf_{i\to\infty} \mathbf{cVal}(G \restriction C_i) = \mathbf{c}(\hat{\pi}). \tag{6}$$

□

We now describe how to translate winning strategies for either player from $\hat{G}$ back to G, i.e. given an optimal maximizing (minimizing) strategy for Eve (Adam) in $\hat{G}$ we construct the corresponding optimal regret minimizing strategy (memoryless regret maximizing counter-strategy) for Eve (Adam) in G. For clarity, we follow this same naming convention throughout this section: again, we say a strategy is an optimal maximizing (minimizing) strategy when we speak about antagonistic and cooperative games, we say a strategy is an optimal regret maximizing (regret minimizing) when we speak about regret games. When this does not suffice, we explicitly state which kind of game we are speaking about.

Let $\hat{\epsilon} \in \mathfrak{S}_{\exists}(\hat{G})$ be an optimal maximizing strategy of Eve in $\hat{G}$ and $\hat{\alpha} \in \mathfrak{S}_{\forall}(\hat{G})$ be an optimal minimizing strategy of Adam. Indeed, in [14] it was shown that mean-payoff games are positionally determined. We will now define a strategy for Eve in G which for every play prefix s constructs a valid play prefix $\hat{s}$ in $\hat{G}$ and plays as $\hat{\epsilon}$ would in $\hat{G}$ for $\hat{s}$. More formally, for a play prefix s from G, denote by $[s]_{\mathbf{1}}^{-1}$ the corresponding sequence of vertex and edge-set pairs in $\hat{G}$ (indeed, it is the inverse function of $[\cdot]_{\mathbf{1}}$, which is easily seen to be bijective). Define $\sigma \in \mathfrak{S}_{\exists}(G)$ as follows: $\sigma(s) = [\hat{\epsilon}([s]_{\mathbf{1}}^{-1})]_{\mathbf{1}}$ for all play prefixes $s \in V^* \cdot V_{\exists}$ in G consistent with a positional strategy of Adam.

For a fixed strategy of Eve we can translate the optimal minimizing strategy of Adam in $\hat{G}$ into an optimal memoryless regret maximizing counter-strategy of his in G. Formally, for an arbitrary strategy σ for Eve in G, define $\hat{\sigma} \in \mathfrak{S}_{\exists}(\hat{G})$ as follows: $\hat{\sigma}(\hat{s}) = \sigma([\hat{s}]_{\mathbf{1}})$ for all $\hat{s} \in \hat{V}^* \cdot \hat{V}_{\exists}$. Let τ_{σ} be an optimal (positional) maximizing strategy for Adam in $G \restriction \lim\,[\pi_{\hat{\sigma}\hat{\alpha}}]_{\mathbf{2}}$.

It is not hard to see the described strategy of Eve ensures a regret value of at most $-\mathbf{aVal}(\hat{G})$. Slightly less obvious is the fact that for any strategy of Eve, the counter-strategy τ_{σ} of Adam is such that $\sup_{\sigma' \in \mathfrak{S}_{\exists}} \mathbf{Val}_{G}(\sigma', \tau_{\sigma}) - \mathbf{Val}_{G}(\sigma, \tau_{\sigma}) \geq -\mathbf{aVal}(\hat{G})$.

Lemma 6 *For payoff functions* LimInf, LimSup, $\underline{\mathsf{MP}}$*, and* $\overline{\mathsf{MP}}$*:*

$$\mathbf{Reg}_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G) = -\mathbf{aVal}(\hat{G}).$$

Proof The proof is decomposed into two parts. First, we describe a strategy $\sigma \in \mathfrak{S}_\exists(G)$ which ensures a regret value of at most $-\mathbf{aVal}(\hat{G})$. Second, we show that for any $\sigma \in \mathfrak{S}_\exists(G)$ there is a $\tau \in \Sigma^1_\forall(G)$ such that

$$\sup_{\sigma' \in \mathfrak{S}_\exists} \mathbf{Val}_G(\sigma', \tau) - \mathbf{Val}_G(\sigma, \tau) \geq -\mathbf{aVal}(\hat{G}).$$

The result follows. □

We have already mentioned earlier that for a play $\hat{\pi}$ in $\hat{G}$ we have that $[\hat{\pi}]_\mathbf{1}$ is a play in G. Let $\mathrm{PPref}(G)$ denote the set of all play prefixes consistent with a positional strategy for Adam in G. It is not difficult to see that $[\cdot]_\mathbf{1}$ is indeed a bijection between plays of $\hat{G}$ and plays of G consistent with positional strategies for Adam.

It follows from the determinacy of antagonistic games defined by the payoff functions considered in this work that there are optimal strategies for Eve and Adam that ensure a payoff of, respectively, at least and at most a value $\mathbf{aVal}(\hat{G})$ against any strategy of the opposing player. Let $\hat{\epsilon} \in \mathfrak{S}_\exists(\hat{G})$ be an optimal maximizing strategy of Eve in $\hat{G}$ and $\hat{\alpha} \in \mathfrak{S}_\forall(\hat{G})$ be an optimal minimizing strategy of Adam.

(*First part*). Define a strategy σ from $\mathfrak{S}_\exists(G)$ as follows: $\sigma(s) = [\hat{\epsilon}([s]_\mathbf{1}^{-1})]_\mathbf{1}$ for all $s \in \mathrm{PPref}(G) \cdot V_\exists$. We claim that

$$\mathbf{reg}^\sigma_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G) \leq -\mathbf{aVal}(\hat{G}).$$

Towards a contradiction, assume there are $\tau \in \Sigma^1_\forall(G)$ and $\sigma' \in \mathfrak{S}_\exists(G)$ such that

$$\mathbf{Val}_G(\sigma', \tau) - \mathbf{Val}_G(\sigma, \tau) > -\mathbf{aVal}(\hat{G}).$$

Define a strategy $\hat{\tau} \in \mathfrak{S}_\forall(\hat{G})$ as follows: $\hat{\tau}(\hat{s}) = \tau([\hat{s}]_\mathbf{1})$ for all $\hat{s} \in \hat{V}^* \cdot \widehat{V \setminus V_\exists}$. From the definition of $\hat{\epsilon}$ and our assumption we get that

$$\mathbf{Val}_G(\sigma', \tau) - \mathbf{Val}_G(\sigma, \tau) > -\mathbf{aVal}(\hat{G}) \geq -\mathbf{Val}_{\hat{G}}(\hat{\epsilon}, \hat{\tau}). \tag{7}$$

It is straightforward to verify that $[\pi_{\sigma\tau}]_\mathbf{1}^{-1} = \pi_{\hat{\epsilon}\hat{\tau}}$. Therefore, from Lemma 5, we have:

$$\mathbf{Val}(\pi_{\sigma'\tau}) > \mathbf{Val}(\pi_{\sigma\tau}) - \mathbf{Val}(\pi_{\hat{\epsilon}\hat{\tau}}) = \mathbf{cVal}(G \restriction \lim\,[\pi_{\hat{\epsilon}\hat{\tau}}]_\mathbf{2}). \tag{8}$$

At this point we note that, since τ is a positional strategy, it holds that $\mathbf{Val}_G(\sigma', \tau)$ is at most the highest payoff value attainable in G restricted to the edges allowed by τ. Formally, if $E_\tau = \{(u, v) \in E \mid u \in V \setminus V_\exists \implies v = \tau(u)\}$ then $\mathbf{Val}(\pi_{\sigma'\tau}) \leq \mathbf{cVal}(G \restriction E_\tau)$. Also, by construction of $\hat{\tau}$ we get that $E_\tau \subseteq \lim\,[\pi_{\hat{\epsilon}\hat{\tau}}]_\mathbf{2}$. It should be clear that this implies $\mathbf{cVal}(G \restriction \lim\,[\pi_{\hat{\epsilon}\hat{\tau}}]_\mathbf{2}) \geq \mathbf{cVal}(G \restriction E_\tau)$. This contradicts Eq. (8).

(*Second part*). For the second part of the proof we require the following result which relates positional strategies for Adam in G that agree on certain vertices to strategies in sub-graphs defined by plays in $\hat{G}$.

Claim Let $\sigma \in \mathfrak{S}_\exists(G)$ and $\tau, \tau' \in \Sigma^1_\forall(G)$. Then $\pi_{\sigma\tau} = \pi_{\sigma\tau'}$ if and only if $\tau' \in \Sigma^1_\forall(G \restriction \lim\,[[\pi_{\tau\sigma}]_\mathbf{1}^{-1}]_\mathbf{2})$

Proof (only if) Note that by construction of $\hat{G}$ we have that once Adam chooses an edge $\big((u, C), (v, D)\big)$ from a vertex $(v, C) \in \hat{V} \setminus \hat{V}_\exists$ then on any subsequent visit to a vertex $(u, C') \in \hat{V} \setminus \hat{V}_\exists$ he has no other option but to go to (v, C'). That is, his choice is restricted to be consistent with the history of the play. For a play $\hat{\pi}$ in $\hat{G}$, it is clear that the sequence $[\hat{\pi}]_\mathbf{2}$ is the decreasing sequence of sets of edges consistent (for Adam) with the history of the play in the same manner. In particular, for any $\tau' \in \Sigma^1_\forall(G)$ and any play π in G consistent with τ' we have that τ' is a valid strategy for Adam in $G \restriction E'$ where $E' = \lim\,[[\pi]_\mathbf{1}^{-1}]_\mathbf{2}$. As $\pi_{\sigma\tau} = \pi_{\sigma\tau'}$ is a play consistent with τ', the result follows.

(if) Suppose $\pi_{\sigma\tau} \neq \pi_{\sigma\tau'}$, and let v be the last vertex in their common prefix. As σ is common to both plays, we have $v \in V \setminus V_\exists$, and $\tau(v) \neq \tau'(v)$. In particular, $(v, \tau'(v)) \notin \lim\,[[\pi_{\tau\sigma}]_\mathbf{1}^{-1}]_\mathbf{2}$ so $\tau' \notin \Sigma^1_\forall(G \restriction \lim\,[[\pi_{\tau\sigma}]_\mathbf{1}^{-1}]_\mathbf{2})$. □

For an arbitrary strategy σ for Eve in G, define $\hat{\sigma} \in \mathfrak{S}_\exists(\hat{G})$ as follows: $\hat{\sigma}(\hat{s} \cdot (v, D)) = (\sigma([\hat{s} \cdot (v, D)]_\mathbf{1}^{-1}), D)$ for all $\hat{s} \cdot (v, D) \in \hat{V}^* \cdot \hat{V}_\exists$. Let τ_σ be an optimal (positional) maximizing strategy for Adam in $G \restriction \lim\,[\pi_{\hat{\sigma}\hat{\alpha}}]_\mathbf{2}$. We claim that for all $\sigma \in \mathfrak{S}_\exists(G)$ we have that

$$\sup_{\sigma' \in \mathfrak{S}_\exists} \mathbf{Val}_G(\sigma', \tau_\sigma) - \mathbf{Val}_G(\sigma, \tau_\sigma) \geq -\mathbf{aVal}(\hat{G}).$$

Towards a contradiction, assume that for some $\sigma \in \mathfrak{S}_\exists(G)$ it is the case that for all $\sigma' \in \mathfrak{S}_\exists(G)$ the left hand side of the above inequality is strictly smaller than the right hand side. By definition of $\hat{\alpha}$ we then get the following inequality.

$$\sup_{\sigma' \in \mathfrak{S}_\exists} \mathbf{Val}_G(\sigma', \tau_\sigma) - \mathbf{Val}_G(\sigma, \tau_\sigma) < -\mathbf{aVal}(\hat{G}) \leq -\mathbf{Val}_{\hat{G}}(\hat{\sigma}, \hat{\alpha}) \tag{9}$$

Using the above Claim it is easy to show that $[\pi_{\sigma\tau_\sigma}]_\mathbf{1} = \pi_{\hat{\sigma}\hat{\alpha}}$. Hence, by Eq. (9) and Lemma 5 we get that:

$$\sup_{\sigma' \in \mathfrak{S}_\exists} \mathbf{Val}(\pi_{\sigma'\tau_\sigma}) < \mathbf{cVal}(G \restriction \lim\,[\pi_{\hat{\sigma}\hat{\alpha}}]_\mathbf{2}) \tag{10}$$

However, by choice of τ_σ, we know that there is a strategy $\sigma'' \in \mathfrak{S}_\exists(G)$ such that $\mathbf{Val}(\pi_{\sigma''\tau_\sigma}) = \mathbf{cVal}(G \restriction \lim\,[\pi_{\hat{\sigma}\hat{\alpha}}]_\mathbf{2})$. This contradicts Eq. (10) and completes the proof of the Theorem. □

If G was constructed from a Inf or Sup game H, then one could easily transfer the described strategy of Eve, σ into a strategy for her in H which achieves the same regret. In order to have a symmetric result we still lack the ability to transfer a strategy of Adam from $\hat{G}$ to the original game H. Consider a modified construction in which we additionally keep track of the minimal (resp. maximal) weight seen so far by a play, just like described in Sect. 3. Denote the corresponding game by $\tilde{G}$. The vertex set $\tilde{V}$ of $\tilde{G}$ is thus a set of triples of the form (v, C, x) where x is the minimal (resp. maximal) weight the play has witnessed. We observe that in the proof of the above result the intuition behind why we can transfer a strategy of Adam from $\hat{G}$ back to G as a memoryless strategy, **although the vertices in $\hat{G}$ already encode additional information**, is that once we have fixed a strategy of Eve in G, this gives us enough information about the prefix of the play before visiting any Adam vertex. In other words, we construct a strategy of Adam tailored to spoil a specific strategy of Eve, σ, in G using the information we gather from $[\cdot]_\mathbf{1}^{-1}$ and his optimal strategy in $\hat{G}$. These properties still hold in $\tilde{G}$. Thus, we get the following result.

Lemma 7 *For payoff functions* Inf, Sup*:* $\mathbf{Reg}_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G) = -\mathbf{aVal}(\tilde{G})$.

We recall a result from [14] which gives us an algorithm for computing the value $\mathbf{Reg}_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G)$ in polynomial space. In [14] the authors show that the value of a mean-payoff game G is equivalent to the value of a finite *cycle forming game* Γ_G played on G. The game is identical to the mean-payoff game except that it is finite. The game is stopped as soon as a cycle is formed and the value of the game is given by the mean-payoff value of the cycle.

Proposition 1 *(Finite Mean-Payoff Game* [14]*) The value of a mean-payoff game G is equal to the value of the finite cycle forming game* Γ_G *played on the same weighted arena.*

As LimInf and LimSup games are also equivalent to their finite cycle forming game (see [2]) it follows that one can use an Alternating Turing Machine to compute the value of a game and that said machine will stop in time bounded by the length of the largest simple cycle in the arena. We note the length of the longest simple path in $\hat{G}$ is bounded by $|V|(|E|+1)$. Hence, we can compute the winner of $\hat{G}$ in alternating polynomial time. Since $\mathsf{APTIME} = \mathsf{PSPACE}$, this concludes the proof of Lemma 4.

4.2 Lower bounds

We give a reduction from the QSAT Problem to the problem of determining whether, given $r \in \mathbb{Q}$, $\mathbf{Reg}_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G) \lhd r$ for the payoff functions LimInf, $\underline{\mathsf{MP}}$, and $\overline{\mathsf{MP}}$ (for $\lhd \in \{<, \leq\}$). Then we provide a reduction from the complement of the 2- disjoint- paths Problem for LimSup, Sup, and Inf.

The crux of the reduction from QSAT is a gadget for each clause of the QSAT formula. Visiting this gadget allows Eve to gain information about the highest payoff obtainable in the gadget, each entry point corresponds to a literal from the clause, and the literal is visited when it is made true by the valuation of variables chosen by Eve and Adam in the reduction described below. Figure 7 depicts an instance of the gadget for a particular clause. Let us focus on the mean-payoff function. Note that staying in the inner 6-vertex triangle would yield a mean-payoff value of 4. However, in order to do so, Adam needs to cooperate with Eve at all three corner vertices. Also note that if he does cooperate in at least one of these vertices then Eve can secure a payoff value of at least $\frac{11}{3}$.

Lemma 8 *For* $r \in \mathbb{Q}$*, weighted arena G and payoff function* LimInf, $\underline{\mathsf{MP}}$*, or* $\overline{\mathsf{MP}}$*, determining whether* $\mathbf{Reg}_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G) \lhd r$*, for* $\lhd \in \{<, \leq\}$*, is* PSPACE*-hard.*

The QSAT Problem asks whether a given fully quantified boolean formula (QBF) is satisfiable. The problem is known to be PSPACE-complete [17]. 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 following 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. Also w.l.o.g., we assume that every non-trivially true clause has at least one existentially quantified variable (as otherwise the answer to the problem is trivial).

It is common to consider a QBF as a game between an existential and a universal player. The existential player chooses a truth value for existentially quantified variable x_i and the universal player responds with a truth value for x_{i+1}. After n turns the truth value of Φ determines the winner: the existential player wins if Φ is true and the universal player wins otherwise. The game we shall construct mimics the choices of the existential and universal player and makes sure that the regret of the game is small if and only if Φ is true.

Let us first consider the strict regret threshold problem. We will construct a weighted arena G in which Eve wins in the strict regret threshold problem for threshold 2 if and only if Φ is true.

Lemma 9 *For weighted arena G and payoff function* LimInf, $\underline{\mathsf{MP}}$, *or* $\overline{\mathsf{MP}}$, *determining whether* $\mathbf{Reg}_{\mathfrak{S}_\exists,\Sigma^1_\forall}(G) < 2$ *is* PSPACE*-hard.*

Proof We first describe the value-choosing part of the game (see Fig. 6). $V_\exists$ contains vertices for every existentially quantified variable from the QBF and $V \setminus V_\exists$ contains vertices for every universally quantified variable. At each of this vertices, there are two outgoing edges with weight 0 corresponding to a choice of truth value for the variable. For the variable x_i vertex, the **true** edge leads to a vertex from which Eve can choose to move to any of the clause gadgets corresponding to clauses where the literal x_i occurs (see dotted incoming edge in Fig. 7) or to advance to x_{i+1}. The **false** edge construction is similar, while leading to the literal $\overline{x_i}$ rather than to x_i. From the vertices encoding the choice of truth value for x_n Eve can either visit the clause gadgets for it or move to a "final" vertex $\Phi \in V_\exists$. This final vertex has a self-loop with weight 2.

To conclude the proof, we describe the strategy of Eve which ensures the desired property if the QBF is satisfiable and a strategy of Adam which ensures the property is falsified otherwise.

Assume the QBF is true. It follows that there is a strategy of the existential player in the QBF game such that for any strategy of the universal player the QBF will be true after they both choose values for the variables. Eve now follows this strategy while visiting all clause gadgets corresponding to occurrences of chosen literals. At every gadget clause she visits she chooses to enter the gadget. If Adam now decides to take the weight 4 edge, Eve can achieve a mean-payoff value of $\frac{11}{3}$ or a LimInf value of 3 by staying in the gadget. In this case the claim trivially holds. We therefore focus in the case where Adam chooses to take Eve back to the vertex from which she entered the gadget. She can now go to the next clause gadget and repeat. Thus, when the play reaches vertex Φ, Eve must have visited every clause gadget and Adam has chosen to disallow a weight 4 edge in every gadget. Now Eve can ensure a payoff value of 2 by going to Φ. As she has witnessed that in every clause gadget there is at least one vertex in which Adam is not helping her, alternative strategies might have ensured a mean-payoff of at most $\frac{11}{3}$ and a LimInf value of at most 3. Thus, her regret is less than 2.

Conversely, if the universal player had a winning strategy (or, in other words, the QBF was not satisfiable) then the strategy of Adam consists in following this strategy in choosing values for the variables and taking Eve out of clause gadgets if she ever enters one. If the play arrives at Φ we have that there is at least one clause gadget that was not visited by the play. We note there is an alternative strategy of Eve which, by choosing a different valuation of some variable, reaches this clause gadget and with the help of Adam achieves value 4. Hence, this strategy of Adam ensures regret of exactly 2. If Eve avoids reaching Φ then she can ensure a value of at most 0, which means an even greater regret for her. □

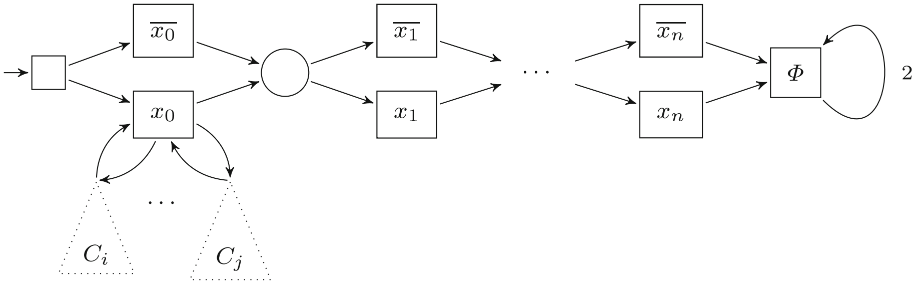

Fig. 6 Depiction of the reduction from QBF

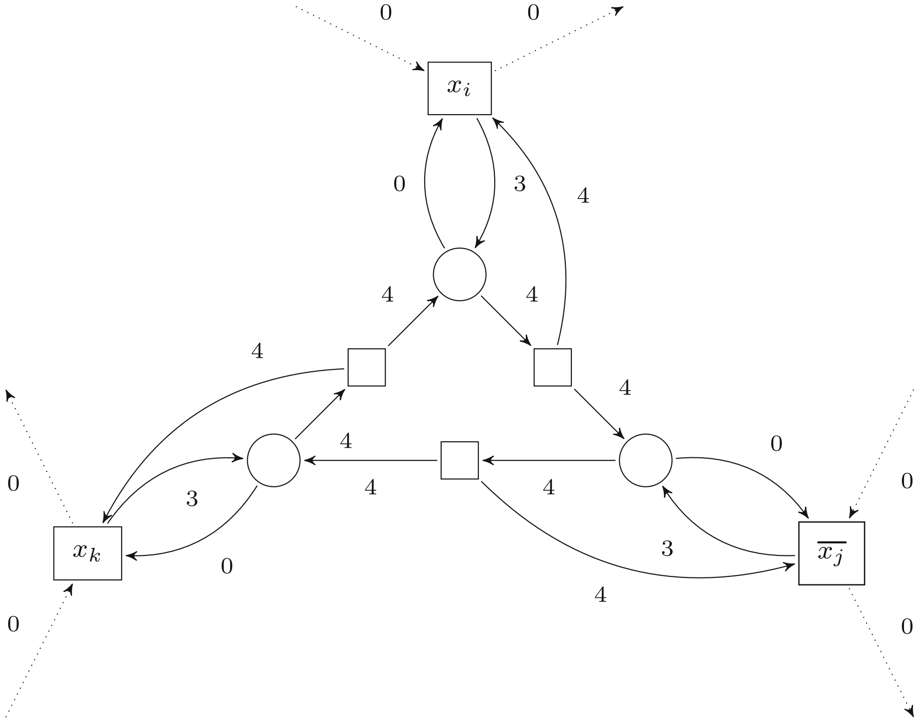

Fig. 7 Clause gadget for the QBF reduction for clause $x_i \vee \neg x_j \vee x_k$

We observe that the above reduction can be readily parameterized. That is, we can replace the 4 value, the 3 value and the 2 value by arbitrary values A, B, C satisfying the following constraints:

- $A > B > C$,
- $\frac{2A+B}{3} - C < r$ so that Eve wins if Φ is true,
- $A - C \geq r$ so that Adam wins if Φ is false, and
- $A - \frac{2A+B}{3} < r$ so that he never helps Eve in the clause gadgets.

Indeed, the valuation of A, B, C we chose: 4, 3, 2 with $r = 2$, satisfies these inequalities exactly. It is not hard to see that if we find a valuation for r, A, B, C which meets the first restriction and the last three having changed from strict to non-strict, and vice-versa, we can get a reduction that works for the non-strict regret threshold problem. That is, find values such that

- $A > B > C$,
- $\frac{2A+B}{3} - C \leq r$ so that Eve wins if Φ is true,
- $A - C > r$ so that Adam wins if Φ is false, and
- $A - \frac{2A+B}{3} \leq r$ so that he never helps Eve in the clause gadgets.

For example, one could consider $A = 10$, $B = 7$, $C = 5$ and $r = 4$.

Lemma 10 *For weighted arena G and payoff function* LimInf, $\underline{\mathsf{MP}}$, *or* $\overline{\mathsf{MP}}$, *determining whether* $\mathbf{Reg}_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G) \leq 4$ *is* PSPACE-*hard.*

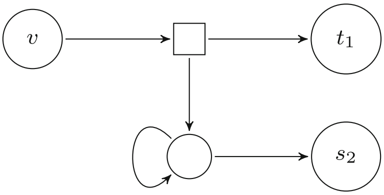

Fig. 8 Regret gadget for 2-disjoint-paths reduction

Lemma 11 *For $r \in \mathbb{Q}$, weighted arena G and payoff function* Inf, Sup, *or* LimSup, *determining whether* $\mathbf{Reg}_{\mathfrak{S}_\exists, \Sigma^1_\forall}(G) \lhd r$, *for* $\lhd \in \{<, \leq\}$, *is* coNP*-hard.*

Proof We provide a reduction from the complement of the 2- DISJOINT- PATHS PROBLEM on directed graphs [15]. As the problem is known to be NP-complete, the result follows. In other words, we sketch how to translate a given instance of the 2- DISJOINT- PATHS PROBLEM into a weighted arena in which Eve can ensure regret value strictly less than 1 if and only if the answer to the 2- DISJOINT- PATHS PROBLEM is negative.

Consider a directed graph G and distinct vertex pairs (s_1, t_1) and (s_2, t_2). W.l.o.g. we assume that for all $i \in \{1, 2\}$: (i) t_i is reachable from s_i, and (ii) t_i is a sink (i.e. has no outgoing edges). in G. We now describe the changes we apply to G in order to get the underlying graph structure of the weighted arena and then comment on the weight function. Let all vertices from G be Adam vertices and s_1 be the initial vertex. We replace all edges on t_1—edges of the form (v, t_1) incident, for some v—by a copy of the gadget shown in Fig. 8. Next, we add self-loops on t_1 and t_2 with weights 1 and 2, respectively. Finally, the weights of all remaining edges are 0.

We claim that, in this weighted arena, Eve can ensure regret strictly less than 1—for payoff functions Sup and LimSup—if and only if in G the vertex pairs (s_1, t_1) and (s_2, t_2) cannot be joined by vertex-disjoint paths. Indeed, we claim that the strategy that minimizes the regret of Eve is the strategy that, in states where she has a choice, tells her to go to t_1.

First, let us prove that this strategy has regret strictly less than 1 if and only if no two disjoint paths in the graph exist between the pairs of states (s_1, t_1) and (s_2, t_2). Assume the latter is the case. Then if Adam chooses to always avoid t_1, then clearly the regret is 0. If t_1 is eventually reached, then the choice of Eve secures a value of 1 (for all payoff functions). Note that if she had chosen to go towards s_2 instead, as there are no two disjoint paths, we know that either the path constructed from s_2 by Adam never reaches t_2, and then the value of the path is 0—and the regret is 0 for Eve—or the path constructed from s_2 reaches t_1 again since Adam is playing positionally—and, again, the regret is 0 for Eve. Now assume that two disjoint paths between the source-target pairs exist. If Eve changed her strategy to go towards s_2 (instead of choosing t_1) then Adam has a strategy to reach t_2 and achieve a payoff of 2. Thus, her regret would be equal to 1.

Second, we claim that any other strategy of Eve has a regret greater than or equal to 1. Indeed, if Eve decides to go towards s_2 (instead of choosing to go to t_1) then Adam can choose to loop on the state before s_2 and the payoff in this case is 0. Hence, the regret of Eve is at least 1.

Note that minimal changes are required for the same construction to imply the result for Inf. Further, the weight function and threshold r can be accommodated so that Eve wins for the non-strict regret threshold. Hence, the general result follows. □

4.3 Memory requirements for Eve

It follows from our algorithms for computing regret in this variant that Eve only requires strategies with exponential memory. Examples where exponential memory is necessary can be easily constructed.

Corollary 2 *For all payoff functions* Sup, Inf, LimSup, LimInf, $\underline{\mathsf{MP}}$ *and* $\overline{\mathsf{MP}}$*, for all game graphs* G*, there exists* m *which is* $2^{\mathcal{O}(|G|)}$ *such that:*

$$\mathbf{Reg}_{\mathfrak{S}_\exists,\Sigma^1_\forall}(G) = \mathbf{Reg}_{\Sigma^m_\exists,\Sigma^1_\forall}(G).$$

5 Variant III: Adam plays word strategies

For this variant, we provide tight upper and lower bounds for all the payoff functions: the regret threshold problem is $\mathsf{EXPTIME}$-complete for Sup, Inf, LimSup, and LimInf, and undecidable for $\underline{\mathsf{MP}}$ and $\overline{\mathsf{MP}}$. For the later case, the decidability can be recovered when we fix a priori the size of the memory that Eve can use to play, the decision problem is then NP-complete. Finally, we show that our notion of regret minimization for word strategies generalizes the notion of *good for games* introduced by Henzinger and Piterman in [19], and we also formalize the relation that exists with the notion of determinization by pruning for weighted automata introduced by Aminof et al. in [1].

5.1 Additional definitions

We say that a strategy of Adam is a *word strategy* if his strategy can be expressed as a function $\tau : \mathbb{N} \to [\max\{deg^+(v) \mid v \in V\}]$, where $[n] = \{i \mid 1 \leq i \leq n\}$ and $deg^+(v)$ 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 every turn, the strategy τ of Adam will tell him to move to the i-th successor (or to a sink state, if its outdegree is less than i) of the vertex according to the fixed order. We denote by $\mathfrak{W}_\forall$ the set of all such strategies for Adam. When considering word strategies, it is more natural to see the arena as a (weighted) automaton.

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, $w : \Delta \to \mathbb{Z}$ assigns weights to transitions. A *run* of Γ on a word $a_0a_1 \ldots \in A^\omega$ is a sequence $\rho = q_0a_0q_1a_1 \ldots \in (Q \times A)^\omega$ such that $(q_i, a_i, q_{i+1}) \in \Delta$, for all $i \geq 0$, and has *value* $\mathbf{Val}(\rho)$ determined by the sequence of weights of the transitions of the run and the payoff function. The value Γ assigns to a word is the supremum of the values of all its runs on the word. We say the automaton is deterministic if Δ is functional.

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 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. the value of the best run on the word spelled out by Adam. For instance, if all vertices in Fig. 1 are replaced by states, Eve can choose the successor of v_1 regardless of what letter Adam plays and from v_2 and v_3 Adam chooses the successor by choosing to play a or b. Furthermore, his choice of letter tells Eve what would have happened had the play been at the other state.

The following result summarizes the results of this section:

Theorem 3 *Deciding if the regret value is less than a given threshold (strictly or non-strictly) playing against word strategies of Adam is* EXPTIME*-complete for* Inf*,* Sup*,* LimInf*, and* LimSup*; it is undecidable for* $\underline{\mathsf{MP}}$ *and* $\overline{\mathsf{MP}}$.

5.2 Upper bounds

There is an EXPTIME algorithm for solving the regret threshold problem for Inf, Sup, LimInf, and LimSup. This algorithm is obtained by a reduction to parity and Streett games.

Lemma 12 *For* $r \in \mathbb{Q}$*, weighted automaton* Γ *and payoff function* Inf*,* Sup*,* LimInf*, or* LimSup*, determining whether* $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\Gamma) \lhd r$*, for* $\lhd \in \{<, \leq\}$*, can be done in exponential time.*

We show how to decide the strict regret threshold problem. However, the same algorithm can be adapted for the non-strict version by changing strictness of the inequalities used to define the parity/Streett accepting conditions.

Proof We focus on the LimInf and LimSup payoff functions. The result for Inf and Sup follows from the translation to LimInf and LimSup games given in Sect. 3. Our decision algorithm consists in first building a deterministic automaton for $\Gamma = (Q_1, q_I, A, \Delta_1, w_1)$ using the construction provided in [9]. We denote by $D_\Gamma = (Q_2, s_I, A, \Delta_2, w_2)$ this deterministic automaton and we know that it is at most exponentially larger than Γ. Next, we consider a simulation game played by Eve and Adam on the automata Γ and D_Γ. The game is played for an infinite number of rounds and builds runs in the two automata, it starts with the two automata in their respective initial states (q_I, s_I), and if the current states are q_1 and q_2, then the next round is played as follows:

- Adam chooses a letter $a \in A$, and the state of the deterministic automaton is updated accordingly, i.e. $q_2' = \Delta_2(q_2, a)$, then
- Eve updates the state of the non-deterministic automaton to q_1' by reading a using one of the edges labelled with a in the current state, i.e. she chooses q_1' such that $q_1' \in \Delta_1(q_1, a)$. The new state of the game is (q_1', q_2').

Eve wins the simulation game if the minimal weight seen infinitely often in the run of the non-deterministic automaton is larger than or equal to the minimal weight seen infinitely often in the deterministic automaton minus r. It should be clear that this happens exactly when Eve has a regret bounded by r in the original regret game on the word which is spelled out by Adam.

Let us focus on the lim inf payoff function now. We will sketch how this game can be translated into a parity game. For completeness, we now provide a formal definition of the latter.

A parity game is a pair (G, Ω) where G is a non-weighted arena and $\Omega : V \to \mathbb{N}$ is a function that assigns a priority to each vertex. Plays, strategies, and other notions are defined as with games played on weighted arenas. A play in a parity game induces an infinite sequence of priorities. We say a play is winning for Eve if and only if the minimal priority seen infinitely often is odd. The *parity index* of a parity game is the number of priorities labelling its vertices, that is $|\{\Omega(v) \mid v \in V\}|$.

To obtain the translation, we keep the structure of the game as above but we assign priorities to the edges of the games instead of weights. We do it in the following way. If $X = \{x_1, x_2, \ldots, x_n\}$ is the ordered set of weight values that appear in the automata (note that $|X|$ is bounded by the number of edges in the non-deterministic automaton), then we need the set of priorities $D = \{2, \ldots, 2n + 1\}$. We assign priorities to edges in the game as follows:

- when Adam chooses a letter a from q_2, then if the weight that labels the edge that leaves q_2 with letter a in the deterministic automaton is equal to $x_i \in X$, then the priority is set to $2i + 1$,
- when Eve updates the non-deterministic automaton from q_1 with a edge labelled with weight w, then the color is set to $2i$ where i is the index in X such that $x_{i-1} \leq w + r < x_i$.

It should be clear then along a run, the minimal color seen infinitely often is odd if and only if the corresponding run is winning for Eve in the simulation game. So, now it remains to solve a parity game with exponentially many states and polynomially many priorities w.r.t. the size of Γ. This can be done in exponential time with classical algorithms for parity games.

LimSup *to Streett games.* Let us now focus on LimSup. In this case we will reduce our problem to that of determining the winner of a Streett game with state-space exponential w.r.t. the original game but with number of Streett pairs polynomial (w.r.t. the original game). Recall that a Streett game is a pair $(G, \mathcal{F})$ where G is a game graph (with no weight function) and $\mathcal{F} \subseteq \mathcal{P}(V) \times \mathcal{P}(V)$ is a set of Streett pairs. We say a play is winning for Eve if and only if for all pairs $(E, F) \in \mathcal{F}$, if a vertex in E is visited infinitely often then some vertex in F is visited infinitely often as well.

Consider a LimSup automaton $\Gamma = (Q, q_I, A, \Delta, w)$. For $x_i \in \{w(d) \mid d \in \Delta\}$ let us denote by $\mathcal{A}^{\geq x_i}$ the Büchi automaton with Büchi transition set equivalent to all transitions with weight of at least x_i. We denote by $\mathcal{D}^{\geq x_i} = (Q_i, q_{i,I}, A, \delta_i, \Omega_i)$ the deterministic parity automaton with the same language as $\mathcal{A}^{\geq x_i}$.[3] From [24] we have that $\mathcal{D}^{\geq x_i}$ has at most $2|Q|^{|Q|}|Q|!$ states and parity index $2|Q|$ (the number of priorities). Now, let $x_1 < x_2 < \cdots < x_l$ be the weights appearing in transitions of Γ. We construct the (non-weighted) arena $G_\Gamma = (V, V_\exists, E, v_I)$ and Streett pair set $\mathcal{F}$ as follows

- $V = Q \times \prod_{i=1}^{l} Q_i \cup Q \times \prod_{i=1}^{l} Q_i \times A \cup Q \times \prod_{i=1}^{l} Q_i \times A \times Q$;
- $V_\exists = Q \times \prod_{i=1}^{l} Q_i \times A$;
- $v_I = (q_I, q_{1,I}, \ldots, q_{l,I})$;
- E contains
  - $\big((p, p_1, \ldots, p_l)), (p, p_1, \ldots, p_l, a)\big)$ for all $a \in A$,
  - $\big((p, p_l, \ldots, p_l, a), (p, p_1, \ldots, p_l, a, q)\big)$ if $(p, a, q) \in \Delta$,
  - $\big((p, p_l, \ldots, p_l, a, q), (q, q_1, \ldots, q_l)\big)$ if for all $1 \leq i \leq l$: $(p_i, a, q_i) \in \delta_i$;
- For all $1 \leq i \leq l$ and all even y such that $\text{Range}(\Omega_i) \ni y$, $\mathcal{F}$ contains the pair (E_i, F_i) where
  - $E_{i,y} = \{(p, \ldots, p_i, \ldots, p_l, a, q) \mid \Omega_i(p_i, a, \delta(p_i, a)) = y\}$, and
  - $F_{i,y} = \{(p, \ldots, p_j, \ldots, p_l, a, q) \mid (\Omega_i(p_i, a, \delta(p_i, a)) < y \wedge y \pmod 2 = 1) \vee w(p, a, q) \geq x_i - r\}$.

It is not hard to show that in the resulting Streett game, a strategy σ of Eve is winning against any strategy τ of Adam if and only if for every automaton $\mathcal{D}^{\geq x_i}$ which accepts the word induced by τ then the run of Γ induced by σ has payoff of at least $x_i - r$, if and only if Eve has a winning strategy in Γ to ensure regret is less than r.

[3] Since δ_i is deterministic, we sometimes write $\delta_i(p, a)$ to denote the unique $q \in Q_i$ such that $(p, a, q) \in \delta_i$.

Note that the number of Streett pairs in G_Γ is polynomial w.r.t. the size of Γ, i.e.

$$\begin{aligned}|\mathcal{F}| &\leq \sum_{i=0}^{l} |\mathrm{Range}(\Omega_i)|\\ &\leq l \cdot 2|Q|\\ &\leq |Q|^2 \cdot 2|Q| = 2|Q|^3.\end{aligned}$$

From [25] we have that Streett games can be solved in time $\mathcal{O}(mn^{k+1}kk!)$ where n is the number of states, m the number of transitions and k the number of pairs in $\mathcal{F}$. Thus, in this case we have that G_Γ can be solved in

$$\mathcal{O}\big((2|Q|^{|Q|}|Q|!)^{3+2|Q|^3} \cdot 2|Q|^3 \cdot (2|Q|^3)!\big).$$

which is still exponential time w.r.t. the size of Γ. □

5.3 Lower bounds

We first establish EXPTIME-hardness for the payoff functions Inf, Sup, LimInf, and LimSup by giving a reduction from countdown games [22]. That is, we show that given a countdown game, we can construct a game where Eve ensures regret less than 2 if and only if Counter wins in the original countdown game.

Lemma 13 *For $r \in \mathbb{Q}$, weighted automaton Γ and payoff function* Inf, Sup, LimInf, *or* LimSup, *determining whether $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\Gamma) \lhd r$, for $\lhd \in \{<, \leq\}$, is* EXPTIME*-hard.*

Let us first formalize what a countdown game is. A countdown game $\mathcal{C}$ consists of a weighted graph (S, T), where S is the set of states and $T \subseteq S \times (\mathbb{N}\backslash\{0\}) \times S$ is the transition relation, and a target value $N \in \mathbb{N}$. If $t = (s, d, s') \in T$ then we say that the duration of the transition t is d. A configuration of a countdown game is a pair (s, c), where $s \in S$ is a state and $c \in \mathbb{N}$. A move of a countdown game from a configuration (s, c) consists in player Counter choosing a duration d such that $(s, d, s') \in T$ for some $s' \in S$ followed by player Spoiler choosing s'' such that $(s, d, s'') \in T$, the new configuration is then $(s'', c + d)$. Counter wins if the game reaches a configuration of the form (s, N) and Spoiler wins if the game reaches a configuration (s, c) such that $c < N$ and for all $t = (s, d, \cdot) \in T$ we have that $c + d > N$.

Deciding the winner in a countdown game $\mathcal{C}$ from a configuration $(s, 0)$ – where N and all durations in $\mathcal{C}$ are given in binary – is EXPTIME-complete.

Proof (of Lemma 13) Let us fix a countdown game $\mathcal{C} = ((S, T), N)$ and let $n = \lfloor \log_2 N \rfloor + 2$.

Simplifying assumptions. Clearly, if Spoiler has a winning strategy and the game continues beyond his winning the game, then eventually a configuration (s, c), such that $c \geq 2^n$, is reached. Thus, we can assume w.l.o.g. that plays in $\mathcal{C}$ which visit a configuration (s, N) are winning for Counter and plays which don't visit a configuration (s, N) but eventually get to a configuration (s', c) such that $c \geq 2^n$ are winning for Spoiler.

Additionally, we can also assume that T in $\mathcal{C}$ is total. That is to say, for all $s \in S$ there is some duration d such that $(s, d, s') \in T$ for some $s' \in S$. If this were not the case then for every s with no outgoing transitions we could add a transition $(s, N + 1, s_\bot)$ where $s_\bot$ is a newly added state. It is easy to see that either player has a winning strategy in this new game if and only if he has a winning strategy in the original game.

Reduction. We will now construct a weighted arena Γ with $W = 2$ such that, in a regret game with payoff function Sup played on Γ, Eve can ensure regret value strictly less than 2 if and only if Counter has a winning strategy in $\mathcal{C}$.

As all weights are 0 in the arena we build, with the exception of self-loops on sinks, the result holds for Sup, LimSup and Inf. We describe the changes required for the inf result at the end.

Implementation. The alphabet of the weighted arena $\Gamma = (Q, q_I, A, \Delta, w)$ is $A = \{b_i \mid 0 \leq i \leq n\} \cup \{c_i \mid 0 < i \leq n\} \cup \{bail, choose\} \cup S$. We now describe the structure of Γ (i.e. Q, Δ and w).

Initial gadget. Figure 9 depicts the initial state of the arena. Here, Eve has the choice of playing left or right. If she plays to the left then Adam can play *bail* and force her to $\bot_0$ while the alternative play resulting from her having chosen to go right goes to $\bot_2$. Hence, playing left already gives Adam a winning strategy to ensure regret 2, so she plays to the right. If Adam now plays *bail* then Eve can go to $\bot_2$ and as $W = 2$ this implies the regret will be 0. Therefore, Adam plays anything but *bail*.

Counter gadget. Figure 10 shows the left sub-arena. All states from $\{\overline{x_i} \mid 0 \leq i \leq n\}$ have incoming transitions from the left part of the initial gadget with symbol $A \backslash \{bail\}$ and weight 0. Let $y_0 \ldots y_n \in \mathbb{B}$ be the (little-endian) binary representation of N, then for all x_i such that $y_i = 1$ there is a transition from x_i to $\bot_0$ with weight 0 and symbol *bail*. Similarly, for all $\overline{x_i}$ such that $y_i = 0$ there is a transition from $\overline{x_i}$ to $\bot_0$ with weight 0 and symbol *bail*. All the remaining transitions not shown in the figure cycle on the same state, e.g. x_i goes to x_i with symbol *choose* and weight 0.

The sub-arena we have just described corresponds to a counter gadget (little-endian encoding) which keeps track of the sum of the durations "spelled" by Adam. At any point in time, the states of this sub-arena in which Eve believes alternative plays are now will represent the binary encoding of the current sum of durations. Indeed, the initial gadget makes sure Eve plays into the right sub-arena and therefore she knows there are alternative play prefixes that could be at any of the $\overline{x_i}$ states. This corresponds to the 0 value of the initial configuration.

Adder gadget. Let us now focus on the right sub-arena in which Eve finds herself at the moment. The right transition with symbol $A \backslash \{bail\}$ from the initial gadget goes to state s – the initial state from $\mathcal{C}$. It is easy to see how we can simulate Counter's choice of duration

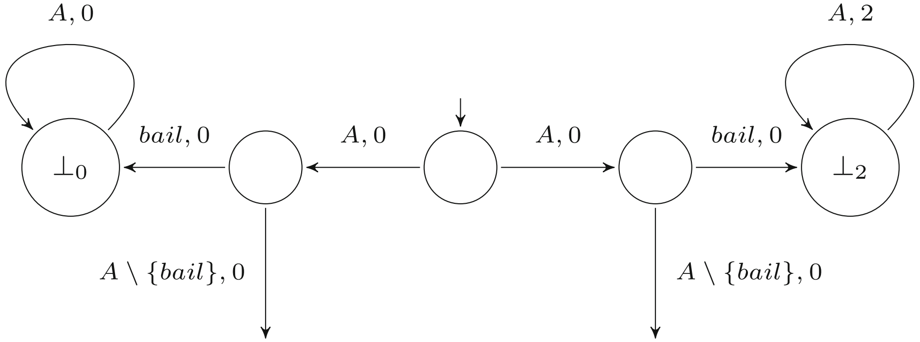

Fig. 9 Initial gadget used in reduction from countdown games

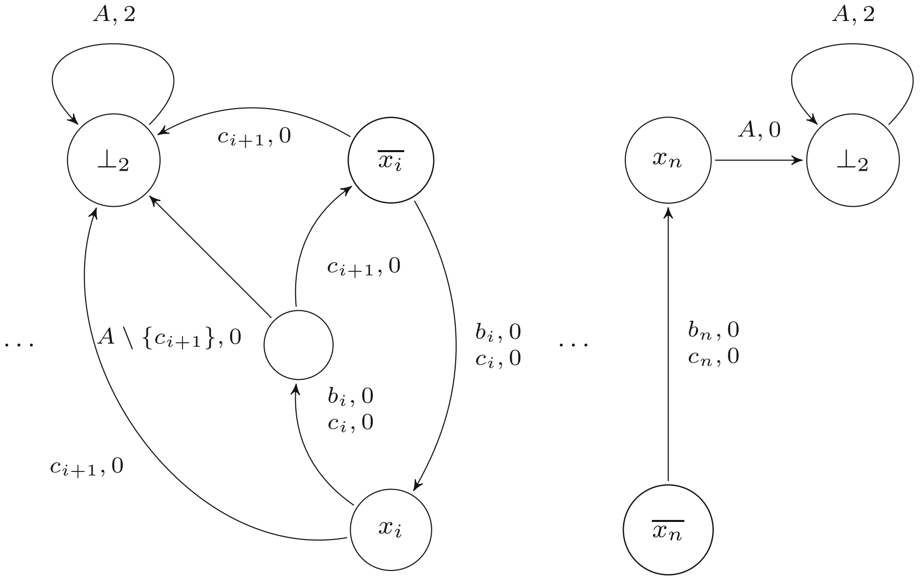

Fig. 10 Counter gadget

and Spoiler's choice of successor. From s there are transitions to every (s, c), such that $(s, c, s') \in T$ for some $s' \in S$ in $\mathcal{C}$, with symbol *choose* and weight 0. Transitions with all other symbols and weight 0 going to $\bot_1$ – a sink with a 1-weight cycle with every symbol – from s ensure Adam plays *choose*, lest since $W = 2$ the regret of the game will be at most 1 and Eve wins.

Figure 11 shows how Eve forces Adam to "spell" the duration c of a transition of $\mathcal{C}$ from (s, c). For concreteness, assume that Eve has chosen duration 9. The top source in Fig. 11 is therefore the state $(s, 9)$. Again, transitions with all the symbols not depicted go to $\bot_1$ with weight 0 are added for all states except for the bottom sink. Hence, Adam will play b_0 and Eve has the choice of going straight down or moving to a state where Adam is forced to play c_1. Recall from the description of the counter gadget that the belief of Eve encodes the binary representation of the current sum of delays. If she believes a play is in x_1 (and therefore none in $\overline{x_1}$) then after Adam plays b_0 it is important for her to make him play c_1 or this alternative play will end up in $\bot_2$. It will be clear from the construction that Adam always has a strategy to keep the play in the right sub-arena without reaching $\bot_1$ and therefore if any alternative play from the left sub-arena is able to reach $\bot_2$ then Adam wins (i.e. can ensure regret 2). Thus, Eve decides to force Adam to play c_1. As the duration was 9 this gadget now forces Adam to play b_4 and again presents the choice of forcing Adam to play c_5 to Eve. Clearly this can be generalized for any duration. This gadget in fact simulates a *cascade configuration* of n 1-bit adders.

Finally, from the bottom sink in the adder gadget, we have transitions with symbols from S with weight 0 to the corresponding states (thus simulating Spoiler's choice of successor state). Additionally, with any symbol from S and with weight 0 Eve can also choose to go to a state q_{bail} where Adam is forced to play *bail* and Eve is forced into $\bot_0$.

Argument. Note that if the simulation of the counter has been faithful and the belief of Eve encodes the value N then by playing *bail*, Adam forces all of the alternative plays in the left sub-arena into the $\bot_0$ sink. Hence, if Counter has a winning strategy and Eve faithfully

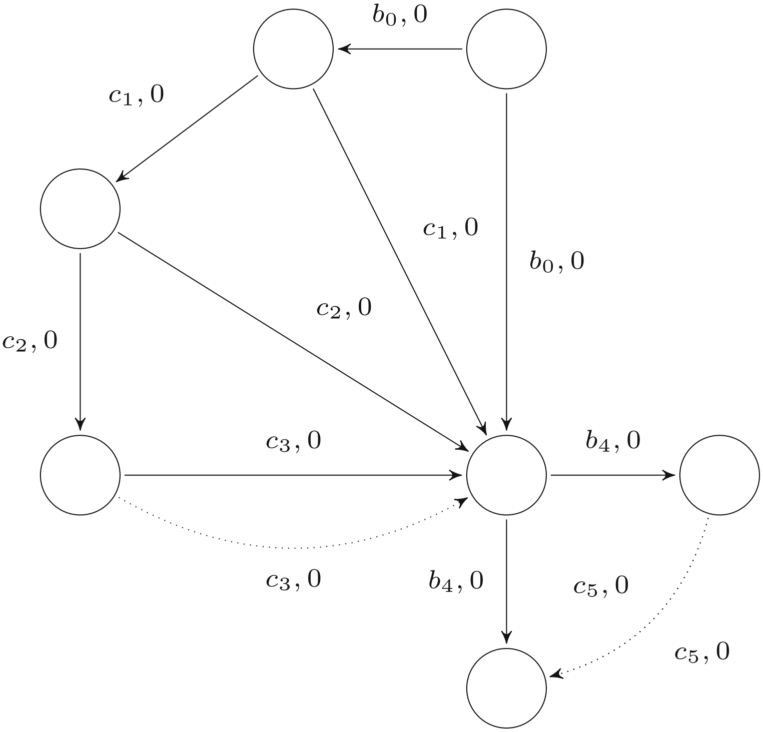

Fig. 11 Adder gadget: depicted $+9$

simulates the $\mathcal{C}$ she can force this outcome of all plays going to $\bot_0$. Note that from the right sub-arena we have that $\bot_2$ is not reachable and therefore the highest payoff achievable was 1. Therefore, her regret is of at most 1.

Conversely, if both players faithfully simulate $\mathcal{C}$ and the configuration N is never reached, i.e. Spoiler had a winning strategy in $\mathcal{C}$ then eventually some alternative play in the left sub-arena will reach x_n and from there it will go to $\bot_2$. Again, the construction makes sure that Adam always has a strategy to keep the play in the right sub-arena from reaching $\bot_1$ and therefore this outcome yields a regret of 2 for Eve.

Changes for Inf. For the same reduction to work for the Inf payoff function we add an additional symbol $kick$ to the alphabet of Γ. We also add deterministic transitions with $kick$, from all states which are not sinks $\bot_x$ for some x, to $\bot_0$. Finally, all non-loop transitions in the initial gadget are now given a weight of 2; the ones in the counter gadget are given a weight of 2 as well; the ones in the adder gadget (i.e. right sub-arena) are given a weight of 1.

We observe that if Counter has a winning strategy in the original game $\mathcal{C}$ then Eve still has a winning strategy in Γ. The additional symbol $kick$ allows Adam to force Eve into a 0-loop but also ensures that all alternative plays also go to $\bot_0$, thus playing $kick$ is not beneficial to Adam unless an alternative play is already at $\bot_2$. Conversely, if Spoiler has a winning strategy in $\mathcal{C}$ then Adam has a strategy to allow an alternative play to reach $\bot_2$ while Eve remains in the adder gadget. He can then play $kick$ to ensure the payoff of Eve is 0 and achieve a maximal regret of 2.

Once again, we observe that the above reduction can be readily parameterized. That is, we can replace the 2 value, the 1 value and the 0 value from the $\bot_2$, $\bot_1$, $\bot_0$ sink loops by arbitrary values A, B, C satisfying the following constraints:

- $A > B > C$,
- $A - C \geq r$ so that Eve loses by going left in the initial gadget,
- $A - B < r$ so that she does not lose by faithfully simulating the adder if she has a winning strategy from the countdown game, or in other words: if Adam cheats then $A - B$ is low enough to punish him,
- $B - C < r$ so that she does not regret having faithfully simulated addition, that is, if she plays her winning strategy from the countdown game then she does not consider $B - C$ too high and regret it.

Changing the strictness of the last three constraints and finding a suitable valuation for r and A, B, C suffices for the reduction to work for the non-strict regret threshold problem. Such a valuation is given by $A = 2$, $B = 1$, $C = 0$ with $r = 1$. □

To show undecidability of the problem for the mean-payoff function we give a reduction from the threshold problem in *mean-payoff games with partial-observation*. This problem was shown to be undecidable in [12,20].

Lemma 14 *For $r \in \mathbb{Q}$, weighted automaton Γ and payoff function $\underline{\mathsf{MP}}$ or $\overline{\mathsf{MP}}$, determining whether $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\Gamma) \lhd r$, for $\lhd \in \{<, \leq\}$, is undecidable even if Eve is only allowed to play finite memory strategies.*

A mean-payoff game with partial-observation (MPGPO for short) G is a tuple $(Q, q_I, A, \Delta, w, Obs)$ where Q is a set of states, q_I is the initial state of the game, A is a finite set of actions, $\Delta \subseteq Q \times A \times Q$ is the transition relation, $w : \Delta \to \mathbb{Q}$ is a weight function and $Obs \subseteq \mathcal{P}(Q)$ is a partition of Q into *observations*. In these games a play is started by placing a token on q_I, Eve then chooses an action from A and Adam resolves non-determinism by choosing a valid successor (w.r.t. Δ). Additionally, Eve does not know which state Adam has chosen as the successor, she is only revealed the observation containing the state. More formally: a *concrete play* in such a game is a sequence $q_0 a_0 q_1 a_1 \ldots \in (Q \times A)^\omega$ such that $q_0 = q_I$ and $(q_i, a_i, q_{i+1}) \in \Delta$, for all $i \geq 0$. An *abstract play* is then a sequence $\psi = o_0 a_0 o_1 a_1 \ldots \in (Obs \times A)^\omega$ such that there is some concrete play $\pi = q_0 a_0 q_1 a_1 \ldots$ and $q_i \in o_i$, for all $i \geq 0$; in this case we say that π is a *concretization* of ψ. Strategies of Eve in this game are of the form $\sigma : (Obs \times A)^* Obs \to A$, that is to say they are *observation-based*. Strategies of Adam are not symmetrical, he is allowed to use the exact state information, i.e. his strategies are of the form $\tau : (Q \times A)^* \to Q$.

The threshold problem for mean-payoff games is defined as follows. Given $\nu \in \mathbb{Q}$, determining whether Eve has an observation-based strategy such that, for all counter-strategies of Adam, the resulting abstract play has no concretization with mean-payoff value (strictly) less than ν. For convenience, let us denote this problem by $\mathbf{maxMPGPO}(>\nu)$ and by $\mathbf{maxMPGPO}(\geq\nu)$ when the inequality is strict and non-strict, respectively. Note that in this case Eve is playing to maximize the mean-payoff value of all concrete runs corresponding to the abstract play being played while Adam is minimizing the same.

It was shown in [12,20] that both problems are undecidable for $\underline{\mathsf{MP}}$ and for $\overline{\mathsf{MP}}$. That is, determining if $\mathbf{maxMPGPO}(>\nu)$ or $\mathbf{maxMPGPO}(\geq\nu)$ is undecidable regardless of the definition used for the mean-payoff function. Further, if we ask for the existence of finite memory observation-based strategies of Eve only, both definitions ($\underline{\mathsf{MP}}$ and $\overline{\mathsf{MP}}$) coincide and the problem remains undecidable.

Consider a given MPGPO $H = (Q, q_I, A, \Delta, w, Obs)$, and denote by H' the game obtained by multiplying by -1 all values assigned by w to the transitions of H. Clearly, we get that the answer to whether **maxMPGPO**$(>\nu)$ (resp. **maxMPGPO**$(\geq\nu)$) in H is affirmative if and only if in H' Eve has an observation-based strategy to ensure that against any strategy of Adam, the resulting abstract play is such that all concretizations have mean-payoff value of less than or equal to $-\nu$ (resp. strictly less than $-\nu$). Denote these problems by **minMPGPO**$(<\mu)$ and **minMPGPO**$(\leq\mu)$, respectively. It follows that for any definition of the mean-payoff function, these problems are undecidable (even if we are only interested in finite memory strategies of Eve).

Simplifying assumptions. We assume, w.l.o.g., that in mean-payoff games with partial-observation the transition relation is total. As the weights in mean-payoff games with partial-observation can be *shifted and scaled*, we can assume w.l.o.g. that ν is any integer N. Furthermore, we can also assume that the mean-payoff value of any concrete play in a game is bounded from below by 0 and from above by M (this can again be achieved by shifting and scaling).

Proof (of Lemma 14) We give a reduction from the threshold problem of *mean-payoff games with partial observation* [12,20] that resembles the reduction used for the proof of Lemma 13. More specifically, given a mean-payoff game with partial-observation $H = (S, s_I, T, B, c, Obs)$, we construct a weighted automaton $\Gamma_H = (Q, q_I, \Delta, A, w)$ with the same payoff function such that

$$\mathbf{Reg}_{\Sigma_\exists, \Sigma_\forall^1}(\Gamma_H) < R$$

if and only if the answer to **minMPGPO**$(<N)$ is affirmative. The reduction we describe works for any R, N, M, C such that

- $C < R$,
- $\frac{M}{2} - C < R$, and
- $\frac{N}{2} \leq R$,

for concreteness we consider $R = 4$, $N = 4$, $M = 6$ and $C = 3$.

Let us describe how to construct the weighted arena Γ_H from G. The alphabet of Γ_H is $A = B \cup \{bail\} \cup Obs$. The structure of Γ_H includes a gadget such as the one depicted in Fig. 9. Recall from the proof of Lemma 12 that this gadget ensures Eve chooses to go to the right sub-arena, lest Adam has a spoiling strategy. As the left sub-arena we have a modified version of H. First, for every state $s \in S$ and every action $b \in B$, we add an intermediate state (s, b) such that when b is played from s the play deterministically goes to (s, b) and for any transition (s, b, s') in H we add a transition in Γ_H from (s, b) to s' with action $o_{s'}$, where $o_{s'}$ is the observation containing s'. Second, we add transitions from every $s \in S$ to $\bot_C$ for symbol $bail$ with weight 0 and from every (s, b) to $\bot_C$ with symbol o if there is no $s' \in o$ such that $(s, b, s') \in T$. The sink $\bot_C$ has, for every symbol $a \in A$, a weight C self-loop. As the right sub-arena we will have states q_b for all $b \in B$. For any such q_b there are transitions with weight 0 and symbol b to q_{obs} and transitions with weight 0 and symbols $A\backslash\{b\}$ to $\bot_C$. From q_{obs} with any symbol from Obs, there are 0-weight transitions to $q_{b'}$ (for any $b' \in B$) and transitions with weight 0 and symbols $A\backslash Obs$ to $\bot_C$. All q_b have incoming edges from the state of the initial gadget which leads to the right sub-arena.

We claim that Eve has a strategy σ in Γ_H to ensure regret less than R if and only if the answer to **minMPGPO**$(<N)$ is affirmative. Assume that the latter is the case, i.e. in H Eve has an observation-based strategy to ensure that against any strategy of Adam the abstract play has no concretization with mean-payoff value greater than or equal to N. Let us describe the strategy of Eve in Γ_H. First, she plays into the right sub-arena of the game. Once there,

she tries to visit states $q_{b_0}q_{b_1}\dots$ based on her strategy for H. If Adam, at some q_{b_i} does not play b_i, or at some visit to q_{obs} he plays a non-observation symbol, then Eve goes to $\bot_C$. The play then has value C. Since no alternative play in the left sub-arena can have value greater than $\frac{M}{2}$ and we have that $\frac{M}{2} - C < R$, Eve wins. Thus, we can assume that Adam, at every q_{b_i} plays the symbol b_i and at every visit to q_{obs} plays an observation. Note that, by construction of the left sub-arena, we are forcing Adam to reveal a sequence of observations to Eve and allowing her to choose a next action. It follows that the value of the play in Γ_H is 0. Any alternative play in the right sub-arena would have value of at most C as the highest weight in it is C. In the left sub-arena, we have that all alternative plays have value less than $\frac{N}{2}$. Indeed, since she has followed her winning strategy from H, and since by construction we have that all alternative plays in the left sub-arena correspond to concretizations of the abstract path spelled by Adam and Eve, if there were some play with value of at least $\frac{N}{2}$ this would contradict her strategy being optimal. As $C < R$ and $\frac{N}{2} < R$, we have that Eve wins the regret game, i.e. her strategy ensures regret less than R.

Conversely, assume that the answer to **minMPGPO**($<N$) is negative. Then regardless, of which strategy from H Eve decides to follow, we know there will be some alternative play in the left sub-arena with value of at least $\frac{N}{2}$. If Adam allows Eve to play any such strategy then the value of the play is 0 and her regret is at least $\frac{N}{2} \leq R$, which concludes the proof for the strict regret threshold problem.

We observe that the restriction on N, M, R and C can easily be adapted to allow for a reduction from **minMPGPO**($\leq N$) to the non-strict regret threshold problem.

Finally, we note that in the above proof Eve might require infinite memory as it is known that in mean-payoff games with partial-observation the protagonist might require infinite memory to win. Yet, as we have already mentioned, even if we ask whether Eve has a winning finite memory observation-based strategy, the problem remains undecidable. Notice that the above construction–when restricting Eve to play with finite memory–gives us a reduction from this exact problem. Hence, even when restricting Eve to use only finite memory, the problem is undecidable. □

5.4 Memory requirements for Eve and Adam

It is known that positional strategies suffice for Eve in parity games. On the other hand, for Streett games she might require exponential memory (see, e.g. [13]). This exponential blow-up, however, is only on the number of pairs—which we have already argued remains polynomial w.r.t. the original automaton. It follows that:

Corollary 3 *For payoff functions* Sup, Inf, LimSup, LimInf, *for all weighted automata $\mathcal{A}$, there exists m which is $2^{\mathcal{O}(|\mathcal{A}|)}$ such that:*

$$\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{A}) = \mathbf{Reg}_{\Sigma_\exists^m,\mathfrak{W}_\forall}(\mathcal{A}).$$

5.5 Fixed memory for Eve

Since the problem is EXPTIME-hard for most payoff functions and already undecidable for $\underline{\mathsf{MP}}$ and $\overline{\mathsf{MP}}$, we now fix the memory Eve can use.

Theorem 4 *For $r \in \mathbb{Q}$, weighted automaton Γ and payoff function* Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, *or* $\overline{\mathsf{MP}}$, *determining whether $\mathbf{Reg}_{\Sigma_\exists^m,\mathfrak{W}_\forall}(\Gamma) \lhd r$, for $\lhd \in \{<, \leq\}$, can be done in* NTIME$(m^2|\Gamma|^2)$.

Denote by $\mathfrak{R}_\forall \subseteq \mathfrak{W}_\forall$ the set of all word strategies of Adam which are regular. That is to say, $w \in \mathfrak{R}_\forall$ if and only if w is *ultimately periodic*. It is well-known that the mean-payoff value of ultimately periodic plays in weighted arenas is the same for both $\overline{\mathsf{MP}}$ and $\underline{\mathsf{MP}}$.

Before proving the theorem we first show that ultimately periodic words suffice for Adam to spoil a finite memory strategy of Eve. Let us fix some useful notation. Given weighted automaton Γ and a finite memory strategy σ for Eve in Γ we denote by Γ_σ the deterministic automaton embodied by a refinement of Γ that is induced by σ.

Lemma 15 *For* $r \in \mathbb{Q}$*, weighted automaton* Γ*, and payoff function* Inf*,* Sup*,* LimInf*,* LimSup*,* $\underline{\mathsf{MP}}$*, or* $\overline{\mathsf{MP}}$*, if* $\mathbf{Reg}_{\Sigma_\exists^m,\mathfrak{W}_\forall}(\Gamma) \rhd r$ *then* $\mathbf{Reg}_{\Sigma_\exists^m,\mathfrak{R}_\forall}(\Gamma) \rhd r$*, for* $\rhd \in \{>, \geq\}$.

Proof For Inf, Sup, LimInf, and LimSup the result follows from Lemma 12. It is known that positional strategies suffice for either player to win a parity game. Thus, if Adam wins the parity game defined in the proof of Lemma 12 then he has a positional strategy to do so. Now, for any strategy of Eve in the original game, one can translate the winning strategy of Adam in the parity game into a spoiling strategy of Adam in the regret game. This strategy will have finite memory and will thus correspond to an ultimately periodic word. Hence, it suffices for us to show the claim follows for mean-payoff. We do so for $\underline{\mathsf{MP}}$ and $\geq$ but the result for $\overline{\mathsf{MP}}$ follows from minimal changes to the argument (a small quantifier swap in fact) and for $>$ variations we need only use the strict versions of Eqs. (*) and (11). We assume without loss of generality that all weights are non-negative.

Let σ be the best (regret minimizing) strategy of Eve in Γ which uses at most memory m. We claim that if Adam has a word strategy to ensure the regret of Eve in Γ is at least r then he also has a regular word strategy to do so.

Consider the bi-weighted graph G constructed by taking the synchronous product of Γ and Γ_σ while labelling every edge with two weights: the value assigned to the transition by the weight function of Γ_σ and the value assigned to the transition by that of Γ. For a path π in G, denote by $w_i(\pi)$ the sum of the weights of the edges traversed by π w.r.t. the i-th weight function. Also, for an infinite path π, denote by $\underline{\mathsf{MP}}_i$ the mean-payoff value of π w.r.t. the i-th weight function. Clearly, Adam has a word strategy to ensure a regret of at least r against the strategy σ of Eve if and only if there is an infinite path π in G such that $\underline{\mathsf{MP}}_2(\pi) - \underline{\mathsf{MP}}_1(\pi) \geq r$. We claim that if this is the case then there is a simple cycle χ in G such that $\frac{1}{|\chi|}w_2(\chi) - \frac{1}{|\chi|}w_1(\chi) \geq r$. The argument is based on the cycle decomposition of π (see, e.g. [14]).

Assume, for the sake of contradiction, that all the cycles χ in G satisfy the following:

$$\frac{1}{|\chi|}w_2(\chi) - \frac{1}{|\chi|}w_1(\chi) \leq r - \epsilon, \text{ for some } 0 < \epsilon \leq r, \tag{*}$$

and let us consider an arbitrary infinite path $\pi = v_0v_1 \ldots$. Let $l = \underline{\mathsf{MP}}_1(\pi)$. We will show

$$\liminf_{k\to\infty} \frac{w_2(\langle v_j\rangle_{j\leq k})}{k} - l \leq r - \epsilon, \tag{11}$$

from which the required contradiction follows.

For any $k \geq 0$, the cycle decomposition of $\langle v_j\rangle_{j\leq k}$ tells us that apart from a small sub-path, π', of length at most n (the number of states in G), the prefix $\langle v_j\rangle_{j\leq k}$ can be decomposed into simple cycles $\chi_1, \ldots, \chi_t$ such that $w_i(\langle v_j\rangle_{j\leq k}) = w_i(\pi') + \sum_{j=1}^{t} w_i(\chi_j)$ for $i = 1, 2$. If W is the maximum weight occurring in G, then from Eq. (*) we have:

$$\begin{aligned} w_2(\langle v_j \rangle_{j \le k}) &\le nW + \sum_{j=1}^{t} w_2(\chi_j) \\ &\le nW + (r - \epsilon) \sum_{j=1}^{t} |\chi_j| + \sum_{j=1}^{t} w_1(\chi_j) \\ &\le nW + k(r - \epsilon) + w_1(\langle v_j \rangle_{j \le k}). \end{aligned}$$

Now, it follows from the definition of the limit inferior that for any $\epsilon' > 0$ and any $K > 0$ there exists $k > K$ such that $w_1(\langle v_j \rangle_{j \le k}) \le k(l + \epsilon')$. Thus for any $\epsilon' > 0$ and $K' > 0$, there exists $k > \max\{K', nW/\epsilon'\}$ such that

$$\frac{w_2(\langle v_j \rangle_{j \le k})}{k} \le \frac{nW}{k} + (r - \epsilon) + (l + \epsilon') < (l + r - \epsilon) + 2\epsilon'.$$

Equation (11) then follows from the definition of limit inferior.

The above implies that Adam can, by repeating χ infinitely often, achieve a regret value of at least r against strategy σ of Eve. As this can be done by him playing a regular word, the result follows. □

We now proceed with the proof of the theorem. The argument is presented for mean-payoff ($\underline{\mathsf{MP}}$) but minimal changes are required for the other payoff functions. For simplicity, we use the non-strict threshold for the emptiness problems. However, the result from [9] is independent of this. Further, the exact same argument presented here works for both cases. Thus, if suffices to show the result follows for $\ge$.

Proof (of Theorem 4) We will "guess" a strategy for Eve which uses memory at most m and verify (in polynomial time w.r.t. m and the size of Γ) that it ensures a regret value of strictly less than r.

Let $\mathcal{A}$ be the mean-payoff ($\underline{\mathsf{MP}}$) automaton constructed as the synchronous product of Γ and Γ_σ. The new weight function maps a transition to the difference of the values of the weight functions of the two original automata. We claim that the language of $\mathcal{A}$ is empty (for accepting threshold $\ge r$) if and only if $\mathbf{reg}^{\sigma}_{\Sigma_{\exists}^{m}, \mathfrak{W}_{\forall}}(\Gamma) < r$. Indeed, there is a bijective map from every run of $\mathcal{A}$ to a pair of plays π, π' in Γ such that both π and π' are consistent with the same word strategy of Adam and π is consistent with σ. It will be clear that $\mathcal{A}$ has size at most $m|\Gamma|$. As emptiness of a weighted automaton $\mathcal{A}$ can be decided in $O(|\mathcal{A}|^2)$ time [9], the result will follow.

We now show that if the language of $\mathcal{A}$ is not empty then Adam can ensure a regret value of at least r against σ in Γ and that, conversely, if Adam has a spoiling strategy against σ in Γ then that implies the language of $\mathcal{A}$ is not empty.

Let ρ_x be a run of $\mathcal{A}$ on x. From the definition of $\mathcal{A}$ we get that $\underline{\mathsf{MP}}(\rho_x) = \liminf_{i \to \infty} \frac{1}{i} \sum_{j=0}^{i} (a_j - b_j)$ where $\alpha_x = \langle a_i \rangle_{i \ge 0}$ and $\beta_x = \langle b_i \rangle_{i \ge 0}$ are the infinite sequences of weights assigned to the transitions of ρ by the weight functions of Γ and Γ_σ respectively. It is known that if a mean-payoff automaton accepts a word y then it must accept an ultimately periodic word y', thus we can assume that x is ultimately periodic (see, e.g. [9]). Furthermore, we can also assume the run of the automaton on x is ultimately periodic. Recall that for ultimately periodic runs we have that $\underline{\mathsf{MP}}(\rho_x) = \overline{\mathsf{MP}}(\rho_x)$. We get the following

$$\begin{aligned}
\underline{\mathsf{MP}}(\rho_x) &= \limsup_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i}(a_j - b_j) \\
&\le \limsup_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} a_j + \limsup_{i\to\infty} \frac{-1}{i}\sum_{j=0}^{i} b_j && \text{sub-additivity of } \limsup \\
&\le \limsup_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} a_j - \liminf_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} b_j \\
&\le \liminf_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} a_j - \liminf_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} b_j && \text{ultimate periodicity.}
\end{aligned}$$

Thus, as x and ρ_x can be be mapped to a strategy of Adam in Γ which ensures regret of at least r against σ, the claim follows.

For the other direction, assume Adam has a word strategy τ in Γ which ensures a regret of at least r against σ. From Lemma 15 it follows that τ and the run ρ of Γ with value $\Gamma(\tau)$ can be assumed to be ultimately periodic w.l.o.g.. Denote by ρ_σ and w_σ the run of Γ_σ on τ and the weight function of Γ_σ respectively. We then get that

$$\begin{aligned}
&\liminf_{i\to\infty} \frac{1}{i} w_\sigma(\rho_\sigma) - \liminf_{i\to\infty} \frac{1}{i} w(\rho) \\
&\quad = \liminf_{i\to\infty} \frac{1}{i} w_\sigma(\rho_\sigma) + \limsup_{i\to\infty} \frac{-1}{i} w(\rho) \\
&\quad = \liminf_{i\to\infty} \frac{1}{i} w_\sigma(\rho_\sigma) + \liminf_{i\to\infty} \frac{-1}{i} w(\rho) && \text{ultimate periodicity} \\
&\quad \le \underline{\mathsf{MP}}(\psi_\tau) && \text{super-additivity of } \liminf,
\end{aligned}$$

where ψ_τ is the corresponding run of $\mathcal{A}$ for τ and ρ. Hence, $\mathcal{A}$ has at least one word in its language. □

We provide a matching lower bound. The proof is an adaptation of the NP-hardness proof from [1].

Theorem 5 *For $r \in \mathbb{Q}$, weighted automaton Γ and payoff function* Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, *or* $\overline{\mathsf{MP}}$, *determining whether* $\mathbf{Reg}_{\Sigma^1_\exists,\mathfrak{W}_\forall}(\Gamma) \lhd r$, *for* $\lhd \in \{<,\le\}$, *is* NP-*hard.*

Proof We give a reduction from the SAT problem, i.e. satisfiability of a CNF formula. The construction presented is based on a proof in [1]. The idea is simple: given boolean formula Φ in CNF we construct a weighted automaton Γ_Φ such that Eve can ensure regret value of 0 with a positional strategy in Γ_Φ if and only if Φ is satisfiable.

Let us now fix a boolean formula Φ in CNF with n clauses and m boolean variables $x_1, \ldots, x_m$. The weighted automaton $\Gamma_\Phi = (Q, q_I, A, \Delta, w)$ has alphabet $A = \{bail, \#\} \cup \{i \mid 1 \le i \le n\}$. Γ_Φ includes an initial gadget such as the one depicted in Fig. 9. Recall that this gadget forces Eve to play into the right sub-arena. As the left sub-arena of Γ_Φ we attach the gadget depicted in Fig. 12. All transitions shown have weight 1 and all missing transitions in order for Γ_Φ to be complete lead to a state $\bot_0$ with a self-loop on every symbol from A with weight 0. Intuitively, as Eve must go to the right sub-arena then all alternative plays in the left sub-arena correspond to either Adam choosing a clause i and spelling $i\#i$

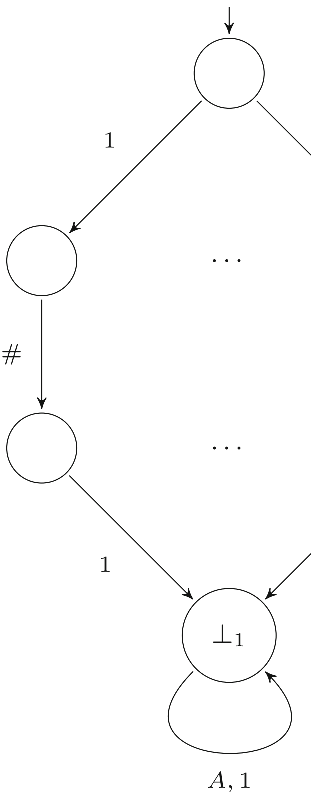

Fig. 12 Clause choosing gadget for the SAT reduction. There are as many paths from *top* to *bottom* ($\bot_1$) as there are clauses (n)

to reach $\bot_1$ or reaching $\bot_0$ by playing any other sequence of symbols. The right sub-arena of the automaton is as shown in Fig. 13, where all transitions shown have weight 1 and all missing transitions go to $\bot_0$ again. Here, from q_0 we have transitions to state x_j with symbol i if the i-th clause contains variable x_j. For every state x_j we have transitions to j_{true} and j_{false} with symbol #. The idea is to allow Eve to choose the truth value of x_j. Finally, every state j_{true} (or j_{false}) has a transition to $\bot_1$ with symbol i if the literal x_j (resp. $\neg x_j$) appears in the i-th clause.

The argument to show that Eve can ensure regret of 0 if and only if Φ is satisfiable is straightforward. Assume the formula is indeed satisfiable. Assume, also, that Adam chooses $1 \leq i \leq n$ and spells $i\#i$. Since Φ is satisfiable there is a choice of values for $x_1, \ldots, x_m$ such that for each clause there must be at least one literal in the i-th clause which makes the clause true. Eve transitions, in the right sub-arena from q_0 to the corresponding value and when Adam plays # she chooses the correct truth value for the variable. Thus, the play reaches $\bot_1$ and, as $W = 1$ in Φ it follows that her regret is 0. If Adam does not play as assumed then we know all plays in Γ_Φ reach $\bot_0$ and again her regret is 0. Note that this strategy can be realized with a positional strategy by assigning to each x_j the choice of truth value and choosing from q_0 any valid transition for all $1 \leq i \leq n$.

Conversely, if Φ is not satisfiable then for every valuation of variables $x_1, \ldots, x_m$ there is at least one clause which is not true. Given any positional strategy of Eve in Γ_Φ we can

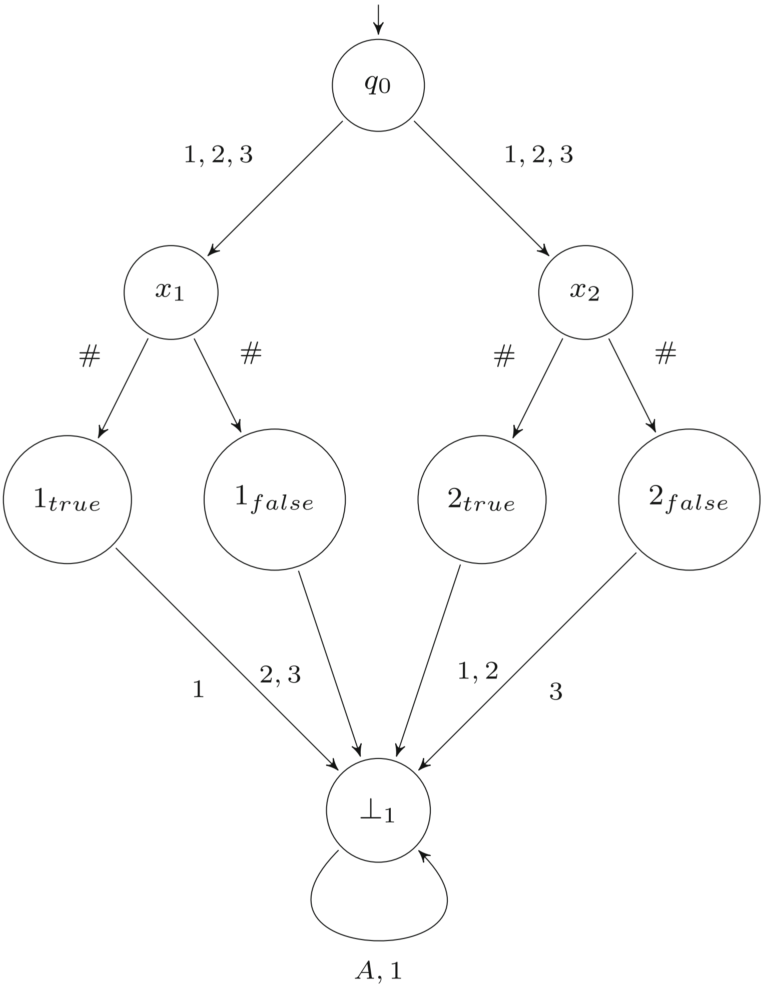

Fig. 13 Value choosing gadget for the SAT reduction. Depicted is the configuration for $(x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$

extract the corresponding valuation of the boolean variables. Now Adam chooses $1 \leq i \leq n$ such that the i-th clause is not satisfied by the assignment. The play will therefore end in $\perp_0$ while an alternative play in the left sub-arena will reach $\perp_1$. Hence the regret of Eve in the game is 1.

To complete the proof we note that the above analysis is the same for payoff functions Inf, LimInf, LimSup, and MP. For Sup it suffices to change all the weights in the gadgets from 1 to 0.

We observe that, once more, we can adapt the values of the loops in the sinks $\perp_1$ and $\perp_0$ to get the same result for the non-strict regret threshold problem. □

5.6 Relation to other works

Let us first extend the definitions of *approximation, embodiment* and *refinement* from [1] to the setting of ω-words. Consider two weighted automata $\mathcal{A} = (Q_{\mathcal{A}}, q_I, A, \Delta_{\mathcal{A}}, w_{\mathcal{A}})$ and $\mathcal{B} = (Q_{\mathcal{B}}, q_I, A, \Delta_{\mathcal{B}}, w_{\mathcal{B}})$ and let $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a *metric*.[4] We say $\mathcal{B}$ *(strictly) α-approximates $\mathcal{A}$ (with respect to d)* if $d(\mathcal{B}(w), \mathcal{A}(w)) \leq \alpha$ (resp. $d(\mathcal{B}(w), \mathcal{A}(w)) < \alpha$) for all words $w \in A^{\omega}$. We say $\mathcal{B}$ *embodies* $\mathcal{A}$ if $Q_{\mathcal{A}} \subseteq Q_{\mathcal{B}}$, $\Delta_{\mathcal{A}} \subseteq \Delta_{\mathcal{B}}$ and $w_{\mathcal{A}}$ agrees with $w_{\mathcal{B}}$ on $\Delta_{\mathcal{A}}$. For an automaton $\mathcal{A} = (Q, q_I, A, \Delta, w)$ and an integer $k \geq 0$, the k-refinement of $\mathcal{A}$ is the automaton obtained by refining the state space of $\mathcal{A}$ using k boolean variables. Intuitively, this corresponds to having 2^k copies of every state, with each copy of p transitioning to all copies of q with a if $(p, a, q) \in \Delta$. The automaton $\mathcal{A}$ is said to be *(strictly) (α, k)-determinizable by pruning* (DBP, for short) if the k-refinement of $\mathcal{A}$ embodies a deterministic automaton which (strictly) α-approximates $\mathcal{A}$. The next result follows directly from the above definitions.

Proposition 2 *For $\alpha \in \mathbb{Q}$, $k \in \mathbb{N}$, a weighted automaton Γ is (strictly) (α, k)-DBP (w.r.t. the difference metric) if and only if $\mathbf{Reg}_{\Sigma_{\exists}^{2^k}, \mathfrak{W}_{\forall}}(\Gamma) \leq \alpha$ (resp. $\mathbf{Reg}_{\Sigma_{\exists}^{2^k}, \mathfrak{W}_{\forall}}(\Gamma) < \alpha$).*

In [19] the authors define *good for games automata*. Their definition is based on a game which is played on an ω-automaton by Spoiler and Simulator. We propose the following generalization of the notion of good for games automata for weighted automata. A weighted automaton $\mathcal{A}$ is *(strictly) α-good for games* if Simulator, against any word $w \in A^{\omega}$ spelled by Spoiler, can resolve non-determinism in $\mathcal{A}$ so that the resulting run has value v and $d(v, \mathcal{A}(w)) \leq \alpha$ (resp. $d(v, \mathcal{A}(w)) < \alpha$), for some metric d. We summarize the relationship that follows from the definition in the following result:

Proposition 3 *For $\alpha \in \mathbb{Q}$, a weighted automaton Γ is (strictly) α-good for games (w.r.t. the difference metric) if and only if $\mathbf{Reg}_{\mathfrak{S}_{\exists}, \mathfrak{W}_{\forall}}(\Gamma) \leq \alpha$ (resp. $\mathbf{Reg}_{\mathfrak{S}_{\exists}, \mathfrak{W}_{\forall}}(\Gamma) < \alpha$).*

6 Discussion

In this work we have considered the regret threshold problem in quantitative games. We have studied three variants which corresponds to different assumptions regarding the behavior of Adam. Our definition of regret is based on the difference measure: Eve attempts to minimize the difference between the value she obtains by playing the game, and the value she could have obtained if she had known the strategy of Adam in advance. In [1] the ratio measure was used instead. We believe some of the results obtained presently can be extended to arbitrary metrics (as in, *e.g.*, [5]). In particular, all hardness statements should hold. We give more precise claims for the ratio measure below.

6.1 For **Inf, Sup, LimInf,** *and* **LimSup**

We have already observed that upper bounds for the regret threshold problem follow directly from our results if regret is defined using ratio (see Remark 1). Furthermore, all hardness results presented here can also be adapted to obtain the same result for ratio. Indeed, the same constructions and gadgets can be used. These, together with correctly chosen regret threshold value r and modified edge weights and inequalities (such as the ones given to prove,

[4] The metric used in [1] is the ratio measure.

for instance, Lemma 8) are sufficient to show the same results hold for regret defined with ratio.

6.2 For MP

All hardness results also hold for regret defined with ratio. As with the other payoff functions, minimal modifications are needed for the proofs given in this work to imply the result for the alternative definition of regret. Regarding the algorithms, we have solved the regret threshold problem for the first two variants. In the third variant, we have considered a restricted version of the game (Theorem 4) and given an algorithm for it by reducing it to an the emptiness problem for mean-payoff automata. We claim the corresponding problems are in the same complexity classes, respectively, when regret is defined with ratio. For the first two, the proofs are almost identical to the ones we have give in the present work for the difference measure. For the third problem, Lemma 15 must be restated for ratio, yet the proof requires minimal modifications to work in that case. Finally, the argument used to prove Theorem 4 requires the reduction to mean-payoff automata be replaced by a reduction to ratio automata. However, all the properties from mean-payoff automata which were used in the proof, are also true for ratio automata (*e.g.* ultimately periodic words being accepted if an arbitrary word is accepted). The latter follow from results regarding ratio games in [4].

Acknowledgements We thank Udi Boker for his comments on how to determinize LimSup automata.

References

1. Aminof, B., Kupferman, O., Lampert, R.: Reasoning about online algorithms with weighted automata. ACM Transactions on Algorithms (2010)
2. Aminof, B., Rubin, S.: First cycle games. In: *SR*, pp. 83–90 (2014)
3. Bell, D.E.: Regret in decision making under uncertainty. Oper. Res. **30**(5), 961–981 (1982)
4. Bloem, R., Chatterjee, K., Greimel, K., Henzinger, T.A., Hofferek, G., Jobstmann, B., Könighofer, B., Könighofer, R.: Synthesizing robust systems. Acta Inf. **51**(3–4), 193–220 (2014)
5. Boker, U., Henzinger, TA.: Exact and approximate determinization of discounted-sum automata. LMCS **10**(1) (2014). doi:10.2168/LMCS-10(1:10)2014
6. Brim, L., Chaloupka, J., Doyen, L., Gentilini, R., Raskin, J.-F.: Faster algorithms for mean-payoff games. Form. Methods Syst. Des. **38**(2), 97–118 (2011)
7. Chakrabarti, A., de Alfaro, L., Henzinger, T.A., Stoelinga, M.: Resource interfaces. In: EMSOFT, volume 2855 of LNCS, pp. 117–133. Springer (2003)
8. Chatterjee, K., Doyen, L., Filiot, E., Raskin, JF.: Doomsday equilibria for omega-regular games. In: VMCAI, vol. 8318, pp. 78–97. Springer (2014)
9. Chatterjee, K., Doyen, L., Henzinger, TA.: Quantitative languages. ACM Transactions on Computational Logic **11**(4), 1–38 (2010)
10. Chatterjee, K., Doyen, L., Henzinger, T.A., Raskin, J.-F.: Generalized mean-payoff and energy games. In: FSTTCS, pp. 505–516 (2010)
11. Damm, W., Finkbeiner, B.: Does it pay to extend the perimeter of a world model? In: FM, volume 6664 of LNCS, pp. 12–26. Springer (2011)
12. Degorre, A., Doyen, L., Gentilini, R., Raskin, J.-F., Toruńczyk, S.: Energy and mean-payoff games with imperfect information. In: CSL, pp. 260–274 (2010)
13. Dziembowski, S., Jurdziński, M., Walukiewicz, I.: How much memory is needed to win infinite games? In: LICS of IEEE computer society, pp. 99–110 (1997)
14. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. Int. J. Game Theory **8**, 109–113 (1979)
15. Eilam-Tzoreff, T.: The disjoint shortest paths problem. Discrete Appl. Math. **85**(2), 113–138 (1998)
16. Filiot, E., Le Gall, T., Raskin, J.-F.: Iterated regret minimization in game graphs. In: MFCS, volume 6281 of LNCS, pp. 342–354. Springer (2010)

17. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York (1979)
18. Halpern, J.Y., Pass, R.: Iterated regret minimization: a new solution concept. Games Econ. Behav. **74**(1), 184–207 (2012)
19. Henzinger, T.A., Piterman, N.: Solving games without determinization. In CSL, pp. 395–410 (2006)
20. Hunter, P., Pérez G.A., Raskin, J.-F.: Mean-payoff games with partial-observation-(extended abstract). In: Reachability Problems, pp. 163–175 (2014)
21. Jurdziński, M.: Deciding the winner in parity games is in $\mathsf{UP} \cup \mathsf{coUP}$. IPL **68**(3), 119–124 (1998)
22. Jurdzinski, M., Sproston, J., Laroussinie, F.: Model checking probabilistic timed automata with one or two clocks. LMCS **4**(3) (2008). doi:10.2168/LMCS-4(3:12)2008
23. Papadimitriou, C.H., Yannakakis, M.: Shortest paths without a map. TCS **84**(1), 127–150 (1991)
24. Piterman, N.: From nondeterministic Büchi and Streett automata to deterministic parity automata. LMCS **3**(3) (2007). doi:10.2168/LMCS-3(3:5)2007
25. Piterman, N., Pnueli, A.: Faster solutions of Rabin and Streett games. In: LICS, pp. 275–284 (2006)
26. Pnueli, A., Rosner, R.: On the synthesis of a reactive module. In: POPL, pp. 179–190. ACM Press (1989)
27. Wen, M., Ehlers, R., Topcu, U.: Correct-by-synthesis reinforcement learning with temporal logic constraints. In: IEEE of IROS, pp. 4983–4990 (2015)
28. Zinkevich, M., Johanson, M., Bowling, M., Piccione, C.: Regret minimization in games with incomplete information. In: NIPS, pp. 905–912 (2008)
29. Zwick, U., Paterson, M.: The complexity of mean payoff games on graphs. TCS **158**(1), 343–359 (1996)