Abstract

Iterated regret minimization has been introduced recently by J.Y. Halpern and R. Pass in classical strategic games. For many games of interest, this new solution concept provides solutions that are judged more reasonable than solutions offered by traditional game concepts—such as Nash equilibrium. Although computing iterated regret on explicit matrix game is conceptually and computationally easy, nothing is known about computing the iterated regret on games whose matrices are defined implicitly using game tree, game DAG or, more generally, game graphs. In this paper, we investigate iterated regret minimization for infinite duration two-player quantitative non-zero sum games played on graphs.

We consider reachability objectives that are not necessarily antagonist. Edges are weighted by integers—one for each player—and the payoffs are defined by the sum of the weights along the paths. Depending on the class of graphs, we give either polynomial or pseudo-polynomial time algorithms to compute a strategy that minimizes the regret for a fixed player. We finally give algorithms to compute the strategies of the two players that minimize the iterated regret for trees, and for graphs with strictly positive weights only.

1 Introduction

The analysis of complex interactive systems like embedded systems or distributed systems is a major challenge of computer-aided verification. Zero-sum games on graphs provide a good framework to model interactions between a component and an environment as they are strictly competitive. However, in the context of modern interactive systems, several components may interact and be controlled independently. Non-zero sum games on graphs are more accurate to model such systems, as the objectives are not necessarily antagonist. There are initial results in this area but a large number of questions are open. In this paper, we adapt to game graphs a new solution concept of non-zero sum games initially defined for strategic games.

In [7], J.Y. Halpern and R. Pass defined the notion of iterated regret minimization. This solution concept assumes that instead of trying to minimize what she has to pay, each player tries to minimize her regret. The regret is informally defined as the difference between what a player actually pays and what she could have paid if she knew the strategy chosen by the other player. More formally, if \( u_1(\lambda_1, \lambda_2) \) represents what Player 1 pays when the pair of strategies \((\lambda_1, \lambda_2)\) is played, \( \text{reg}_1(\lambda_1, \lambda_2) = u_1(\lambda_1, \lambda_2) - \min_{\lambda_1'} u_1(\lambda_1', \lambda_2) \).

Let us illustrate this on an example. Consider the strategic game defined by the matrix of figure 1. In the game underlying this matrix, Player 1 has two strategies \( A_1 \) and \( B_1 \) and Player 2 has two strategies \( A_2 \) and \( B_2 \). The two players choose a strategy at the same time and the pairs of strategies define what the two players have to pay. The regret of playing \( A_1 \) for Player 1 if Player 2 plays \( A_2 \) is equal to 1 because \( u_1(A_1, A_2) = 2 \) when \( u_1(B_1, A_2) = 1 \). Knowing that Player 2 plays \( A_2 \), Player 1 should have played \( B_1 \).

As Players have to choose strategies before knowing how the adversary will play, we associate a regret with each strategy as follows. The regret of a strategy \( \lambda_1 \) of Player 1 is: \( \text{reg}_1(\lambda_1) = \max_{\lambda_2} \text{reg}_1(\lambda_1, \lambda_2) \). In the example, the regret attached to strategy \( A_1 \) is equal to 1.

\footnote{We could have considered rewards instead of penalties, everything is symmetrical.}
1, because when Player 2 plays $A_2$, Player 1 regret is 1, and when Player 2 plays $B_2$ her regret is 0. A rational player should minimize her regret. The regret for Player 1 is thus defined as $\text{reg}_1 = \min_{\lambda_1} \text{reg}_1'(\lambda_1)$, summarizing, we get $\text{reg}_1 = \min_{\lambda_1} \max_{\lambda_2} (u_1(\lambda_1, \lambda_2) - \min_{\lambda_1'} u_1(\lambda_1', \lambda_2))$. A symmetrical definition can be given for Player 2’s regret.

Let us come back to the example. The regret attached to strategy $B_1$ is equal to 1. So the two strategies of Player 1 are equivalent w.r.t. regret minimization. On the other hand, for Player 2, the regret of $A_2$ equals 0, and the regret of $B_2$ equals 3. So, if Player 1 makes the hypothesis that Player 2 is trying to minimize her regret, then she must conclude that Player 2 will play $A_2$. Knowing that, Player 1 recomputes her regret for each action, and in this case, the regret of action $A_1$ is 1 while the regret of $B_1$ is 0. So rational players minimizing their regret should end up playing the pairs $(B_1, A_2)$ in this game.

Reasoning on rationality is formalized by Halpern and Pass by introducing a delete operator that erases strictly dominated strategies. This operator takes sets of strategies $(\Lambda_1, \Lambda_2)$ for each player and returns $D(\Lambda_1, \Lambda_2) = (\Lambda'_1, \Lambda'_2)$ the strategies that minimize regret. Then $D(\Lambda_1', \Lambda_2')$ returns the strategies that minimize regret under the hypothesis that adversaries minimize their regret i.e., choose their strategies in $\Lambda'_1$ and $\Lambda'_2$ respectively. In the case of finite matrix games, this operator is monotone and converges on the strategies that minimize regrets for the two players making the assumption of rationality of the other player.

In this paper, we consider games where the matrix is not given explicitly but defined implicitly by a game graph. More precisely, we consider graphs where vertices are partitioned into vertices that belong to Player 1 and vertices that belong to Player 2. Each edge is annotated by a penalty for Player 1 and one for Player 2. Additionally, there are two designated sets of vertices, one that Player 1 tries to reach and the other one that Player 2 tries to reach. The game starts in the initial vertex of the graph and is played for an infinite number of rounds as follows.

|   | $A_2$ | $B_2$ |
|---|---|---|
| $A_1$ | (2, 1) | (3, 4) |
| $B_1$ | (1, 2) | (4, 3) |

Figure 1: A strategic game with explicit penalty matrix.

In each round, the Player who owns the vertex on which the pebble is placed moves the the pebble to an adjacent vertex using an edge of the graph, and a new round starts. The infinite plays generate an infinite sequence of vertices and the amount that the players have to pay are computed as follows. Player 1 pays $+\infty$ if the sequence does not reach the target set assigned to Player 1, otherwise she pays the sum of edge costs assigned to her on the prefix up to the first visit to her target set. The amount to pay for Player 2 is defined symmetrically. Strategies in such games are functions from the set of histories of plays (sequences of visited vertices) to edges (choice of moves for the pebble).

Let us consider the game graph of Fig. 1. This is a formalization of the so-called Centipede game [9] in our game graphs. We have considered a 5-round variant here, this game can be generalized to any number of rounds. Initially, the pebble is on vertex $A$. Player 1 owns the circle vertices and Player 2 owns the square vertices. The target objective for the two players is the same: they both want to reach vertex $S$. At each round, one of the players has to choose either to stop the game and reach the target, or to let the game continue for at least an additional round. The penalties attached to edges are given as pairs of integers (the first for Player 1 and the second for Player 2). Strategies here are as follows. For each circle vertex, Player 1 must decide either to continue or to go to the target $S$, and symmetrically for Player 2. It can be shown (and computed by our algorithms) that the strategy of Player 1 that survives iterated regret minimization is the strategy that stops the game only in position $E$ and the strategy for Player 2 is the strategy that continue the game to vertex $D$. This pair of strategy has a penalty of $(1, 3)$. This is an interesting and rather nice joint behavior of the two players in comparison of what Nash equilibrium is predicting for this example. Indeed, the only Nash equilibrium in that game is the pair of strategies where the two players decide to stop directly the game and so they have to pay $(5, 7)$. This is a 5-round example but the difference between the penalty of the Nash equilibrium and the iterated regret grows as the number of rounds increases.

---

2A Nash equilibrium is a pair of strategies where no player has an incentive to change her strategy if the other player keeps playing her strategy.
Contributions: We investigate iterated regret minimization for infinite duration two-player quantitative non-zero sum games played on graphs. We focus on reachability objectives that are not necessarily antagonistic.

We first consider target-weighted arenas, where the payoff function is defined for each state of the objectives. We give a PTIME algorithm to compute the regret by reduction to a min-max game.

We then consider edge-weighted arenas. Each edge is labeled by a pair of integers – one for each player – , and the payoffs are defined by the sum of the weights along the path until the first visit to an objective. We give a pseudo-PTIME algorithm to compute the regret in an edge-weighted arena, by reduction to a target-weighted arena.

We also study the problem of iterated regret minimization. We provide a delete operator that removes strictly dominated strategies. We show how to compute the effect of iterating this operator on tree arenas and strictly positive edge-weighted arenas. In the first case, we provide a quadratic time algorithm and in the second case, a pseudo-exponential time algorithm.

Related works: Several notions of equilibria have been proposed in the literature for reasoning on 2-players non-zero-sum games, for instance Nash equilibrium, sequential equilibrium, perfect equilibrium - see [8] for an overview. Those equilibria formalize notions of rational behavior by defining optimality criteria for pairs of strategies. As we have seen in the Centipede game example [9], or as it can be shown for other examples like the Traveller’s dilemma [1], Nash equilibria sometimes suggest pairs of strategies that are rejected by common sense. Regret minimization is an alternative solution concept that sometimes proposes more intuitive solutions and requires more cooperation between players. Recently, non-zero sum games played on graphs have attracted a lot of attention. There have been several papers that study Nash equilibria or particular classes of Nash equilibria [6][3][4][5].

Proofs that are sketched or omitted in the paper are given in Appendix.

2 Weighted Games and Regret

Given a cartesian product \(A \times B\) of two sets, we denote by \(\text{proj}_i\) the \(i\)-th projection, \(i = 1, 2\). It is naturally extended to sequence of elements of \(A \times B\) by \(\text{proj}_i(e_1 \ldots e_n) = \text{proj}_i(e_1) \ldots \text{proj}_i(e_n)\). For all \(k \in \mathbb{N}\), we let \(\{k\} = \{0, \ldots, k\}\).

Reachability Games: Turn-based two-player games are played on arena games by two players. A (finite) game arena is a tuple \(G = (S = S_1 \sqcup S_2, s_0, T)\) where \(S_1, S_2\) are finite disjoint sets of player positions (\(S_1\) for Player 1 and \(S_2\) for Player 2), \(s_0 \in S_1\) is the initial position, and \(T \subseteq S \times S\) is the transition relation. A finite play on \(G\) of length \(n\) is a finite word \(\pi = \pi_0\pi_1 \ldots \pi_n \in S^*\) such that \(\pi_0 = s_0\) and for all \(i = 0, \ldots, n - 1\), \((\pi_i, \pi_{i+1}) \in T\). Infinite plays are defined similarly. We denote by \(P_f(G)\) (resp. \(P_{\infty}(G)\)) the set of finite (resp. infinite) plays on \(G\), and we let \(P(G) = P_f(G) \cup P_{\infty}(G)\). For any node \(s \in S\), we denote by \((G, s)\) the arena \(G\) where the initial position is \(s\).

Let \(i \in \{1, 2\}\). We let \(\neg i = 1\) if \(i = 2\) and \(\neg i = 2\) if \(i = 1\). A strategy \(\lambda_i : P_f(G) \to S \cup \{\bot\}\) for Player \(i\) is a mapping that maps any finite play \(\pi\) whose last position – denoted last(\(\pi\)) – is in \(S_i\) to \(\bot\) if there is no outgoing edge from last(\(\pi\)), and to a position \(s\) such that (last(\(\pi\)), \(s\)) \(\in T\) otherwise. The set of strategies of Player \(i\) in \(G\) is denoted by \(\Lambda_i(G)\). Given a strategy \(\lambda_i \in \Lambda_i(G)\), the outcome \(\text{Out}^G(\lambda_i, \lambda_{\neg i})\) is a play \(\pi = \pi_0 \ldots \pi_n\) such that (i) \(\pi_0 = s_0\), (ii) if \(\pi\) is finite, then there is not outgoing edge from last(\(\pi\)), and (iii) for all \(0 \leq j \leq |\pi|\) and all \(\kappa = 1, 2\), if \(\pi_j \in S_\kappa\), then \(\pi_{j+1} = \lambda_\kappa(\pi_0 \ldots \pi_j)\). We also define \(\text{Out}^G(\lambda_i) = \{\text{Out}^G(\lambda_i, \lambda_{\neg i}) \mid \lambda_{\neg i} \in \Lambda_{\neg i}(G)\}\).

A strategy \(\lambda_i\) is memoryless if for all play \(\pi \in P_f(G)\), \(\lambda_i(\pi)\) only depends on last(\(\pi\)). Thus \(\lambda_i\) can be seen as a function \(S_i \mapsto S \cup \{\bot\}\). It is finite-memory if \(\lambda_i(\pi)\) only

Figure 2: Centipede Game

\[A \quad 0/0 \quad B \quad 6/4 \quad S \quad 3/5 \quad C \quad 0/0 \quad D \quad 5/7 \quad E \quad 0/2 \quad \]

\[0/0 \quad 1/3 \quad 4/2 \quad 0/0 \]
depends on last(π) and on some state of a finite state set. We refer the reader to [5] for formal definitions.

A reachability winning condition (rwc for short) for Player \(i\) is given by a subset of positions \(C_i \subseteq S\) called the target set. A play \(π \in \mathcal{P}(G)\) is winning for Player \(i\) if some position of \(π\) is in \(C_i\). A strategy \(λ_i\) for Player \(i\) is winning if all the plays of \(\text{Out}^G(λ_i)\) are winning. In this paper, we often consider two target sets \(C_1, C_2\) for Player 1 and 2 respectively. We write \((S_1, S_2, s_0, T, C_1, C_2)\) to denote the game arena \(G\) extended with those target sets. Finally, let \(λ_i \in Λ_i(G)\) be a winning strategy for Player \(i\) and \(λ_i \in Λ_i(G)\). Let \(π_0, π_1, \ldots \in \mathcal{P}(G)\) be the outcome of \((λ_i, λ_i)\). The outcome of \((λ_i, λ_i)\) up to \(C_i\) is defined by \(\text{Out}^G(λ_i, λ_i) = π_0, π_1, \ldots \pi_n\) such that \(n = \min\{j \mid π_j ∈ C_i\}\). We also extend this notation to sets of plays \(\text{Out}^G(λ_i)\) naturally.

**Weighted Games** We add weights on edges of arenas and include the target sets. A (finite) **weighted game arena** is a tuple \(G = (S = S_1 ∪ S_2, s_0, T, μ_1, μ_2, C_1, C_2)\) where \((S, s_0, T)\) is a game arena, for all \(i = 1, 2, μ_i : T → \mathbb{N}\) is a weigth function for Player \(i\) and \(C_i\) its target set. We let \(M_i^G\) be the maximal weight of Player \(i\), i.e. \(M_i^G = \max_{e ∈ T} μ_i(e)\) and \(M_i^G = \max(M_i^G, M_2^G)\).

\(G\) is a target-weighted arena (TWA for short) if only the edges leading to a target node are weighted by strictly positive integers, and any two edges leading to the same node carry the same weight. Formally, for all \((s, s') \in T\), if \(s' \notin C_i\), then \(μ_i(s, s') = 0\), otherwise for all \((s', s'') \in T\), \(μ_i(s', s') = μ_i(s'', s')\). Thus for target-weighted arenas, we assume in the sequel that the weight functions map \(C_i\) to \(\mathbb{N}\).

Let \(π = π_0π_1 \ldots π_n\) be a finite play in \(G\). We extend the weight functions to finite plays, so that for all \(i = 1, 2\), \(μ_i(π) = \sum_{j=0}^{n-1} μ_i(π_j, π_{j+1})\). The **utility** \(u_i^G(π)\) of \(π\) (for Player \(i\)) is \(+∞\) if \(π\) is not winning for Player \(i\), and the sum of the weights occurring along the edges defined by \(π\) until the first visit to a target position otherwise. Formally:

\[
u_i^G(π) = \begin{cases} +∞ & \text{if } π \text{ is not winning for Player } i \\ \sum_{j=0}^{\min(k \mid π_k ∈ C_i)-1} μ_i(π_j, π_{j+1}) & \text{otherwise} \end{cases}
\]

We extend this notion to the utility of two strategies \(λ_1, λ_2\) of Player 1 and 2 respectively:

\[
u_i^G(λ_1, λ_2) = u_i^G(\text{Out}^G(λ_1, λ_2))
\]

Let \(i \in \{1, 2\}\). Given a strategy \(λ_i \in Λ_i(G)\), the best response of Player \(−i\) to \(λ_i\), denoted by \(br_i^G(λ_i)\), is the least utility Player \(−i\) can achieve against \(λ_i\). Formally:

\[
br_i^G(λ_i) = \min_{λ_{−i} ∈ Λ_{−i}(G)} u_i^G(λ_i, λ_{−i})
\]

**Regret** Let \(i \in \{1, 2\}\) and let \(λ_1, λ_2\) be two strategies of Player 1 and 2 respectively. The **regret** of Player \(i\) is the difference between the utility Player \(i\) achieves and the best response to the strategy of Player \(−i\). Formally:

\[
\text{reg}_i^G(λ_i, λ_{−i}) = u_i^G(λ_i, λ_{−i}) - \text{br}_i^G(λ_{−i})
\]

Note that \(\text{reg}_i^G(λ_i, λ_{−i}) ≥ 0\), since \(\text{br}_i^G(λ_{−i}) ≤ u_i^G(λ_i, λ_{−i})\). The regret of a strategy \(λ_i\) for Player \(i\) is the maximal regret she gets for all strategies of Player \(−i\):

\[
\text{reg}_i^G(λ_i) = \max_{λ_{−i} ∈ Λ_{−i}(G)} \text{reg}_i^G(λ_i, λ_{−i})
\]

Finally, the regret of Player \(i\) in \(G\) is the minimal regret she can achieve:

\[
\text{reg}_i^G = \min_{λ_i ∈ Λ_i(G)} \text{reg}_i^G(λ_i)
\]

We let \(+∞ = (+∞) = +∞\).

**Proposition 1.** For all \(i = 1, 2\), \(\text{reg}_i^G < +∞\) iff Player \(i\) has a winning strategy.

**Proof.** If Player \(i\) has no winning strategy, then for all \(λ_i \in Λ_i(G)\), there is \(λ_{−i} \in Λ_{−i}(G)\) s.t. \(u_i^G(λ_i, λ_{−i}) = +∞\). Thus \(\text{reg}_i^G(λ_i, λ_{−i}) = +∞\). Therefore \(\text{reg}_i^G = +∞\).

If Player \(i\) has a winning strategy \(λ_i\), then for all \(λ_{−i} \in Λ_{−i}(G)\), \(u_i^G(λ_i, λ_{−i}) < +∞\) and \(\text{br}_i^G(λ_{−i}) ≤ u_i^G(λ_i, λ_{−i}) < +∞\). Thus \(\text{reg}_i^G ≤ \text{reg}_i^G(λ_i) < +∞\). □

**Example 1.** Consider the game arena \(G\) of Fig. We omit Player 2’s weights since we are interested in computing the regret of Player 1. Player 1’s positions are circle nodes and Player 2’s positions are square nodes. The target nodes are represented by double circles. The initial node is \(A\). Let \(λ_1\) be the memoryless strategy defined...
by \( \lambda_1(B) = C \) and \( \lambda_1(C) = E \). For all \( \lambda_2 \in \Lambda_2(G) \), \( \text{Out}^G_1(\lambda_1, \lambda_2) \) is either \( \text{ACE} \) or \( \text{ABCE} \), depending on whether Player 2 goes directly to \( C \) or passes by \( B \). In both cases, the outcome is winning and \( u^G_i(\lambda_1, \lambda_2) = 3 \). What is the regret of playing \( \lambda_1 \) for Player 1? To compute \( \text{reg}^G_i(\lambda_1) \), we should consider all possible strategies of Player 2, but a simple observation allows us to restrict this range. Indeed, to maximize the regret of Player 1, Player 2 should cooperate in subtrees where \( \lambda_1 \) prevents to go, i.e. in the subtrees rooted at \( D \) and \( F \). Therefore we only have to consider the two following memoryless strategies \( \lambda_2 \) and \( \lambda_2' \): both \( \lambda_2 \) and \( \lambda_2' \) move from \( F \) to \( J \) and from \( D \) to \( H \), but \( \lambda_2(A) = B \) while \( \lambda_2'(A) = C \). In both cases, going to \( F \) is a best response to \( \lambda_2 \) and \( \lambda_2' \) for Player 1, i.e. \( \text{br}^G_1(\lambda_2) = \text{br}^G_1(\lambda_2') = 0 \). Therefore we get \( \text{reg}^G_i(\lambda_1, \lambda_2) = u^G_1(\lambda_1, \lambda_2) - \text{br}^G_1(\lambda_2) = 3 - 0 = 3 \). Similarly \( \text{reg}^G_i(\lambda_1, \lambda_2') = 3 \). Therefore \( \text{reg}^G_1(\lambda_1) = 3 \).

As a matter of fact, the strategy \( \lambda_1 \) minimizes the regret of Player 1. Indeed, if she chooses to go from \( B \) to \( D \), then Player 2 moves from \( A \) to \( B \) and from \( D \) to \( G \) (so that Player 1 gets a utility 3) and cooperates in the subtree rooted at \( C \) by moving from \( F \) to \( J \). The regret of Player 1 is therefore 3. If Player 1 moves from \( B \) to \( C \) and from \( C \) to \( F \), then Player 2 moves from \( A \) to \( C \) and from \( F \) to \( I \) (so that Player 1 gets a utility 4), and from \( D \) to \( H \), the regret of Player 1 being therefore 4. Similarly, one can show that all other strategies of Player 1 have a regret at least 3. Therefore \( \text{reg}^G_1(\lambda_1) = 3 \).

Note that the strategy \( \lambda_1 \) does not minimize the regret in the subgame defined by the subtree rooted at \( C \). Indeed, in this subtree, Player 1 has to move from \( C \) to \( F \), and the regret of doing this is \( 4 - 3 = 1 \). However the regret of \( \lambda_1 \) in the subtree is 3. This example illustrates a situation where a strategy that minimizes the regret in the whole game does not necessarily minimize the regret in the subgames. Therefore we cannot apply a simple backward algorithm to compute the regret. As we will see in the next section, we first have to propagate some information in the subgames.

3 Regret Minimization on Target-Weighted Graphs

In this section, our aim is to give an algorithm to compute the regret for Player \( i \). This is done by reduction to a minmax game, defined in the sequel. We say that we solve the regret minimization problem (RMP for short) if we can compute the minimal regret and a (finite representation of a) strategy that achieves this value.

**Minmax games** Let \( G = (S = S_1 \cup S_2, s_0, T, \mu_1, \mu_2, C_1, C_2) \) be a TWA and \( i = 1, 2 \). We define the value \( \text{minmax}^G_i \) as follows:

\[
\text{minmax}^G_i = \min_{\lambda_i \in \Lambda_1(G)} \max_{\lambda_{-i} \in \Lambda_{-i}(G)} u^G_i(\lambda_i, \lambda_{-i})
\]

**Proposition 2.** Given a TWA \( G = (S = S_1 \cup S_2, s_0, T, \mu_1, \mu_2, C_1, C_2) \), \( i \in \{1, 2\} \) and \( K \in \mathbb{N} \), one can decide in time \( O(|S| + |T|) \) whether \( \text{minmax}^G_i \leq K \). The value \( \text{minmax}^G_i \) and a memoryless strategy that achieves this value can be computed in time \( O(|S| + |T|) \).

**Proof.** For all \( j \geq 0 \), we let \( W_j \subseteq S \) be the set of positions from which Player \( i \) has a strategy to reach a position \( s \in C_i \) in at most \( j \) steps, such that \( \mu_i(s) \leq K \) and such that she does not pass by a position \( s' \in C_i \) such that \( \mu_i(s') > K \). Formally, we denote by \( C_i^{> K} \) the set of positions \( s \in C_i \) s.t. \( \mu_i(s) > K \). Then \( W_0 = C_i \setminus C_i^{> K} \) and for all \( j > 0 \), \( W_j = W_{j-1} \cup W_j^3 \cup W_j^4 \), where:

\[
W_j^3 = \{ s \in S_i \setminus C_i^{> K} \mid \exists s' \in W_{j-1}, (s, s') \in T \}
\]
\[
W_j^4 = \{ s \in S_i \setminus C_i^{> K} \mid \forall (s, s') \in T, s' \in W_{j-1} \}
\]
The sequence $W_0, W_1, \ldots$ converges in at most $|S|$ steps to a set $W^*$, and $\minmax^G_i \leq K$ iff $s_0 \in W^*$. In order to compute $W^*$ in time $O(|S| + |T|)$, we add counters to positions that counts the number of their successors that are not already in the current set $W_j$. When adding a new node to $W_j$, we decrement the counter of its predecessor by one (if it was not already 0). Let $s$ be one of its predecessors and $c$ its counter value. If $s \in S_i$ and $c$ is strictly less than the number of its successors, $s$ will be added to $W_{j+1}$. If $s \in S_{\lambda}$ and $c = 0$, then all the successors of $s$ are in $W_j$, therefore $s$ will be added to $W_{j+1}$.

Now, in order to compute the value $\minmax^G_i$, we use the previous algorithm as the building block of a dichotomy algorithm that starts with the maximal finite value which can be achieve by Player $i$ if she has a winning strategy to its target, i.e. $M^G_i$.

If $\minmax^G_i = +\infty$, then any strategy achieves this value. Otherwise in order to extract a strategy, it suffices to keep for each position $s \in W_j \cap S_i$, a pointer to a position $s' \in W_j-1$ such that $(s, s') \in T$ when computing the sequence of $W_j$’s. Note that this strategy is memory-less.

Since roles of the players are symmetric, without loss of generality we can focus on computing the regret of Player 1 only. Therefore we do not consider Player 2’s targets and weights. Let $G = (S = S_1 \uplus S_2, s_0, T, \mu_1, C_1)$ be a TWA (assumed to be fixed from now on). Let $\lambda_1 \in \Lambda_1(G)$ be a winning strategy of Player 1 (if it exists). Player 2 can enforce Player 1 to follow one of the paths of $\text{Out}^{G, C_1}(\lambda_1)$ by choosing a suitable strategy. When choosing a path $\pi \in \text{Out}^{G, C_1}(\lambda_1)$, in order to maximize the regret of Player 1, Player 2 cooperates (i.e. she minimizes the utility) if Player 1 would have deviated from $\pi$.

This leads to the notion of best alternative along a path. Informally, the best alternative along $\pi$ is the minimal value Player 1 could have achieved if she deviated from $\pi$, assuming Player 2 cooperates. Since Player 2 can enforce one of the paths of $\text{Out}^{G, C_1}(\lambda_1)$, to maximize the regret of Player 1, she will choose the path $\pi$ with the highest difference between $u^G_i(\pi)$ and the minimal best alternative along $\pi$. As an example consider the TWA arena of Fig. 4. In this example, if Player 1 moves from $C$ to $E$, then along the path $ACE$, the best alternative is 0. Indeed, the other alternative was to go from $C$ to $F$ and in this case, Player 2 would have cooperated.

We now formally define the notion of best alternative. Let $s \in S_1$. The best value that can be achieved from $s$ by Player 1 when Player 2 cooperates is defined by:

$$\text{best}^1_1(s) = \min_{\lambda_1 \in \Lambda_1(G,s)} \min_{\lambda_2 \in \Lambda_2(G,s)} u^{(G,s)}_1(\lambda_1, \lambda_2)$$

Let $(s, s') \in T$. The best alternative of choosing $s'$ from $s$ for Player 1, denoted by $\text{ba}^G_1(s, s')$, is defined as the minimal value she could have achieved by choosing another successor of $s$ (assuming Player 2 cooperates). Formally:

$$\text{ba}^G_1(s, s') = \left\{ \begin{array}{ll} +\infty & \text{if } s \in S_2 \\
 \min_{(s, s'') \in T, s'' \neq s} \text{best}^1_1(s'') & \text{if } s \in S_1 \\
 \min_{0 \leq i < n} \text{ba}^G_1(s_j, s_{j+1}) & \text{otherwise} \end{array} \right.$$
Because the number of best alternatives is bounded by \(|C_1|\), the game \(G'\) can be constructed in polynomial time:

**Proposition 4.** \(G'\) can be constructed in time \(O((|C_1| + \log_2(M^G_1)) \times (|S| + |T|))\).

Since the best alternative information depends only on the paths, the paths of \(G\) and those of \(G'\) are in bijection. This bijection can be extended to strategies. In particular, we define two mappings \(\Phi_1, \Phi_2: \Lambda_i(G) \rightarrow \Lambda_i(G')\), for all \(i = 1, 2\). For all path \(\pi = s_0s_1\ldots\) in \(G\) (finite or infinite), we denote by \(B(\pi)\) the path of \(G'\) defined by \((s_0, b_0)(s_1, b_1)\ldots\) where \(b_0 = +\infty\) and for all \(j > 0\), \(b_j = ba_i^1(s_0\ldots s_{j-1})\). The mapping \(B\) is bijective, and its inverse corresponds to \(\text{proj}_1\).

The mapping \(\Phi_1\) maps any strategy \(\lambda_1 \in \Lambda_i(G)\) to a strategy \(\Phi_1(\lambda_1) \in \Lambda_i(G')\) such that \(\Phi_1(\lambda_1)\) behaves as \(\lambda_1\) on the first projection of the play and adds the best alternative information to the position. Let \(h \in S^*\) such that last\((h)\) \(\in S_i\). Let \(s = \lambda_1(\text{proj}_1(h))\). Then \(\Phi_1(\lambda_1)(h) = (s, ba_i^1(\text{proj}_1(h), s))\). The inverse mapping \(\Phi_1^{-1}\) just projects the best alternative information away. In particular, for all \(\lambda'_1 \in \Lambda_i(G')\), and all \(\lambda' \in S^*\) such that last\((h)\) \(\in S_i\), \(\Phi_1^{-1}(\lambda'_1)(h) = \text{proj}_1(\lambda_1(B(h)))\).

Then, \(\Phi_1\)'s are bijective and \(\Phi_1\) preserves the regret values:

**Lemma 1.** \(\forall \lambda_1 \in \Lambda_i(G), \text{reg}^{G'}_1(\lambda_1) = \text{reg}^{G'}_1(\Phi_1(\lambda_1))\).

The best alternative information is crucial to compute the regret. This is a global information that allows us to compute the regret locally, as stated by the next lemma. For all \((s, b) \in C_i\), let \(\nu_1(s, b) = \mu_1(s) - \min(\mu_1(s), b)\). We extend \(\nu_1\) to pairs of strategies as usual – \(\nu_1(\lambda_1, \lambda_2)\) being infinite if \(\lambda_1\) is losing. -

**Lemma 2.** \(\forall \lambda_1 \in \Lambda_i(G'), \text{reg}^{G'}_1(\lambda_1) = \max_{\lambda_2 \in \Lambda_2(G')} \nu_1(\lambda_1, \lambda_2)\).

**Proof.** (Sketch) It is clear if \(\lambda_1\) is losing. If it is winning, then let \(\lambda_2\) which maximizes \(\text{reg}^{G'}_1(\lambda_1)\) and \(\pi = \text{Out}^{G', C_1}(\lambda_1, \lambda_2)\). Without changing the regret values, we can assume that \(\lambda_2\) cooperates if Player 1 would have deviated from \(\pi\), i.e. \(\lambda_2\) minimizes the utility in the subgames \((G, s)\) where \(s\) is not the successor of some element of \(\pi\). The best response to \(\lambda_2\) is either the value \(U^{G'}_1(\lambda_1, \lambda_2)\), i.e. \(\mu_1(\text{last}(\pi))\), or the minimal best alternative along \(\pi\). By Proposition 3, this minimal best alternative along \(\pi\) is exactly \(\text{proj}_2(\text{last}(\pi))\).

Therefore \(\text{bf}^{G'}_1(\lambda_2) = \min(\mu'_1(\text{last}(\pi)), ba_i^1(\pi))\) and \(\text{reg}^{G'}_1(\lambda_1) = \nu^1(\text{last}(\pi)) = \nu^1(\lambda_1, \lambda_2)\). Conversely, for any strategy \(\lambda_2\) which maximizes \(\nu^1(\lambda_1, \lambda_2)\), we can also assume without changing the value \(\nu^1(\lambda_1, \lambda_2)\) that \(\lambda_2\) cooperates if Player 1 would have deviated from \(\text{Out}^{G'}(\lambda_1, \lambda_2)\), and we therefore have \(\text{reg}^{G'}_1(\lambda_1, \lambda_2) = \nu^1(\lambda_1, \lambda_2)\).

We can now reduce the RMP to a min-max problem:

**Lemma 3.** Let \(H = (S', s'_0, T', \nu', C'_1)\) where \(S', s'_0, T', C'_1\) are defined in Definition 1. Then

\[
\text{reg}^G_1 = \text{minmax}^H
\]

**Proof.** It is a consequence of Lemmas 1 and 2.

As a consequence of Propositions 2, 4 and Lemma 3, we can solve the RMP on TWAs. We first compute the graph of best alternatives and solve a minmax game. This gives us a memoryless strategy that achieves the minimal regret in the graph of best alternatives. To compute a strategy in the original graph, we apply the inverse mapping \(\Phi_1^{-1}\): this gives a finite-memory strategy whose memory is exactly the best alternative seen along the current finite play. Therefore the needed memory is bounded by the number of best alternatives, which is bounded by \(|\mathcal{C}_1|\).

**Theorem 1.** The RMP on a TWA \(G = (S, s_0, T, \mu_1, C_1)\) can be solved in \(O(|\mathcal{C}_1| \cdot \log_2(M_1^G) \times (|S| + |T|))\).

4 Regret Minimization in Edge-Weighted Graphs

In this section, we give a pseudo-polynomial time algorithm to solve the RMP in weighted arenas (with weights on edges). In a first step, we prove that if the regret is
finite, the strategies minimizing the regret generates outcomes whose utility is bounded by some value which depends on the graph. This allows us to reduce the problem to the RMP in a TWA, which can then be solved by the algorithm of the previous section.

Let \( G = (S = S_1 \uplus S_2, s_0, T, \mu_1, C_1) \) be a weighted game arena with objective \( C_1 \). As in the previous section, we assume that we want to minimize the regret of Player 1, so we omit the weight function and the target of Player 2.

**Definition 2** (Bounded strategies). Let \( B \in \mathbb{N} \) and \( \lambda_1 \in \Lambda_1(G) \). The strategy \( \lambda_1 \) is bounded by \( B \) if for all \( \lambda_2 \in \Lambda_2(G) \), \( u^G(\lambda_1, \lambda_2) \leq B \).

Note that a bounded strategy is necessarily winning, since by definition, the utility of some outcome is infinite if it is loosing. The following lemma states that the winning strategies that minimize the regret of Player 1 are bounded.

**Lemma 4.** For all weighted arena \( G = (S, s_0, T, \mu_1, C_1) \) and for any strategy \( \lambda_1 \in \Lambda_1(G) \) winning in \( G \) for Player 1 that minimizes her regret, \( \lambda_1 \) is bounded by \( 2M^G|S| \).

**Proof.** Since we consider reachability games, it is well-known that if there is a winning strategy for Player 1, there is a memoryless strategy \( \gamma_1 \) winning for Player 1 (see for instance [5]). In particular, for all \( \lambda_2 \in \Lambda_2(G) \), \( \text{Out}^G \gamma_1(\lambda_1, \lambda_2) \) does not contain twice the same position. Indeed, if there is a loop, then the strategy is memoryless, Player 2 can enforce Player 1 to take this loop infinitely many times. Therefore for all \( \lambda_2 \in \Lambda_2(G) \), \( u^G(\gamma_1, \lambda_2) \leq M^G|S| \). Therefore the following holds: \( \forall \lambda_2 \in \Lambda_2(G) \), \( b_1^G(\lambda_2) \leq M^G|S| \). Moreover, \( \text{reg}^G(\gamma_1) \leq M^G|S| \). Indeed, let \( \lambda_2 \) which maximizes \( \text{reg}^G(\gamma_1, \lambda_2) \). Then \( \text{reg}^G(\gamma_1) = u^G(\gamma_1, \lambda_2) - b_1^G(\lambda_2) \). Since \( u^G(\gamma_1, \lambda_2) \leq M^G|S| \) and \( 0 \leq b_1^G(\lambda_2) \leq M^G|S| \), we get \( \text{reg}^G(\gamma_1) \leq M^G|S| \). Thus \( \text{reg}^G(\gamma_1) \leq M^G|S| \).

Finally let \( \lambda_1 \) be a winning strategy which minimizes the regret of Player 1, and \( \lambda_2 \in \Lambda_2(G) \). We have \( \text{reg}^G(\lambda_1, \lambda_2) \leq M^G|S| \) (by \( \ast \ast \)), therefore \( u^G(\lambda_1, \lambda_2) - b_1^G(\lambda_2) \leq M^G|S| \), which gives \( u^G(\lambda_1, \lambda_2) \leq 2M^G|S| \) (by \( \ast \)).

Let \( B = 2M^G|S| \). Thanks to Lemma 4 we can reduce the RMP in a weighted arena into the RMP in a TWA. Indeed, it suffices to enforce every position of the arena with the sum of the weights occurring along the path used to reach this position. A position may be reachable by several paths, therefore it will be duplicated as many times as they are different path utilities. This may be unbounded, but Lemma 4 ensures that it is sufficient to sum the weights up to \( B \) only. This may results in a larger graph, but its size is still pseudo-polynomial (polynomial in the maximal weight and the size of the graph).

**Definition 3.** Let \( G = (S = S_1 \uplus S_2, s_0, T, \mu_1, C_1) \) be a weighted game arena. The graph of utility is the TWA \( G' = (S' = S'_1 \uplus S'_2, s'_0, T', \mu'_1, C'_1) \) defined by:

- \( S'_i \subseteq S_i \times [B] \) for all \( i \in \{1, 2\} \);
- for all \( (s, u), (s', u') \in S' \), \( (s, u), (s', u') \in T' \) iff \( (s, u) \in T \) and \( u' = u + \mu_1(s, s') \);
- \( C'_1 = (C_1 \times [B]) \cap S' \) and \( \forall (s, u) \in C'_1, \mu'_1(s, u) = u \).

We now prove that \( \text{reg}^G(\gamma_1) = \text{reg}^{G'}(\gamma'_1) \). The utility information added to the nodes of \( G \) is uniquely determined by the path used to reach the current position. Therefore the strategies of both players in \( G \) can naturally be mapped to strategies in \( G' \). More formally, we define a mapping \( \Phi \) from \( \Lambda_1(G) \cup \Lambda_2(G) \) into \( \Lambda_1(G') \cup \Lambda_2(G') \). Let \( i \in \{1, 2\} \) and \( \lambda_i \in \Lambda_i(G) \). Let \( h \in \mathbb{P}_i(G') \) such that last \( (h) \in S'_i \).

Let \( s = \lambda_i(\text{proj}_1(h)) \) and \( u = \mu_1(\text{proj}_1(h), s) \).

Then \( \Phi(\lambda_i)(h) = \begin{cases} \perp & \text{if } u > B \\ (s, u) & \text{otherwise} \end{cases} \)

The mapping \( \Phi \) is surjective, but not necessarily injective. Indeed, two strategies that behave similarly up to an utility \( B \) are mapped to the same strategy in \( G' \). Let \( \lambda'_i \in \Lambda_i(G') \). Any strategy \( \lambda_i \in \Lambda_i(G) \) that behaves like \( \lambda'_i \) (on the first projections of plays) while the utility of the play is bounded by \( B \) is a preimage of \( \lambda'_i \). Formally, for all \( h = (s_0, u_0, \ldots, s_n, u_n) \in \mathbb{P}(G) \), we let \( \tilde{h} = (s_0, u_0)(s_1, u_1) \ldots (s_n, u_n) \) where for all \( j \), \( u_i = \mu_1(s_0, \ldots, s_j) \). Then, any strategy \( \lambda_i \in \Lambda_i(G) \) is a preimage of \( \lambda'_i \) iff for all finite play \( h \in \mathbb{P}(G) \) such that last \( (h) \in S_i \), all \( s \in S \), all \( u \in \mathbb{N} \), if \( \lambda'_i(\tilde{h}) \) is defined and equal to \( (s, u) \), then \( \lambda_i(h) = s \).

**Lemma 5.** For all \( i = 1, 2 \), \( \Phi(\Lambda_i(G)) = \Lambda_i(G') \).
We denote by $\Lambda_{1}^{\leq B}(G)$ the set of strategies bounded by $B$. The mapping $\Phi$ preserves the regret values for bounded strategies:

**Lemma 6.** $\forall \lambda_1 \in \Lambda_{1}^{\leq B}(G), \text{reg}_{i}^{G}(\lambda_1) = \text{reg}_{i}^{\Phi(\lambda_1)}$.

**Proof.** (Sketch) This lemma is supported by the following result: for all $\lambda_1 \in \Lambda_1(G)$ and $\lambda_2 \in \Lambda_2(G)$. If $\text{Out}^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$ is winning for Player 1 in $G'$ or $u^G_1(\lambda_1, \lambda_2) \leq B$, then $u^G_1(\lambda_1, \lambda_2) = u^{G'}_1(\Phi(\lambda_1), \Phi(\lambda_2))$.

Note that any strategy $\lambda_1 \in \Lambda_1(G)$ is bounded by $B$ iff $\Phi(\lambda_1)$ is winning in $G'$ for Player 1. We can now prove the correctness of the reduction:

**Lemma 7.** $\text{reg}_{i}^{G} = \text{reg}_{G'}^{i}$

**Proof.** Suppose that $\text{reg}_{i}^{G} = +\infty$. If $\text{reg}_{i}^{G'} < +\infty$, then there is a strategy $\lambda_1' \in \Lambda_1'(G')$ winning in $G'$ for Player 1. By Lemma 6 $\lambda_1' = \Phi(\lambda_1)$ for some $\lambda_1 \in \Lambda_1(G)$. Since $\Phi(\lambda_1)$ is winning, $\lambda_1$ is bounded by $B$, and for a fortiori winning. Thus $\text{reg}_{i}^{G'} < +\infty$, which is a contradiction. Therefore $\text{reg}_{i}^{G'} = \text{reg}_{i}^{G} = +\infty$.

To solve the RMP for a weighted arena $G$, we first construct the graph of utility $G'$, and then apply Theorem 4 since $G'$ is a TWA. Correctness is ensured by Lemma 7. This returns a finite-memory strategy of $G'$ that minimizes the regret, whose memory is the best alternative seen so far. To obtain a strategy of $G$ minimizing the regret, one applies the inverse mapping $\Phi^{-1}$ defined previously. This gives us a finite-memory strategy whose memory is the utility of the current play up to $M^G$ and the best alternative seen so far.

**Theorem 2.** The RMP on a weighted arena $G = (S = S_1 \cup S_2, s_0, T, \mu_1, \mu_2, C_1, C_2)$ can be solved in time $O((M^G)^2 \cdot \log_2(|S| \cdot M^G) \cdot |S| \cdot C_1 \cdot (|S| + |T|))$.

5 Iterated Regret Minimization (IRM)

In this section, we show how to compute the iterated regret for tree arenas and for weighted arenas where weights are strictly positive (by reduction to a tree arena).

Let $G = (S = S_1 \cup S_2, s_0, T, \mu_1, \mu_2, C_1, C_2)$ be a weighted arena. Let $i \in \{1, 2\}$, $P_i \subseteq \Lambda_i(G)$ and $P_{-i} \subseteq \Lambda_{-i}(G)$. The regret of Player $i$ when she plays strategies of $P_i$ and when Player $-i$ plays strategies of $P_{-i}$ is defined by:

$$
\text{reg}_{i}^{G, P_i, P_{-i}} = \min_{\lambda_i \in P_i} \max_{\lambda_{-i} \in P_{-i}} \sum_{t \in T} r_{i}^{G}(\lambda_i, \lambda_{-i}) \text{br}_{-i}(\lambda_{-i})$$

For all $\lambda_i \in P_i$ and $\lambda_{-i} \in P_{-i}$, we define $\text{reg}_{i}^{G, P_i, P_{-i}}(\lambda_i)$ and $\text{reg}_{i}^{G, P_i, P_{-i}}(\lambda_{-i})$ accordingly. We now define the strategies of rank $j$, which are the one that survived $j$ times the deletion of strictly dominated strategies. The strategies of rank 0 for Player $i$ is $\Lambda_i(G)$. The strategies of rank 1 for both players are those which minimize their regret against strategy of rank 0. More generally, the strategies of rank $j$ for Player $i$ are the strategies of rank $j - 1$ which minimize her regret against Player $-i$’s strategies of rank $j - 1$. Formally, strategies of rank $j$ are obtained via a delete operator $D : 2^{\Lambda_1(G)} \times 2^{\Lambda_2(G)}$ $\rightarrow 2^{\Lambda_1(G)} \times 2^{\Lambda_2(G)}$ such that for all $P_1 \subseteq \Lambda_1(G)$ and all $P_2 \subseteq \Lambda_2(G)$,

$$D(P_1, P_2) = \{ \lambda_1 \in P_1 | \text{reg}_{i}^{G, P_i, P_{-i}}(\lambda_1) \times \lambda_2 \in P_2 | \text{reg}_{i}^{G, P_i, P_{-i}}(\lambda_2) \}$$

We denote by $D^j$ the composition of $D$ $j$ times.

**Definition (j-th regret).** Let $j \geq 0$. The set of strategies of rank $j$ for Player $i$ is $P_i^j = \text{proj}_j(D^j(\Lambda_1(G), \Lambda_2(G)))$. The $j + 1$-th regret for Player $i$ is defined by $\text{reg}_{i}^{G, j+1} = \text{reg}_{i}^{G, P_i^j, p_{-i}^j}$. In particular, $\text{reg}_{i}^{G, 1} = \text{reg}_{i}^{G}$.

**Proposition 5.** Let $i \in \{1, 2\}$. For all $j \geq 0$, $P_i^{j+1} \subseteq P_i^j$ and $\text{reg}_{i}^{G, j+1} \leq \text{reg}_{i}^{G, j}$.
Proof. (Sketch) $D^{j+1}_i \subseteq P^j_i$ is by definition the operator $D$. For all $\lambda_{-i} \in P^j_i$, $br_{G,P^j_i}(\lambda_{-i}) \geq br_{G,P^j_i}(\lambda_{-i})$ (because we minimize over less strategies). Thus for all $\lambda_i \in P^j_i$ and $\lambda_{-i} \in P^j_i$, $reg_{G,j+1}(\lambda_i,\lambda_{-i}) \leq reg_{G,j}(\lambda_i,\lambda_{-i})$. Since $P^j_i \subseteq P^2_i$, $reg_{G,j+1}(\lambda_i) \leq reg_{G,j}(\lambda_i)$ (because we maximize over less strategies). Therefore $reg_{G,j+1} \leq reg_{G,j}(\lambda_i) = reg_{G,j}$.  

Clearly, the sequence of regrets converges:

**Proposition 6.** There is an integer $\ast \geq 1$ such that for all $j \geq \ast$, for all $i \in \{1, 2\}$, $reg_{G,j} = reg_{G,\ast}$.

**Definition 5** (iterated regret). For all $i = 1, 2$, the iterated regret of Player $i$ is $reg_{G,\ast}$.

**Example 2.** As we already saw in the Centipede Game depicted on Fig. 2, the Player 1’s strategy minimizing her regret is to stop at the last step (move from A to B, from C to D and from E to S). Its regret value is 1. The Player 2’s strategy minimizing her regret is also to stop at the last step, i.e. to move from B to C and from D to E, her regret being 2. Therefore $reg_1 = reg_{2,1} = 1$ and $reg_2 = reg_{2,1} = 1$. If Player 1 knows that Player 2 will ultimately move to E, she can play the same strategy as before, and her regret $reg_{G,2} = 0$. Similarly $reg_{G,2} = 0$. Therefore $reg_{G,\ast} = reg_{G,\ast}$.

5.1 IRM in Tree Arenas

In this section, we let $i \in \{1, 2\}$ and $G = (S = S_1 \uplus S_2, s_0, T, \mu_1, \mu_2, C_1, C_2)$ be a finite edge-weighted tree arena. We can transform $G$ into a target-weighted tree arena such that $C_1 = C_2$ (denoted by $C$ in the sequel) is the set of leaves of the tree, if we allow the functions $\mu_i$ to take the value $+\infty$. This transformation results in a new target-weighted tree arena $G' = (S = S_1 \uplus S_2, s_0, T, \mu_1', \mu_2', C)$ with the same set of states and transitions as $G$ and for all leaf $s \in C$, $\mu_i'(s) = u_i^G(\pi)$, where $\pi$ is the root-to-leaf path leading to $s$. The time complexity of this transformation is $O(|\mathcal{S}|)$.

We now assume that $G = (S = S_1 \uplus S_2, s_0, T, \mu_1, \mu_2, C)$ is a target-weighted tree arena where $C$ is the set of leaves. Our goal is to define a delete operator $D$ such that $D(G)$ is a subtree of $G$ such that for all $i = 1, 2$, $\Lambda_i(D(G))$ are the strategies of $\Lambda_i(G)$ that minimize $reg_{G,i}$. In other words, any pairs of subsets of strategies for both players in $G$ can be represented by a subtree of $G$. This is possible since all the strategies in a tree arena are memoryless. A set of strategies $P_i \subseteq \Lambda_i(G)$ is therefore represented by removing from $G$ all the edges $(s, s')$ such that there is no strategy $\lambda_i \in P_i$ such that $\lambda_i(s) = s'$. In our case, one first compute the sets of strategies that minimize regret. This is done as in Section 3 by constructing the tree of best alternatives $H$ (but in this case with the best alternative of both players) and by solving a min-max game. From $H$ we delete all edges that are not compatible with a strategy that minimize the minmax value of some player. We obtain therefore a subtree $D(H)$ of $H$ such that any strategy of $H$ is a strategy of $D(H)$ for Player $i$ iff it minimizes the minmax value in $H$ for Player $i$. By projecting away the best alternative information in $D(H)$, we obtain a subtree $D(G)$ of $G$ such that any Player $i$’s strategy of $G$ is a strategy of $D(G)$ iff it minimizes Player $i$’s regret in $G$. We can iterate this process to compute the iterated regret, and we finally obtain a subtree $D^*(G)$ such that any strategy of $G$ minimizes the iterated regret for Player $i$ iff it is a Player $i$’s strategy in $D^*(G)$.

**Definition 6.** The tree of best alternatives of $G$ is the tree $H = (S' = S'_1 \uplus S'_2, s'_0, T', \mu'_1, \mu'_2, C')$ defined by:

- $S'_i = \{(s, b_1, b_2) \mid s \in S_i, b_n = ba^C_i(\pi_s), k = 1, 2\}$, where $\pi_s$ is the path from the root $s_0$ to $s$;
- $s'_0 = (s_0, +\infty, +\infty)$;
- $\forall s, s' \in S', (s, s') \in T'$ iff $(\text{proj}_1(s), \text{proj}_1(s')) \in T$;
- $C' = \{s \in S' \mid \text{proj}_1(s) \in C\}$;
- $\forall (s, b_1, b_2) \in C', \mu'_1(s, b_1, b_2) = \mu_i(s) - \min(\mu_i(s), b_1)$.

Note that $H$ is isomorphic to $G$. There is indeed a one-to-one mapping $\Phi$ between the states of $G$ and the states of $H$: for all $s \in S$, $\Phi(s)$ is the only state $s' \in S'$ of the form $s' = (s, b_1, b_2)$. Moreover, this mapping is naturally extended to strategies. Since all strategies are memoryless, any strategy $\lambda_i \in \Lambda_i(G)$ is a function $S_i \rightarrow S_i$. Thus, for all $s' \in S'_i$, $\Phi(\lambda_i)(s') = \Phi(\lambda_i)(\Phi^{-1}(s'))$. Without loss of generality and for a technical reason, we assume that any strategy $\lambda_i$ is only defined for states $s \in S_i$ that are compatible with this strategy, i.e. if $s$ is not reachable under $\lambda_i$ then the value of $\lambda_i$ does not need to be defined. The lemmas of Section 3 still hold for the tree $H$.
Lemma 8. For all $i \in \{1, 2\}$, $\Phi(\Lambda_i(G)) = \Lambda_i(H)$ and any strategy $\lambda_i \in \Lambda_i(G)$ minimizes $\text{reg}_i^G$ iff $\Phi(\lambda_i)$ minimizes $\text{minmax}_i^H$. Moreover $\text{reg}_i^G = \text{minmax}_i^H$.

As in Section 3 the RMP on a tree arena can be solved by min-max game. For all $s \in S'$, we define $\text{minmax}_i^H(s) = \text{minmax}_i^{(H,s)}$ and compute this value by a backward induction algorithm. In particular, $\text{minmax}_i^H = \text{minmax}_i^H(s_0')$ and for all $s \in S'$:

$$\text{minmax}_i^H(s) = \begin{cases} 
\mu_i'(s) & \text{if } s \in C' \\
\min_{(s,s')} \text{minmax}_i^H(s') & \text{if } s \in S' \\
\max_{(s,s')} \text{minmax}_i^H(s') & \text{if } s \in S'' 
\end{cases}$$

Theorem 3. The RMP on a tree arena $G = (S, s_0, T, \mu_1, \mu_2, C)$ can be solved in $O(|S|)$.

The backward algorithm not only allows us to compute $\text{minmax}_i^H$ for all $i \in \{1, 2\}$, but also to compute a subtree $D(H)$ that represents all the Player $i$'s strategies that achieve this value. We actually define the operator $D$ in two steps. First, we remove the edges $(s, s') \in T'$, such that $s \in S'$ and $\text{minmax}_i^H(s') > \text{minmax}_i^H$ for all $i = 1, 2$. We obtain a new graph $H'$ consisting of several disconnected tree components. In particular, there are some states no longer reachable from the root $s_0'$, and we keep the connected component that contains $s_0'$ and obtain a new tree $D(H)$.

Player $i$'s strategies in $D(H)$ are not in the stricter sense strategies of $H$, as they do not specify what to play when Player $-i$ leads Player $i$ to a position that is not in $D(H)$. More formally, let $\lambda_i$ be a strategy of Player $i$ defined on $D(H)$ and $\lambda_{-i}$ a strategy of Player $-i$ on $H$. If there is a position $s$ of $D(H)$ owned by Player $-i$ such that $\lambda_{-i}$ leads to $s$ when Player $i$ plays $\lambda_i$, and if $\lambda_{-i}(s') = s'$ for some position $s'$ not in $D(H)$, then $\lambda_i(s')$ is undefined. This never happens when $\lambda_i$ is opposed to a strategy $\lambda_{-i}$ of $D(H)$, but it may happen when opposed to a strategy $\lambda_{-i}$ of $H$. For this reason, we define the strategies $\Lambda_i(H)$ of $D(H)$ for Player $i$ as the strategies of $H$ such that for all $s \in S'$, $(s, \lambda_i(s))$ is an edge of $H'$. We denote again by $\Lambda_i(D(H))$ this set of strategies. With this definition, any strategy $\lambda_i \in \Lambda_i(H)$ is defined on its outcomes in $H$, but when opposed to any strategy $\lambda_{-i} \in \Lambda_{-i}(D(H))$, its outcomes are in $D(H)$. Thus, when we iterate this operator, we do not need to remember $H'$ and we can consider only the tree $D(H)$. The tree $D(H)$ represents the strategy of $H$ that minimize the regret in the following sense:

Lemma 9. Let $i \in \{1, 2\}$. Let $\lambda_i \in \Lambda_i(H)$; $\text{minmax}_i^H(\lambda_i) = \text{minmax}_i^H$ iff $\lambda_i \in \Lambda_i(D(H))$.

Since there is a one-to-one correspondence between the strategies minimizing the regret in $G$ and the strategies minimizing the minmax value in $H$, we can define $D(G)$ by applying to $D(H)$ the isomorphism $\Phi^{-1}$, in other words by projecting the best alternatives away, and by restoring the functions $\mu_i$. The set of strategies $\Lambda_i(D(G))$ of $D(G)$ is defined as $\Phi^{-1}(\Lambda_i(D(H)))$ (in other words, these are the strategies of $D(H)$ where we project the best alternatives away). Let $\lambda_i \in \Lambda_i(G)$, by Lemma 8 it minimizes $\text{reg}_i^G$ iff $\Phi(\lambda_i)$ minimizes $\text{minmax}_i^H$, and by Lemma 9 iff $\Phi(\lambda_i) \in \Lambda_i(D(H))$, and finally, iff $\lambda_i \in \Lambda_i(D(G))$. $D(G)$ represents the strategies of $G$ minimizing the regret in the following sense:

Lemma 10. Let $i \in \{1, 2\}$. Let $\lambda_i \in \Lambda_i(G)$; $\text{reg}_i^G(\lambda_i) = \text{reg}_i^G$ iff $\lambda_i \in \Lambda_i(D(G))$.

We obtain a new tree $D(G)$ whose Player $i$’s strategies minimize the regret of Player $i$, for all $i = 1, 2$. We can iterate the regret computation on $D(G)$ and get the Player $i$’s strategies that minimize the regret of rank 2 of Player $i$, for all $i = 1, 2$. We continue iteration we get a tree $G'$ such that $D(G') = G'$. We let $D^0(G) = G$ and $D^{i+1}(G) = D(D^i(G))$. Remind that $P_i^0$ are Player $i$’s strategies of $G$ that minimize the $j$-th regret.

Proposition 7. Let $i \in \{1, 2\}$ and $j > 0$. We have $\text{reg}_i^{G,j} = \text{reg}_i^{D_{j-1}(G)}$ and $P_i^j = \Lambda_i(D^j(G))$.

Proof. (sketch) By induction on $j$. It is clear for $j = 1$ and Lemma 10 ensures the correctness of the induction.

Theorem 4. Let $G = (S = S_1 \uplus S_2, s_0, T, \mu_1, \mu_2, C)$ be a tree arena. For all $i = 1, 2$, the iterated regret of Player $i$, $\text{reg}_i^{G,*}$, can be computed in $O(|S|^2)$.

Proof. By Propositions 6 and 7 there is an integer $j$ such that $\text{reg}_i^{G,*} = \text{reg}_i^{D_j(G)}$. According to the definition of $D(G)$, $j \leq |S|$ because we remove at least one edge of the tree at each step. Since $|D(G)|$ can be constructed in $O(|S|)$, the whole time complexity is $O(|S|^2)$.\]
5.2 IRM in Positive Weighted Arenas

A weighted arena $G$ is said to be positive if all edges are weighted by strictly positive weights only. In this section, we let $G = (S = S_1 \cup S_2, s_0, T, \mu_1, \mu_2, C_1, C_2)$ be a positive weighted arena. Remind that $P_i^j(G)$ is the set of strategies that minimize $\text{reg}^{G,j}_i$, for all $j \geq 0$ and $i = 1, 2$.

**Definition 7** ($j$-winning and $j$-bounded strategies).

Let $i \in \{1, 2\}$ and $\lambda_i \in \Lambda_i(G)$. The strategy $\lambda_i$ is $j$-winning if for all $\lambda_{-i} \in P_{-i}^j(G)$, $\text{Out}^j(\lambda_i, \lambda_{-i})$ is winning. It is $j$-bounded by some $B \geq 0$ if it is $j$-winning, and for all $\lambda_{-i} \in P_{-i}^j(G)$ and all $\kappa \in \{i, -i\}$, $\mu_{\kappa}(\text{Out}^j(\lambda_i, \lambda_{-i})) \leq B$.

Note that $j$-boundedness differs from boundedness as we require that the utilities of both players are bounded. We let $b^G = 6(M^G)^3|S|$. We get a similar result than the boundedness of strategies that minimize the regret of rank 1, but for any rank:

**Lemma 11.** For all $i = 1, 2$ and all $j \geq 0$, all $j$-winning strategies of Player $i$ which minimize the $(j+1)$-th regret are $j$-bounded by $b^G$.

**Proof.** (Sketch) First, if the regrets of first rank are infinite for both players, then by definition of the iterated regret, $P_1^i = \Lambda_i(G)$ and $P_2^i = \Lambda_2(G)$ and thus their regrets are infinite at any rank. Therefore there is no winning strategy at any rank (otherwise one of the regrets would be finite).

Suppose that the first regret of Player $i$ is finite for some $i = 1, 2$. By Lemma 6 the winning strategies minimizing her first regret are bounded by $2M^G|S|$. Since the weights are strictly positive, the lengths of the outcomes until $C_i$ are bounded by $2M^G|S|$, which allows us to bound the utilities of Player $i$ until a first visit to $C_i$ by $2(M^G)^2|S|$. Since $P_i^1(G) \subseteq P_i^j(G)$ for all $j \geq 1$, the strategies of Player $i$ (which are necessarily winning as the regret is finite) at any rank are bounded by $2(M^G)^2|S|$. This bound is then used (non-trivially) to bound the winning strategies of Player $i$ by $6(M^G)^3|S|$. The full proof is in Appendix.

**Lemma 11** allows us to reduce the problem to the iterated regret minimization in a weighted tree arena, by unfolding the graph arena $G$ up to some maximal utility value. Lemma 11 suggests to take $b^G$ for this maximal value. However the best responses to a strategy $j$-bounded by $b^G$ are not necessarily bounded by $b^G$, but they are necessarily $j$-bounded by $b^G \cdot M^G$, since the weights are strictly positive. Therefore we let $B^G = b^G \cdot M^G$ and take $B^G$ as the maximal value. Since the $j$-winning strategies are $j$-bounded by $b^G$ and the best responses are $j$-bounded by $B^G$, we do not loss the set of finite plays $\pi$ of $G$ such that $\mu_G(\pi) \leq K$, for all $i = 1, 2$.

Note that $P_K(G)$ is finite since $G$ has only strictly positive weights. The unfolding of $G$ up to $B^G$ is naturally defined by a tree weighted arena whose set of positions is $P_{B^G}(G)$.

**Definition 8.** Let $G = (S = S_1 \cup S_2, s_0, T, \mu_1, \mu_2, C_1, C_2)$ be a positive weighted arena. The $B^G$-unfolding of $G$ is the weighted tree arena $G' = (S' = S'_1 \cup S'_2, s_0, T', \mu_1, \mu_2, C_1', C_2')$ such that $S'_i = \{ \pi \in P_{B^G}(G) \mid \text{last}(\pi) \in S_i \}$ and for all $\pi, \pi' \in S'_i$, $(\pi, \pi') \in T'$ if $\text{last}(\pi)$, $\text{last}(\pi') \in T$ and $\pi' = \pi.\text{last}(\pi')$, and for all $i = 1, 2$, $\pi \in C_i$ if $\text{last}(\pi) \in C_i$ and $\mu_i(\pi, \pi') = \mu_i(\text{last}(\pi), \text{last}(\pi'))$.

We now prove that $\text{reg}^{G,*}_i = \text{reg}^{G'}_i$, for all $i = 1, 2$. As for edge-weighted arenas, this is done by defining a surjective mapping $\Phi$ from $\Lambda_i(G)$ to $\Lambda_i(G')$. For all $i = 1, 2$ and all $\lambda_i \in \Lambda_i(G)$, and all $\pi \in P_f(G)$ such that $\text{last}(\pi) \in S_i$, $\Phi(\lambda_i)(\pi) = \perp$ if there is $\kappa \in \{1, 2\}$ such that $\mu_\kappa(\lambda_i, \lambda_{-i}) > B^G$, and $\Phi(\lambda_i)(\pi) = \lambda_i(\pi)$ otherwise. This mapping is surjective, but not injective, since two strategies that behave similarly up to some utility $B^G$ are mapped to the same strategy.

**Lemma 12.** For all $j \geq 1$, $\Phi(P_i^j(G)) = P_i^j(G')$ and for all $\lambda_i \in P_i^j(G)$, $\text{reg}^{G',j}_i(\lambda_i) = \text{reg}^{G,j}_i(\Phi(\lambda_i))$.

This allows us to prove the correctness of the reduction:

**Lemma 13.** For all $i = 1, 2$, $\text{reg}^{G,*}_i = \text{reg}^{G',*}_i$.

**Proof.** We prove that for all $j \geq 1$, $\text{reg}^{G,j}_i = \text{reg}^{G',j}_i$. Let $\lambda_i \in P_i^j(G)$. By definition of $P_i^j(G)$, $\lambda_i$ minimizes the $j$-th regret, so that $\text{reg}^{G,j}_i(\lambda_i) = \text{reg}^{G',j}_i(\Phi(\lambda_i))$. By (1), $\text{reg}^{G,j}_i(\lambda_i) = \text{reg}^{G,j}_i(\Phi(\lambda_i))$. By (2), $\Phi(\lambda_i) \in P_i^j(G')$, therefore $\Phi(\lambda_i)$ minimizes the $j$-th regret in $G'$, so that $\text{reg}^{G',j}_i(\Phi(\lambda_i)) = \text{reg}^{G,j}_i(\lambda_i)$, from which we get $\text{reg}^{G,j}_i = \text{reg}^{G',j}_i$.

By applying the algorithm of Section 5.1 we get:
Theorem 5. The iterated regret for both players in a positive weighted arena $G$ can be computed in pseudo-exponential time (exponential in $|S|$, $|T|$ and $M^G$).

For all $i = 1, 2$, the procedure of Section 5.1 returns a finite-memory strategy $\lambda_i$ minimizing the iterated regret in $G'$ whose memory is the best alternatives seen so far by both players. From $\lambda_i$, we can compute a finite-memory strategy in $G$ minimizing the iterated regret of Player $i$, the needed memory is the best alternatives seen by both players and the current finite play up to $B^G$. When the utility is greater than $B^G$, then any move is allowed. Therefore one needs to add one more bit of memory expressing whether the utility is greater than $B^G$.

Finally, the unfolding of the graph arena up to $B^G$ is used to finitely represent the (potentially infinite) sets of strategies of rank $j$ in $G$. Finding such a representation is not obvious for the full class of weighted arenas, since before reaching its objective, a player can take a 0-cost loop finitely many times without affecting her minimal regret. This suggests to add fairness conditions on edges to compute the iterated regret. This is illustrated by the following example.

![Figure 4: Free loops](image)

Example 3. Consider the left example of Fig. 4. Player 1's strategies minimizing the regret are those that pass finitely many times by the edge $(A, B)$ and finally move to $C$. The regret is therefore 5. Similarly, the strategies minimizing Player 2's regret are those that pass finitely many times by $(B, A)$ and finally move to $C$. The regret is 5 as well. The regret of rank 2 for Player 1 is 5 as well, and the set of strategies minimizing it is also the same as before (and similar for Player 2). Indeed, the regret of a Player 1's strategy that passes $K$ times by $(A, B)$ is 5, since Player 2 can maximize her regret with a strategy that passes at least $K$ times by $(B, A)$. Thus $\text{reg}_{1}^{G, \star} = \text{reg}_{2}^{G, \star} = 5$.

On the right example, Player 1 has no winning strategy at the first rank and her regret is $+\infty$. However the strategies of Player 2 minimizing her regret are the ones that pass finitely many times through the loop. Therefore all the strategies of Player 1 are winning at rank 2. The iterated regret of both players is 0.

6 Conclusion

The theory of infinite qualitative non-zero sum games over graphs is still in an initial development stage. We adapted a new solution concept from strategic games to game graphs, and gave algorithms to compute the regret and iterated regret. The strategies returned by those algorithms have a finite memory. One open question is to know whether this memory is necessary. In other words, are memoryless strategies sufficient to minimize the (iterated) regret in game graphs? Another question is to determine the lower bound on the complexity of (iterated) regret minimization. Iterated regret minimization over the full class of graphs is still open. Finally, we think that this work can easily be extended to an $n$-player setting.

References

[1] Kaushik Basu. The traveler’s dilemma: Paradoxes of rationality in game theory. *American Economic Review*, 84(2):391–95, 1994.

[2] Thomas Brihaye, Véronique Bruyère, and Julie De Pril. Equilibria in quantitative reachability games. 2010. submitted, available at http://www.ulb.ac.be/di/ssd/cfv/.

[3] Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdzinski. Games with secure equilibria. In LICS, pages 160–169, 2004.

[4] Dana Fisman, Orna Kupferman, and Yoad Lustig. Rational synthesis. 2010. to appear in TACAS’10.
7 Appendix

7.1 Missing Proofs of Section

Proposition

Proof. Proof by induction on $|\pi|$.

If $|\pi| = 0$, then $ba_{G_1}^G(\pi) = +\infty = b_0$.

We now assume that the property is true for any finite path $\pi$ in $G''$ from $s'_0$ to some $(s, b)$ of length $k$. Let $\pi = (s_0, b_0) \ldots (s_k, b_k) (s_{k+1}, b_{k+1})$ be a path of length $k+1$. We have:

\[
ba_{G_1}^G(\pi) = \min_{0 \leq j < k+1} ba_{G_1}^G(s'_j, s'_{j+1})
\]

\[
= \min(\min_{0 \leq j < k} ba_{G_1}^G(s'_j, s'_{j+1}), ba_{G_1}^G(s'_{k}, s'_{k+1}))
\]

\[
= \min(b_k, ba_{G_1}^G(s'_k, s'_{k+1})) \text{ by induction hypothesis}
\]

\[
= \min(b_k, ba_{G_1}^G(s, s')) \text{ by definition of } G''
\]

\[
(\star) \text{ According to Definition } \forall (s, b) \in C_1': \mu_{G_1}^G(s, b) = \mu_1(s).
\]

Thus $\forall (s, b) \in S', \text{best}_{G'}_1((s, b)) = \text{best}_{G_1}^G(s)$ and $\forall ((s, b), (s', b')) \in T', ba_{G'}_1((s, b), (s', b')) = ba_{G_1}^G(s, s')$.

Proposition

Proof. Constructing $G'$ is done in three steps:

1. compute all the values $\text{best}_{G_1}^G(s)$, for all $s \in S$; this step is equivalent to looking for the shortest path to the objective and has a complexity of $O(log_2(M_G^G)(|S| + |T|))$.

2. compute all the values $ba_{G_1}^G(s, s')$, for all $(s, s') \in T$ such that $s \in S_1$; it can be computed with a time complexity $O(|T|)$

3. construct $G'$ by a fixpoint algorithm; this graph has at most $|C_1| \times |S|$ states and $|C_1| \times |T|$ transitions.
Lemma 2

Proof. Let $\lambda_1 \in \Lambda_1(G')$. If $\lambda_1$ is losing, there is a strategy $\lambda_2 \in \Lambda_2(G')$ such that $\text{Out}^{G'}(\lambda_1, \lambda_2)$ is losing. Therefore $\text{reg}^{G'}_1(\lambda_1) = \text{reg}^{G'}_1(\lambda_1, \lambda_2) = +\infty = \nu_1(\lambda_1, \lambda_2) = \max_{\lambda_2 \in \Lambda_2(G')} \nu_1(\lambda_1, \lambda_2)$.

Suppose that $\lambda_1$ is a winning strategy and let $\lambda_2 \in \Lambda_2(G')$ which maximizes $\text{reg}^{G'}_1(\lambda_1, \lambda_2)$. Let $\pi = s_0 s_1 \ldots s_n = \text{Out}^{G', G_1}(\lambda_1, \lambda_2)$. We define a strategy $\lambda'_2$ that plays as $\lambda_2$ on $\pi$ and cooperates with Player 1 if she would have deviated from $\pi$. Formally, for all $h \in \mathcal{P}(G')$ such that $\text{last}(h) \in S'_{\lambda_2}$ we let $\lambda'_2(h) = s_{j+1}$ if there is $j < n$ such that $h = s_0 s_1 \ldots s_j$. Otherwise we let $\lambda'_2(h) = s$ such that $(\text{last}(h), s) \in T'$ and $\text{best}^{G'}(s)$ is minimal (among the successors of last($h$)).

Clearly, $\pi = \text{Out}^{G', G_1}(\lambda_1, \lambda_2)$ and $b_1^{G'}(\lambda'_2) \leq b_1^{G'}(\lambda_2)$. Therefore $\text{reg}^{G'}_1(\lambda_1, \lambda_2) \leq \text{reg}^{G'}_1(\lambda_1, \lambda'_2)$. Since $\lambda_2$ maximizes the regret, we get $\text{reg}^{G'}_1(\lambda_1, \lambda_2) = \text{reg}^{G'}_1(\lambda_1, \lambda'_2)$.

The best response to $\lambda'_2$ either deviates from $\pi$ or not. If the best response deviates from $\pi$ at a node $s_j$, $j < n$, i.e. chooses a node $s'$ such that $s' \neq s_{j+1}$, then the utility of the best response, according to the definition of $\lambda'_2$, is $\text{best}^{G'}_1(s')$. The best response to $\lambda'_2$ minimizes over all those possibilities, therefore $b_1^{G'}(\lambda'_2) = \min(\mu_1(s_n) - \min_{\pi \notin s_{j+1}} \text{best}^{G'}_1(s'))$, i.e. $\min(\mu_1(s_n), b_1^{G'}(\pi)) = \text{proj}_2(\text{best}^{G'}_1(\pi))$. By Proposition 2 $b_1^{G'}(\pi) = \text{proj}_2(\text{best}^{G'}_1(\pi))$. Therefore $\text{reg}^{G'}_1(\lambda_1, \lambda_2) = \text{reg}^{G'}_1(\lambda_1, \lambda'_2) = \mu_1(s_n) - \min(\mu_1(s_n), p_2(s_n)) = \nu_1(s_n)$. From which we get $\text{reg}^{G'}_1(\lambda_1) \leq \max_{\lambda_2 \in \Lambda_2(G')} \nu_1(\lambda_1, \lambda_2)$.

Conversely, let $\lambda_2$ which maximizes $\nu_1(\lambda_1, \lambda_2)$. Since $\lambda_1$ is winning, we can define $\pi = \text{Out}^{G', G_1}(\lambda_1, \lambda_2)$. Similarly as forth direction of the proof, one can construct a strategy $\lambda'_2$ that plays like $\lambda_2$ along $\pi$ and cooperates with Player 1 when deviating from $\pi$. Clearly, this strategy has the same outcome as $\lambda_2$ and we get $\text{reg}^{G'}_1(\lambda_1, \lambda'_2) = \nu_1(\lambda_1, \lambda_2)$. Finally we have $\text{reg}^{G'}_1(\lambda_1) \geq \text{reg}^{G'}_1(\lambda_1, \lambda'_2) = \nu_1(\lambda_1, \lambda_2) = \max_{\lambda_2} \nu_1(\lambda_1, \lambda_2)$.

Lemma 14. Let $\lambda_1 \in \Lambda_1(G)$ and $\lambda_2 \in \Lambda_2(G)$. If $\text{Out}^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$ is winning for Player 1 in $G'$ or $\text{Out}^{G'}(\lambda_1, \lambda_2) \leq B$, then $\nu_1^{G'}(\lambda_1, \lambda_2) = \nu_1^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$.

Proof. If $\nu_1^{G'}(\lambda_1, \lambda_2) \leq B$, then $\text{Out}^{G'}(\lambda_1, \lambda_2)$ is winning, and we let $\pi = \text{Out}^{G', G_1}(\lambda_1, \lambda_2)$. We enrich $\pi$ with the utilities of Player 1 by defining a path $\pi' = (s_0, u_0) \ldots (s_n, u_n)$ where $\pi = s_0 \ldots s_n$ and for all $j \leq n$, $u_j = \mu_1^{G'}(s_0 \ldots s_j)$. Since $\pi$ is bounded, we have $u_j \leq B$ for all $j \leq n$, and by definition of $G'$, $\pi'$ is a path of $G'$. By definition of $\Phi$ we clearly have $\pi' = \text{Out}^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$, from which we get $\nu_1^{G'}(\lambda_1, \lambda_2) = \nu_1^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$.

If $\text{Out}^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$ is winning for Player 1, we let $\pi' = \text{Out}^{G', G_1}(\Phi(\lambda_1), \Phi(\lambda_2))$ and $\pi = \text{proj}_1(\pi')$. Clearly, $\pi'$ is a winning play for Player 1 in $G'$ and by definition of $\Phi$, $\pi' = \text{Out}^{G', G_1}(\lambda_1, \lambda_2)$, from which we get the equality of the utilities.

7.2 Missing Proofs of Section 4

Lemma 6. The proof of this lemma is supported by the following lemma, which says that under certain conditions, the utility of the outcomes in $G$ and $G'$ are equal modulo $\Phi$:

Lemma 14. Let $\lambda_1 \in \Lambda_1(G)$ and $\lambda_2 \in \Lambda_2(G)$. If $\text{Out}^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$ is winning for Player 1 in $G'$ or $\text{Out}^{G'}(\lambda_1, \lambda_2) \leq B$, then $\nu_1^{G'}(\lambda_1, \lambda_2) = \nu_1^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$.

Proof. If $\nu_1^{G'}(\lambda_1, \lambda_2) \leq B$, then $\text{Out}^{G'}(\lambda_1, \lambda_2)$ is winning, and we let $\pi = \text{Out}^{G', G_1}(\lambda_1, \lambda_2)$. We enrich $\pi$ with the utilities of Player 1 by defining a path $\pi' = (s_0, u_0) \ldots (s_n, u_n)$ where $\pi = s_0 \ldots s_n$ and for all $j \leq n$, $u_j = \mu_1^{G'}(s_0 \ldots s_j)$. Since $\pi$ is bounded, we have $u_j \leq B$ for all $j \leq n$, and by definition of $G'$, $\pi'$ is a path of $G'$. By definition of $\Phi$ we clearly have $\pi' = \text{Out}^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$, from which we get $\nu_1^{G'}(\lambda_1, \lambda_2) = \nu_1^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$.

If $\text{Out}^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$ is winning for Player 1, we let $\pi' = \text{Out}^{G', G_1}(\Phi(\lambda_1), \Phi(\lambda_2))$ and $\pi = \text{proj}_1(\pi')$. Clearly, $\pi'$ is a winning play for Player 1 in $G'$ and by definition of $\Phi$, $\pi' = \text{Out}^{G', G_1}(\lambda_1, \lambda_2)$, from which we get the equality of the utilities.
Proof. (Proof of Lemma 9) Let $\lambda_2 \in \Lambda_2(G)$ which maximizes $\text{reg}^{G'}_1(\lambda_1, \lambda_2)$, and $\lambda_1^*$ be the best response to $\lambda_2$. Therefore $\text{reg}^{G'}_1(\lambda_1) = u^{G'}_1(\lambda_1, \lambda_2) - u^{G'}_1(\lambda_1^*, \lambda_2)$. Since $\lambda_1$ is bounded by $B$, we have $u^{G'}_1(\lambda_1, \lambda_2) \leq B$ and $u^{G'}_1(\lambda_1^*, \lambda_2) \leq B$ (since $\lambda_1^*$ is at least as good as $\lambda_1$). By Lemma 14 we get $u^{G'}_1(\lambda_1, \lambda_2) = u^{G'}_1(\Phi(\lambda_1), \Phi(\lambda_2))$ and $u^{G'}_1(\lambda_1^*, \lambda_2) = u^{G'}_1(\Phi(\lambda_1^*), \Phi(\lambda_2))$.

By definition of the best response, $\text{br}^{G'}_1(\Phi(\lambda_2)) \leq u^{G'}_1(\Phi(\lambda_1^*), \Phi(\lambda_2))$. Therefore

$$\text{reg}^{G'}_1(\lambda_1) = \text{reg}^{G'}_1(\lambda_1, \lambda_2) = u^{G'}_1(\lambda_1, \lambda_2) - u^{G'}_1(\lambda_1^*, \lambda_2) = u^{G'}_1(\Phi(\lambda_1), \Phi(\lambda_2)) - u^{G'}_1(\Phi(\lambda_1^*), \Phi(\lambda_2)) \leq u^{G'}_1(\Phi(\lambda_1), \Phi(\lambda_2)) - \text{br}^{G'}_1(\Phi(\lambda_2)) = \text{reg}^{G'}_1(\Phi(\lambda_1), \Phi(\lambda_2)) \leq \text{reg}^{G'}_1(\Phi(\lambda_1))$$

Conversely, since $\Phi(\Lambda_2(G)) = \Lambda_2(G')$, there exists $\lambda_2 \in \Lambda_2(G)$ such that $\Phi(\lambda_2)$ maximizes $\text{reg}^{G'}_1(\Phi(\lambda_1), \Phi(\lambda_2))$. Similarly, there is $\lambda_1^* \in \Lambda_1(G)$ such that $\Phi(\lambda_1^*)$ is the best response to $\Phi(\lambda_2)$. Since $\lambda_1$ is bounded by $B$, we have $u^{G'}_1(\lambda_1, \lambda_2) \leq B$, and by Lemma 14 we get $u^{G'}_1(\lambda_1, \lambda_2) = u^{G'}_1(\Phi(\lambda_1), \Phi(\lambda_2))$. Since $\lambda_1$ is winning for Player 1 and bounded by $B$ in $G$, $\Phi(\lambda_1)$ is also winning for Player 1 in $G'$. Therefore $\Phi(\lambda_1^*)$ is also winning for Player 1 in $G'$ (since it does at least as good as $\Phi(\lambda_1)$ against $\Phi(\lambda_2)$). Therefore $\text{Out}^{G'}(\Phi(\lambda_1), \Phi(\lambda_2))$ is winning for Player 1 in $G'$, and by Lemma 14 we get $u^{G'}_1(\lambda_1^*, \lambda_2) = u^{G'}_1(\Phi(\lambda_1^*), \Phi(\lambda_2))$. Finally:

$$\text{reg}^{G'}_1(\Phi(\lambda_1)) = \text{reg}^{G'}_1(\Phi(\lambda_1)), \Phi(\lambda_2)) = u^{G'}_1(\Phi(\lambda_1), \Phi(\lambda_2)) - u^{G'}_1(\Phi(\lambda_1^*), \Phi(\lambda_2)) = u^{G'}_1(\lambda_1, \lambda_2) - u^{G'}_1(\lambda_1^*, \lambda_2) \leq u^{G'}_1(\lambda_1, \lambda_2) - \text{br}^{G'}_1(\lambda_2) = \text{reg}^{G'}_1(\lambda_1, \lambda_2) \leq \text{reg}^{G'}_1(\lambda_1)$$

7.3 Missing Proofs of Section 5

Lemma 8 By projecting away the best alternatives of Player $\neg i$ in $H$, we get a tree isomorphic to $H$ which corresponds exactly to the tree of best alternatives defined in Section 3 in which all the results stated in Lemma 8 have been already proved. Clearly, adding the best alternatives of the other player does not change those results.

Lemma 9

Proof. Let $\lambda_i \in \Lambda_i(H)$ such that $\text{minmax}^H_i(\lambda_i) = \text{minmax}^H_i$. Let $s \in S'_i$ be a position of $D(H)$ compatible with $\lambda_i$, i.e., such that there is $\lambda_i \in \Lambda_i(D(H))$ such that $s$ occurs in $\text{Out}^{D(H)}(\lambda_i, \lambda_i)$. Let $s' = \lambda_i(s)$. We have to prove that $s'$ is a position of $D(H)$. We have $\text{minmax}^H_i(s') \leq \text{minmax}^H_i(\lambda_i) = \text{minmax}^H_i$. Indeed, since Player $\neg i$ is able to enforce Player $i$ to go to $s'$ when she plays $\lambda_i$, if $\text{minmax}^H_i(s') > \text{minmax}^H_i$, then $\lambda_i$ does not minimize $\text{minmax}^H_i$. According to the definition of the delete operator, $(s, s')$ is an edge of $D(H)$. Thus $s'$ is a position of $D(H)$, and $\lambda_i \in \Lambda_i(D(H))$.

Conversely, if $\lambda_i \in \Lambda_i(D(H))$. We proceed reductio ad absurdum.

If $\text{minmax}^H_i(\lambda_i) > \text{minmax}^H_i$, there exists $\lambda_{i-j} \in \Lambda_{i-j}(H)$ such that $\text{minmax}^H_i(\lambda_{i-j}) = \text{minmax}^H_i(\lambda_i) > \text{minmax}^H_i$. We let $\pi = \pi_0 \ldots \pi_n = \text{Out}^H(\lambda_i, \lambda_{i-j})$. Let $s, b_1, b_2$ such that $\pi_n = (s, b_1, b_2)$. Since $\pi_n \in G'$, $\text{minmax}^H_i(\lambda_i) = \mu^H_i(\pi_n) = \text{minmax}^H_i(\pi_n)$. We consider the first position $\pi_n$ along $\pi$ such that $k < n$ and $\pi_n$ is owned by Player $\neg i$, i.e., $\pi_k \in S'_i$ and $\pi_{k+1} \ldots \pi_{n-1} \in (S'_i)^*$. This position exists, otherwise all positions $\pi_0, \ldots, \pi_{n-1}$ are owned by Player $\neg i$, and therefore $\text{minmax}^H_i \geq \mu^H_i(\pi_n) = \text{minmax}^H_i(\lambda_i)$, which contradicts our hypothesis. Since $\lambda_i \in \Lambda_i(D(H))$, by definition of $\Lambda_i(D(H))$, $(\pi_k, \pi_{k+1})$ is an edge of $H'$. Since from $\pi_{k+1}$ Player $\neg i$ can enforce Player $i$ to go to $\pi_n$ (since there are only positions owned by Player $\neg i$ along $\pi_{k+1} \ldots \pi_{n-1}$), we have $\text{minmax}^H_i(\pi_{k+1}) \geq \text{minmax}^H_i(\pi_n) > \text{minmax}^H_i$. Since $\pi_k \in S'_i$, this contradicts the definition of $H'$ (and $D(H)$), because the edge $(\pi_k, \pi_{k+1})$ would have been removed.

Thus $\text{minmax}^H_i(\lambda_i) = \text{minmax}^H_i$.

Proposition 7

Proof. Proof by induction on $j$.

If $j = 1$, we have $\text{reg}^{G'}_1 = \text{reg}^{D(G)} = \text{reg}^{G'}_1$ and by Lemma 3 $P_1 = \Lambda_i(D(G))$. 

16
We assume that \( \text{reg}_{i}^{G, j} = \text{reg}_{i}^{D^{i-1}(G)} \) and \( P_{i}^{j} = \Lambda_{i}(D^{i}(G)) \). By definition, \( \text{reg}_{i}^{G, j+1} = \text{reg}_{i}^{G, P_{i}^{j}, P_{i}^{j-1}} \). By induction hypothesis, \( P_{i}^{j} = \Lambda_{i}(D^{i}(G)) \), thus \( \text{reg}_{i}^{G, P_{i}^{j}, P_{i}^{j-1}} = \text{reg}_{i}^{D^{i}(G)} = \text{reg}_{i}^{G, j+1} \).

Moreover, \( \Lambda_{i} \in P_{i}^{j+1} \) iff \( \Lambda_{i} \in P_{i}^{j} \) and \( \text{reg}_{i}^{G, j+1}(\Lambda_{i}) = \text{reg}_{i}^{G, j+1}(\Lambda_{i}) \). By induction hypothesis, \( P_{i}^{j} = \Lambda_{i}(D^{i}(G)) \). We demonstrated that \( \text{reg}_{i}^{G, j+1} = \text{reg}_{i}^{D^{i}(G)} \). By Lemma \ref{lemma:regret_property} applied to the tree \( D^{i}(G) \), we have \( \text{reg}_{i}^{D^{i}(G)}(\Lambda_{i}) = \text{reg}_{i}^{D^{i}(G)}(\Lambda_{i}, \Lambda_{i-1}) = \max_{\Lambda_{i-1} \in \Lambda_{i}(D^{i}(G))} \left( u_{G}^{i}(\Lambda_{i}, \Lambda_{i-1}) - \min_{\Lambda_{i} \in \Lambda_{i}(D^{i}(G))} u_{G}^{i}(\Lambda_{i}^{*}, \Lambda_{i-1}) \right) \).

Since \( P_{i}^{j} = \Lambda_{i}(D^{i}(G)) \), we have \( \text{reg}_{i}^{D^{i}(G)}(\Lambda_{i}) = \text{reg}_{i}^{G, j+1}(\Lambda_{i}) \). So \( \text{reg}_{i}^{G, j+1}(\Lambda_{i}) = \text{reg}_{i}^{D^{i}(G)}(\Lambda_{i}) \). Consequently, \( \Lambda_{i} \in P_{i}^{j+1} \) iff \( \Lambda_{i} \in \Lambda_{i}(D^{i+1}(G)) \).

\[ \square \]

### 7.4 Missing Proofs of Section 5.2

In this section, we prove several lemmas that do not appear in the paper, especially to prove Lemma \ref{lemma:regret_property}.

**Lemma \[\text{[11]}\]**

**Proof.** Suppose that there is no winning strategy for both players in \( G \). Therefore \( \text{reg}_{i}^{G} = \text{reg}_{j}^{G+1} = \text{reg}_{j}^{G+1} = +\infty \). \( P_{i}^{j} = \Lambda_{i}(G) \) and \( P_{i}^{j} = \Lambda_{j}(G) \). It is easy to verify that there is no \( j \)-winning strategy for both players and all ranks \( j \).

Suppose that Player \( i \) has a winning strategy, for some \( i = 1, 2 \). Therefore by Lemma \ref{lemma:no_win} the strategies minimizing the regret are bounded by \( 2M^{2}\mid S \mid \). Since \( S_{j}^{i} \leq S_{0}^{i} = \Lambda_{i}(G) \) for all \( j \geq 0 \), we get that all strategy of \( S_{j}^{i} \) is bounded by \( 2M^{2}\mid S \mid \). Let \( j \geq 0 \), \( \Lambda_{i} \in S_{j}^{i} \) and \( \Lambda_{i} \in \Lambda_{i}(G) \). Let \( \pi = \text{Out}^{G, i}_{C_{i}}(\Lambda_{i}, \Lambda_{i}) \). We have \( \mu_{i}(\pi) \leq 2M^{2}\mid S \mid \), and since the weights are strictly positive integers, \( \mid \pi \mid \leq 2M^{2}\mid S \mid \). Therefore \( \mu_{i}(\pi) \leq 2(2M^{2})^{2}\mid S \mid \). In other words, for all \( j \geq 0 \), all strategy of \( P_{i}^{j} \) is \( j \)-winning and \( j \)-bounded by \( 2(2M^{2})^{2}\mid S \mid \).

It remains to prove that the \( j \)-winning strategies of Player \( i \) minimizing the \( (j + 1) \)-th regret are also \( j \)-bounded. Let \( j_{0} \geq 0 \) be the first natural number such that \( P_{i}^{j_{0}} \) contains a \( j_{0} \)-winning strategy (if it exists). If \( j_{0} = 0 \), then \( P_{i}^{j_{0}} = \Lambda_{i}(G) \). If \( j_{0} > 0 \), then no strategy of \( P_{i}^{j_{0} - 1} \) is \((j_{0} - 1)\)-winning by definition of \( j_{0} \), so that \( \text{reg}_{i}^{G, j_{0}} = +\infty \), from which we get \( P_{i}^{j_{0}} = \Lambda_{i}(G) \). In both cases, we have \( P_{i}^{j_{0}} = \Lambda_{i}(G) \).

Since after reaching her objective, Player \( i \) can play however she wants without affecting her regret, there is a strategy \( \gamma_{i} \in \Lambda_{i}(G) \) that wins against all strategies of \( P_{i}^{j_{0}} \) and which is memoryless once Player \( i \) has reached his objective. Formally, there is a memoryless strategy \( \gamma_{i} : S_{i} \rightarrow S \) such that for all \( \pi \in P_{j}(G) \) such that last(\( \pi \)) \( \in S_{i} \), if \( \pi \) contains a position of \( C_{i} \), then \( \gamma_{i}(\pi) = \gamma_{i}(\text{last}(\pi)) \).

Let \( \Lambda_{i} \in P_{i}^{j_{0}} \). We now bound the size of \( \text{Out}^{G, C_{i}}(\gamma_{i}, \Lambda_{i}) \), which will provide a bound on the utility. Let \( \pi_{i} = \text{Out}^{G, C_{i}}(\gamma_{i}, \Lambda_{i}) \) and \( \pi_{i} = \text{Out}^{G, C_{i}}(\gamma_{i}, \Lambda_{i}) \). We consider two cases:

- if \( \pi_{i} \) is a prefix of \( \pi_{i} \). We already know that \( \Lambda_{i} \) is \( j \)-bounded by \( 2(2M^{2})^{2}\mid S \mid \), therefore we also get \( \mu_{i}(\pi_{i}) \leq 2(2M^{2})^{2}\mid S \mid \), for all \( \kappa = 1, 2 \);

- if \( \pi_{i} \) is a prefix of \( \pi_{i} \), then \( \pi_{i} = \pi_{i} \pi_{i} \). Since \( \Lambda_{i} \) is \( j \)-bounded by \( 2(2M^{2})^{2}\mid S \mid \), \( \mu_{i}(\pi_{i}) \leq 2(2M^{2})^{2}\mid S \mid \), for all \( \kappa = 1, 2 \). Since \( \gamma_{i} \) is memoryless after \( \pi_{i} \), there is no loop in \( \pi_{i} \). Therefore \( \mu_{i}(\pi_{i}) \leq 3(2M^{2})^{2}\mid S \mid \), for all \( \kappa = 1, 2 \). Finally, for all \( \kappa = 1, 2 \), we get \( \mu_{i}(\pi_{i}) \leq 3(2M^{2})^{2}\mid S \mid \).

In both cases, we get \( \mu_{i}(\gamma_{i}, \Lambda_{i}) \leq 3(2M^{2})^{2}\mid S \mid \), for all \( \kappa = 1, 2 \) and all \( \Lambda_{i} \in P_{i}^{j_{0}} \). Therefore 0 \( \leq \mu_{i}(\pi_{i}) \leq 3(2M^{2})^{2}\mid S \mid \). (**) holds for all \( \Lambda_{i} \in P_{i}^{j_{0}} \). Also get \( \text{reg}_{i}^{G, j_{0} \rightarrow 1} \leq \text{reg}_{i}^{G, j_{0} \rightarrow 1} \leq 3(2M^{2})^{2}\mid S \mid \).

Let now \( \Lambda_{i} \) which minimizes \( \text{reg}_{i}^{G, j_{0} \rightarrow 1} \) and \( \Lambda_{i} \in P_{i}^{j_{0}} \). Let \( \pi = \text{Out}^{G, C_{i}}(\gamma_{i}, \Lambda_{i}) \). By (**) we have \( \text{reg}_{i}^{G, j_{0} \rightarrow 1} \leq 3(2M^{2})^{2}\mid S \mid \), ie

\[ \mu_{i}(\pi) = \text{br}_{i}^{G, j_{0} \rightarrow 1}(\Lambda_{i}) \leq 2(2M^{2})^{2}\mid S \mid \]

Since \( P_{i}^{j_{0}} = \Lambda_{i}(G), \text{br}_{i}^{G, j_{0} \rightarrow 1}(\Lambda_{i}) = \text{br}_{i}^{G, \Lambda_{i}}(\Lambda_{i}) \). Therefore by (**) we get \( \text{br}_{i}^{G, j_{0} \rightarrow 1}(\Lambda_{i}) \leq 3(2M^{2})^{2}\mid S \mid \), and \( \mu_{i}(\pi) \leq 6(2M^{2})^{3}\mid S \mid \). The weights being strictly positive, we get \( \mu_{i}(\pi) \leq 6(2M^{2})^{3}\mid S \mid = b_{G}^{i} \).
Lemma 15. Let $i = 1, 2$, $j \geq 0$, $\lambda_i \in P^i_j(G)$ and $\lambda_{-i} \in P^i_{-j}(G)$. Let $o = \text{Out}^G(\lambda_i, \lambda_{-i})$. If $o$ is winning for Player $i$, $\mu_i(o) \leq B^G$ and $\mu_{-i}(o) \leq B^G$, then $u^G_i(\lambda_i, \lambda_{-i}) = \varphi^G_i(\Phi(\lambda_i), \Phi(\lambda_{-i}))$.

Proof. If $u^G_i(\lambda_1, \lambda_2) \leq B$, then $\text{Out}^G(\lambda_1, \lambda_2)$ is winning, and we let $\pi = s_0 \ldots s_n = \text{Out}^G_{\lambda_1, \lambda_2}(\lambda_1, \lambda_2)$. We let $\pi^r = (\pi_0) \ldots (\pi_n)$ where for all $k \leq n$, $\pi_k = s_0 \ldots s_k$. Since $\pi$ is bounded, $\pi_k \in P_{BC}(G')$. By definition of $\Phi$, we clearly have $\pi^r = \text{Out}^G_{\Phi(\lambda_1), \Phi(\lambda_2)}$, from which we get $u^G_i(\lambda_1, \lambda_2) = \varphi^G_i(\Phi(\lambda_1), \Phi(\lambda_2))$.

If $\text{Out}^G(\Phi(\lambda_1), \Phi(\lambda_2))$ is winning for Player 1, we let $\pi^r = \text{Out}^G_{\Phi(\lambda_1), \Phi(\lambda_2)}(\Phi(\lambda_1), \Phi(\lambda_2))$ and $\pi = \text{last}(\pi_0) \ldots \text{last}(\pi_n)$ where $\pi^r = \pi_0 \ldots \pi_n$. Clearly, $\pi$ is a winning play for Player 1 in $G$ and by definition of $\Phi$, $\pi = \text{Out}_{\Phi(G_1)}(\lambda_1, \lambda_2)$, from which we get the equality of the utilities.

Lemma 16. For all $i = 1, 2$, all $j \geq 0$. If $\Phi(P^i_j(G)) = P^i_j(G')$ and there is a strategy $j$-bounded by $b^G$ in $P^i_j(G)$, then for all $\lambda_{-i} \in P^i_{-j}(G)$, $\text{br}^G_i(P^i_j(G))(\lambda_{-i}) = \text{br}^G_i(P^i_{-j}(G'))(\Phi(\lambda_{-i}))$.

Proof. Let $n_i \in P^i_j(G)$ be a strategy $j$-bounded by $b^G$ (it exists by hypothesis), and let $\lambda_{-i} \in P^i_{-j}(G)$. Since $n_i$ is $j$-bounded, it is $j$-winning and by Lemma 15 $u^G_i(n_i, \lambda_{-i}) = u^G_i(\Phi(n_i), \Phi(\lambda_{-i}))$. Therefore $u^G_i(\Phi(\Phi(n_i), \Phi(\lambda_{-i})) < +\infty$.

Let $\lambda_i \in P^i_j(G)$ which minimizes $\text{br}^G_i(P^i_j(\lambda_{-i}))$. Let $\pi = \text{Out}_{\Phi(G_1)}(\lambda_i, \lambda_{-i})$. We have $\mu_i(\pi) \leq u^G_i(\pi, \lambda_{-i}) \leq b^G$. Since the weights are strictly positive integers, $|\pi| \leq b^G$, and therefore $\mu_{-i}(\pi) \leq b^G - M^G = B^G$. By Lemma 15, we get $u^G_i(\lambda_i, \lambda_{-i}) = u^G_i(\Phi(\lambda_i), \Phi(\lambda_{-i}))$. Therefore $u^G_i(\lambda_i, \lambda_{-i}) = \text{br}^G_i(P^i_j(\lambda_{-i}))(\lambda_{-i}) \geq \text{br}^G_i(P^i_{-j}(G'))(\Phi(\lambda_{-i}))$.

Conversely, let $\lambda'_i \in P^i_j(G)$ which minimizes $\text{br}^G_i(P^i_j(\Phi(\lambda_{-i})))$. Therefore $u^G_i(\lambda'_i, \Phi(\lambda_{-i})) \leq u^G_i(\Phi(\pi), \Phi(\lambda_{-i})) < +\infty$. Since $\Phi(P^i_j(G)) = P^i_j(G)$ by hypothesis, there exists $\lambda_i \in P^i_j(G)$ such that $\Phi(\lambda_i) = \lambda'_i$. Since $u^G_i(\lambda'_i, \Phi(\lambda_{-i}))$ is finite, $\text{Out}^G_i(\lambda'_i, \Phi(\lambda_{-i}))$ is winning for Player $i$. It is easy to see that $\text{Out}^G_{G_i, G_{-i}}(\lambda_i, \lambda_{-i}) = \text{last}(\pi_0) \ldots \text{last}(\pi_n)$ where $\pi_0 \ldots \pi_n = \text{Out}^G_{G_i, G_{-i}}(\lambda'_i, \Phi(\lambda_{-i}))$ and that they both have the same utility, i.e. $u^G_i(\lambda_i, \lambda_{-i}) = u^G_i(\lambda'_i, \Phi(\lambda_{-i}))$. Since $u^G_i(\lambda'_i, \Phi(\lambda_{-i})) = \text{br}^G_i(P^i_j(G'))(\Phi(\lambda_{-i}))$, we get $\text{br}^G_i(P^i_j(G'))(\Phi(\lambda_{-i})) \geq \text{br}^G_i(P^i_{-j}(G'))(\Phi(\lambda_{-i}))$.

Lemma 17. For all $j \geq 0$, $i = 1, 2$, $\lambda_i \in P^i_j(G)$. If $\Phi(P^i_j(G)) = P^i_j(G')$, then $\lambda_i$ is $j$-bounded by $B^G$ iff $\Phi(\lambda_i)$ is $j$-winning. If $\lambda_i$ is $j$-bounded by $B^G$, then for all $\lambda_{-i} \in S^i_j(G)$, $u^G_i(\lambda_i, \lambda_{-i}) = u^G_i(\Phi(\lambda_i), \Phi(\lambda_{-i}))$.

Proof. If $\lambda_i \in \Lambda_i(G)$ is $j$-bounded by $B^G$. Then let $\lambda_i \in P^i_j(G')$. Since $\Phi(P^i_j(G)) = P^i_j(G')$, there exists $\lambda'_{-i} \in P^i_{-j}(G)$ such that $\Phi(\lambda_{-i}) = \lambda'_{-i}$. Since $\lambda_i$ is $j$-bounded, we are in the condition of Lemma 15 therefore $u^G_i(\lambda_i, \lambda_{-i}) = u^G_i(\Phi(\lambda_i), \lambda'_{-i}) < +\infty$. Therefore $\Phi(\lambda_i)$ wins against $\lambda'_{-i}$.

Conversely, if $\Phi(\lambda_i)$ is $j$-winning, then let $\lambda_i \in P^i_j(G')$. By hypothesis, $\Phi(\lambda_i) \in P^i_j(G')$. Therefore $\Phi(\lambda_i)$ wins against $\Phi(\lambda_{-i})$. Let $\pi_0 \ldots \pi_n = \text{Out}^G_{G_i, G_{-i}}(\Phi(\lambda_i), \Phi(\lambda_{-i}))$. Clearly, by definition of $\Phi$ and $G'$, $\lambda_i$ wins against $\lambda'_{-i}$ and $\text{Out}^G_{G_i, G_{-i}}(\lambda_i, \lambda_{-i}) = (\pi_0) \ldots \text{last}(\pi_n)$.

Lemma 18. $\forall j \geq 0, \forall i = 1, 2$, if $\Phi(P^i_j(G)) = P^i_j(G')$ then:

(i) $\text{reg}^G_i(j+1) = +\infty \iff \text{reg}^G_i(j+1) = +\infty$

(ii) $\forall \lambda_i \in P^i_{j+1}(G) \cup \Phi^{-1}(P^i_{j+1}(G'))$,

\[ \text{reg}^G_i(j+1)(\lambda_i) = \text{reg}^G_i(j+1)(\Phi(\lambda_i)) \]

(iii) $\Phi(P^i_{j+1}(G)) = P^i_{j+1}(G')$.

Proof. (i) If $\text{reg}^G_i(j+1) < +\infty$, then it means that there is a $j$-winning strategy $\lambda_i \in P^i_j(G)$. By Lemma 17, $\Phi(\lambda_i)$ is $j$-winning. Since by hypothesis, $\Phi(P^i_j(G)) = P^i_j(G')$, $\Phi(\lambda_i) \in P^i_j(G')$, so that $\text{reg}^G_i(j+1) < \text{reg}^G_i(j+1)(\Phi(\lambda_i)) < +\infty$.

Conversely, if $\text{reg}^G_i(j+1) < +\infty$, there is a $j$-winning strategy $\lambda'_i \in P^i_j(G')$. By hypothesis, there is $\lambda_i \in P^i_j(G)$ such that $\Phi(\lambda_i) = \lambda'_i$, and by Lemma 17, $\lambda_i$ is $j$-bounded, and in particular $j$-winning. Therefore $\text{reg}^G_i(j+1) < +\infty$. 18
(ii) The proof is in two parts, depending on whether \( \text{reg}_{G,j+1}^i \) is finite or not.

(iii).a If \( \text{reg}_{G,j+1}^i = +\infty \), then by (i), \( \text{reg}_{G,j+1}^{i'} = +\infty \).

Let \( \lambda_i \in P_j^{i+1}(G) \). Since \( P_j^{i+1}(G) \subseteq P_j^i(G) \), \( \lambda_i \in P_j^i(G) \). By hypothesis, \( \Phi(P_j^i(G)) = P_j^i(G') \), therefore \( \Phi(\lambda_i) \in P_j^i(G') \). Therefore \( \text{reg}_{G,j+1}^i(\Phi(\lambda_i)) = +\infty = \text{reg}_{G,j+1}^{i'}(\lambda_i) \).

If \( \lambda_i \in \Phi^{-1}(P_j^{i+1}(G')) \), then \( \text{reg}_{G,j+1}^i(\Phi(\lambda_i)) = +\infty \). Since \( P_j^{i+1}(G'') \subseteq P_j^i(G') \), \( \Phi(\lambda_i) \in P_j^i(G') \). By hypothesis, \( \Phi(P_j^i(G)) = P_j^i(G') \), therefore \( \lambda_i \in P_j^i(G') \), and \( \text{reg}_{G,j+1}^{i'}(\lambda_i) = +\infty = \text{reg}_{G,j+1}^i(\Phi(\lambda_i)) \), since \( \text{reg}_{G,j+1}^{i'}(\lambda_i) = +\infty \).

(iii).b If \( \text{reg}_{G,j+1}^i < +\infty \), then by (i), \( \text{reg}_{G,j+1}^{i'} < +\infty \). Let \( \lambda_i \in P_j^{i+1}(G) \cup \Phi^{-1}(P_j^{i+1}(G')) \). We prove that for all \( \lambda_i \in P_j^i(G) \),

1. \( u_i^G(\lambda_i, \lambda_{i-1}) = u_i^G(\Phi(\lambda_i), \Phi(\lambda_{i-1})) \)
2. \( b_i^{G,P_j^i(G)}(\lambda_{i-1}) = b_i^{G,P_j^i(G')}(\Phi(\lambda_{i-1})) \)

For 1, If \( \lambda_i \in P_j^{i+1}(G) \), then since \( \text{reg}_{G,j+1}^i < +\infty \), by Lemma 11, \( \lambda_i \) is \( j \)-bounded by \( b^G \), and \textit{a fortiori} by \( b^{G'} \). By Lemma 17 we get the result. If \( \lambda_i \in \Phi^{-1}(P_j^{i+1}(G')) \), then since \( \text{reg}_{G,j+1}^{i'} < +\infty \), \( \Phi(\lambda_i) \in P_j^i(G') \) is necessarily winning. By Lemma 17, \( \lambda_i \) is \( j \)-bounded by \( b^{G'} \) and again by the same lemma, we get the result.

For 2, since \( \text{reg}_{G,j+1}^{i'} < +\infty \), by Lemma 11 there is a strategy of Player \( i \) \( j \)-bounded by \( b^{G'} \) in \( P_j^i(G') \). Therefore we can apply Lemma 11 and we get the result.

Finally we have:

\[
\text{reg}_{G,j+1}^i(\lambda_i) = \max_{\lambda_i \in P_j^i(G)} \left[ u_i^G(\lambda_i, \lambda_{i-1}) - b_i^{G,P_j^i(G)}(\lambda_{i-1}) \right]
\]

\[
= \max_{\lambda_i \in P_j^i(G)} \left[ u_i^G(\Phi(\lambda_i), \Phi(\lambda_{i-1})) - b_i^{G,P_j^i(G)}(\Phi(\lambda_{i-1})) \right]
\]

(by (1) and (2))

\[
= \max_{\lambda_i \in P_j^i(G')} \left[ u_i^G(\Phi(\lambda_i), \lambda_{i-1}) - b_i^{G',P_j^i(G')}(\lambda_{i-1}) \right]
\]

(since \( \Phi(P_j^i(G)) = P_j^i(G') \) by hypothesis)

\[
= \text{reg}_{G,j+1}^i(\Phi(\lambda_i))
\]

(iii) Let \( i \in \{1,2\} \) and \( \lambda_i \in S_{j+1}^i(G) \). Suppose that \( \Phi(\lambda_i) \not\in S_{j+1}^i(G') \). It means that \( \Phi(\lambda_i) \) does not minimize the \( j + 1 \)-th regret in \( G' \). Therefore there exists another strategy \( \lambda_i' \in P_j^{i+1}(G') \) such that \( \text{reg}_{G,j+1}^i(\lambda_i') < \text{reg}_{G,j+1}^i(\Phi(\lambda_i)) \). By (i), we get \( \text{reg}_{G,j+1}^i(\lambda_i') < \text{reg}_{G,j+1}^i(\lambda_i) \), for all \( \lambda_i \in \Phi^{-1}(\lambda_i') \). Since \( P_j^{i+1}(G') \subseteq P_j^i(G') \) and \( \Phi(P_j^i(G)) = P_j^i(G') \), we have \( \Phi^{-1}(\lambda_i') \subseteq P_j^i(G) \), and we get a contradiction on the minimality of \( \lambda_i \).

Conversely, let \( \lambda_i' \in P_j^{i+1}(G') \). Suppose that \( \lambda_i' \not\in \Phi(P_j^{i+1}(G)) \). Since \( \lambda_i' \in P_j^i(G) \), by hypothesis, \( \lambda_i' \in \Phi(P_j^i(G)) \). Therefore there exists \( \lambda_i \in P_j^i(G) \) such that \( \Phi(\lambda_i) = \lambda_i' \), but \( \lambda_i \not\in P_j^{i+1}(G) \). It means that \( \lambda_i \) did not survive to the \( j \)-th iteration. In other words, for all strategy \( \gamma_i \in P_j^{i+1}(G) \), \( \text{reg}_{G,j+1}^i(\lambda_i) < \text{reg}_{G,j+1}^{i'}(\lambda_i) \). Since \( \lambda_i \in \Phi^{-1}(P_j^{i+1}(G')) \), by (ii) we have \( \text{reg}_{G,j+1}^{i'}(\lambda_i) = \text{reg}_{G,j+1}^{i'}(\Phi(\lambda_i)) = \text{reg}_{G,j+1}^i(\lambda_i) \). By (ii), we also have \( \text{reg}_{G,j+1}^{i'}(\lambda_i) = \text{reg}_{G,j+1}^{i'}(\Phi(\lambda_i)) \). Therefore \( \text{reg}_{G,j+1}^{i'}(\Phi(\lambda_i)) < \text{reg}_{G,j+1}^{i'}(\Phi(\lambda_i)) \). Since \( \gamma_i \in P_j^i(G) \) and by hypothesis, \( \Phi(P_j^i(G)) = P_j^i(G') \), we have \( \Phi(\gamma_i) \in P_j^i(G') \). Therefore we get a strategy \( \Phi(\gamma_i) \) of \( P_j^i(G') \) with a lower \( (j + 1) \)-th regret than the \( (j + 1) \)-regret of \( \lambda_i \). This is in contradiction with \( \lambda_i' \in P_j^{i+1}(G') \).

\[\blacksquare\]

Lemma 13

Proof. Clearly, \( \Phi(\Lambda_i(G)) = \Lambda_i(G') \). Therefore we can apply Lemma 18 (proved in Appendix) so that items (i), (ii) and (iii) holds at rank 0. In particular, \( \Phi(P_j^1(G)) = P_j^1(G') \). Therefore we can again apply Lemma 18 at rank 1. More generally, for all \( j \geq 1 \), we have:

1. for all \( \lambda_i \in P_j^i(G) \), \( \text{reg}_{G,j}^i(\lambda_i) = \text{reg}_{G,j}^i(\Phi(\lambda_i)) \);
2. \( \Phi(P_j^i(G)) = P_j^i(G') \).

\[\blacksquare\]