To Reach or not to Reach?
Efficient Algorithms for Total-Payoff Games

Thomas Brihaye¹, Gilles Geeraerts², Axel Haddad¹ and Benjamin Monmege²

¹ Université de Mons, Belgium
thomas.brihaye,axel.haddad@umons.ac.be
² Université libre de Bruxelles, Belgium
gilles.geeraerts,benjamin.monmege@ulb.ac.be

Abstract. Quantitative games are two-player zero-sum games played on directed weighted graphs. We consider variants of usual payoff functions: total-payoff, mean-payoff, discounted-payoff, where we add a reachability objective, i.e., Player 1 wants to reach a target while minimising his payoff. In the mean-payoff and discounted-payoff cases, we show that reachability and classical versions of the game are inter-reducible. In the case of reachability total-payoff, we introduce an efficient value iteration algorithm to compute the values and optimal strategies (when they exist), that runs in pseudo-polynomial time. This contribution allows us to derive the first (to our knowledge) value iteration algorithm for classical total-payoff games, running in pseudo-polynomial time.

1 Introduction

A classical and highly relevant problem in graph theory is the shortest path problem, which asks to compute the shortest paths from all nodes in a directed weighted graph to a designated target. When all weights are non-negative, Dijkstra’s algorithm is a well-known polynomial time algorithm to solve this problem. A natural extension of the shortest path problem consists in partitioning the vertices between two antagonistic players, where the goals of Player 1 and 2 are respectively to find a shortest and a longest path. A polynomial time extension of Dijkstra’s algorithm has been proposed by Khachiyan et al. [9] for this problem.

The shortest path problem becomes more complex when edges may have arbitrary weights (positive and negative). In the one-player case, Floyd-Warshall’s algorithm computes all shortest paths (or detects negative cycles) in polynomial time. The two-player case is quoted as an open problem in [9]. Partial results can be found in [5], where determining whether Player 1 can guarantee to reach the target with an accumulated negative weight is shown in NP \cap co-NP. One of the contributions of the present work is an efficient (pseudo-polynomial time) algorithm for the two-player, arbitrary weights, shortest path problem.

A natural way to formalise the two-player shortest path problem is as a two-player zero-sum game played on a weighted graph, where the goals of the
players combine a qualitative objective (reachability of the target) and a quantitative one (minimising the weight of the path). These two classes of objectives have been extensively studied separately. In zero-sum qualitative games (such as games with $\omega$-regular objectives), it is natural to look for a winning strategy ensuring Player 1 to fulfil his goal, whatever Player 2 does. On the other hand, for quantitative games (such as total-payoff [7], mean-payoff [3] and discounted-payoff [13] games), the notion of winning strategy does not make sense anymore, since in this quantitative setting Player 1 aims at minimising the payoff while Player 2 tries to maximise it. Instead, we want to compute, for each player, the best payoff (either the minimal or maximal one for Player 1 and 2 respectively) that he can guarantee from each vertex. A strategy is then called optimal if it ensures the optimal payoff no matter what the other player is doing.

There are two conventional ways to express the difficulty of games. The former measures the complexity of the winning or optimal strategies. For several classes of qualitative games, including those with reachability winning conditions, it is well-known that memoryless strategies (that depend only on the current vertex) are sufficient. This result carries on to the quantitative games with total-, mean- or discounted-payoff [7]. Another measure is given by the computational complexity of calculating winning strategies (for qualitative games) or the optimal payoffs and strategies (for quantitative games). The existence of memoryless winning/optimal strategies in the aforementioned cases yields an $\text{NP} \cap \text{co-NP}$ upper-bound. Nevertheless, variants of the iterative backward induction technique are often more efficient, both from theoretical and practical points of view. For instance, the attractor technique [12] is a polynomial time algorithm for reachability games. Also, value iteration algorithms [13] with pseudo-polynomial time complexities (i.e., polynomial if weights of the arena, respectively, the discount factor, is encoded in unary) exist for mean-, respectively, discounted-payoff games. As far as we know, no such efficient algorithm exists for total-payoff games, where one attempt is the exponential time policy iteration algorithm of [6].

Note that the combination of qualitative and quantitative objectives has been considered before. Apart from [5] that we have already mentioned, Chatterjee et al. combine in [2] parity winning conditions with mean-payoff. They show that optimal strategies may require infinite memory, and characterise precisely the complexity of solving those games.

In this paper, we consider total-, mean- and discounted-payoff games with reachability objectives, and introduce efficient (pseudo-polynomial time) algorithms to solve them. We first consider, in Section 3, reachability mean- and discounted-payoff games, and show that they are logarithmic space inter-reducible with classical mean- and discounted-payoff games (i.e., without reachability objectives). Since pseudo-polynomial algorithms exist for the mean- and discounted-payoff games, we obtain pseudo-polynomial time algorithms for mean- and discounted-payoff games with reachability objectives. Since no efficient algorithm exists for total-payoff games, we introduce, in Section 4, a value iteration algorithm for reachability total-payoff games that runs in pseudo-polynomial time. We charac-
terise the optimal strategies of both players: while Player 1 may need memory, Player 2 always has a memoryless optimal strategy. We introduce acceleration heuristics to improve the running time of the algorithm in practice. Finally, in Section 5 we rely on a pseudo-polynomial time translation from total-payoff games to reachability total-payoff games to deduce a pseudo-polynomial time algorithm for total payoff-games. Our acceleration techniques still apply in that case, producing a nontrivial class of total-payoff games that we are able to solve in polynomial time.

2 (Reachability) quantitative games

In this paper, vectors indexed by a set \( V \) will alternatively be denoted by \((x_v)_{v \in V}\) as usual, or as mappings \( x \) associating every index \( v \) to its value \( x(v) \). The set of vectors indexed by \( V \) with values in set \( S \) are denoted by \( S^V \). We denote by \( \mathbb{Z} \) the set of integers, \( \mathbb{Z}_\infty = \mathbb{Z} \cup \{-\infty, +\infty\} \), \( \mathbb{R} \) the set of real numbers and \( \mathbb{R}_\infty = \mathbb{R} \cup \{-\infty, +\infty\} \). We denote by \( \preceq \) the pointwise order over vectors from \( \mathbb{R}^V_\infty \), where \( x \preceq y \) if and only if \( x(v) \leq y(v) \) for all \( v \in V \).

2.1 Games over weighted graphs

A weighted graph is a tuple \( \langle V, E, \omega \rangle \) where \( V = V_1 \sqcup V_2 \) is a finite set of vertices partitioned into the sets \( V_1 \) and \( V_2 \) of Player 1 and 2 respectively, \( E \subseteq V \times V \) is a set of directed edges, \( \omega: E \rightarrow \mathbb{Z} \) is the weight function, associating an integer weight with each edge. For every vertex \( v \in V \), the set of successors of \( v \) by \( E \) is denoted by \( E(v) = \{ v' \in V \mid (v, v') \in E \} \). For the sake of exposure, we assume that every graph is deadlock free, i.e., for all vertex \( v \), there exists \( v' \in V \) such that \( (v, v') \in E \). Finally, throughout this paper, we let \( W = \max_{(v, v') \in E} |\omega(v, v')| \) be the greatest edge weight (in absolute value) in a graph.

A history or finite play is a finite path, i.e., a finite sequence of vertices \( \pi = v_0v_1 \cdots v_k \) such that for all \( i < k \), \( (v_i, v_{i+1}) \in E \). A play is an infinite path, i.e., an infinite sequence of vertices \( \pi = v_0v_1 \cdots \) such that every finite prefix \( v_0 \cdots v_k \) is a history. For all \( k \geq 0 \), we denote by \( \pi[k] \) the prefix \( v_0v_1 \cdots v_k \) of \( \pi \).

Let us now recall the notion of game played on a weighted graph (see, e.g., [4] for more details). It is given by a tuple \( \langle V, E, \omega, P \rangle \), where \( \langle V, E, \omega \rangle \) is a weighted graph and \( P \) maps every play or finite play to its cost in \( \mathbb{R}_\infty \). Intuitively, the players play by moving a token along the edges of the game, Player \( i \) moving the token when it is on a vertex from \( V_i \). The game goes on for ever, hence building an infinite path.

In this work, we focus on two classes of games over weighted graphs, namely: purely quantitative games, where the cost of a play depends only on the sequence of visited weights; and reachability quantitative games, where the cost depends on the sequence of visited weights and a reachability objective on the

---

3 If the weights are rationals, one always can, in the cases we consider, multiply the weights by their greatest common denominator to obtain a corresponding game with integer weights.
sequence of visited vertices. Let us now define formally the classical payoff functions we rely upon in this paper. Let \( \pi = v_0v_1 \cdots v_k \) be a finite play. Then: the total-payoff \( TP \) consists in summing up the weights along \( \pi \), i.e., \( TP(\pi) = \sum_{i=0}^{k-1} \omega(v_i, v_{i+1}) \); the mean-payoff \( MP \) computes the average weight of \( \pi \), i.e., \( MP(\pi) = \frac{1}{k} \sum_{i=0}^{k-1} \omega(v_i, v_{i+1}) \); and finally the \( \lambda \)-discounted-payoff \( DP_\lambda \) computes the sum of the weights discounted by a factor \( \lambda \in (0, 1) \), i.e., \( DP_\lambda(\pi) = (1 - \lambda) \sum_{i=0}^{k-1} \lambda^i \omega(v_i, v_{i+1}) \).

We can now lift these definitions from finite plays to infinite plays \( \pi' = v_0v_1 \cdots \). In the purely quantitative case, we let \( TP(\pi) = \liminf_{k \to \infty} TP(\pi[k]) \); \( MP(\pi) = \liminf_{k \to \infty} MP(\pi[k]) \) and \( DP_\lambda(\pi) = \lim_{k \to \infty} DP_\lambda(\pi[k]) \). In the reachability quantitative case, the cost of a play depends on a fixed set of target vertices \( T \subseteq V \). If a play avoids \( T \), its payoff is \(+\infty\); otherwise, the payoff is the one of the shortest prefix ending in the target. Formally, for a (purely quantitative) payoff mapping \( P \), we let \( TRP \) be the reachability payoff associated to \( P \), where for all play or finite play \( \pi = v_0v_1 \cdots \), \( TRP(\pi) = +\infty \) if there is no position \( j \) in \( \pi \) such that \( v_j \in T \); and otherwise, \( TRP(\pi) = P(\pi[i]) \) where \( i \) is the least position in \( \pi \) such that \( v_i \in T \). In particular, we can take \( P \in \{ TP, MP, DP_\lambda \} \), with \( \lambda \in (0, 1) \), and obtain immediately the definitions of the reachability total-, mean- and \( \lambda \)-discounted-payoff mappings \( TRTP, TRMP \) and \( TRDP_\lambda \).

### 2.2 Strategies and values

In this section, we fix a quantitative game \( G = (V, E, \omega, P) \). A strategy for Player \( i \) is a mapping \( \sigma : V^*V_i \to V \) such that for all sequences \( \pi = v_0 \cdots v_k \) with \( v_k \in V_i \), \( \langle v_k, \sigma(\pi) \rangle \in E \). Furthermore, a play or finite play \( \pi = v_0v_1 \cdots \) conforms to a strategy \( \sigma \) of Player \( i \) if for all \( k \) such that \( v_k \in V_i \), \( v_{k+1} = \sigma(\pi[k]) \).

A strategy \( \sigma \) for Player \( i \) \((i = 1, 2)\) is memoryless (or positional) if the images of all histories by \( \sigma \) only depend on their last vertex, i.e., if for all finite plays \( \pi, \pi' \), we have \( \sigma(\pi v) = \sigma(\pi' v) \) for all vertices \( v \in V_i \). A strategy \( \sigma \) for Player \( i \) \((i = 1, 2)\) is said to be finite-memory if there exists a finite set \( M \) (representing the memory of the strategy) with a special element \( m_0 \) (representing the initial content of the memory), a memory-update function up: \( M \times V \to M \), and a decision function dec: \( M \times V_i \to V \) such that for every finite play \( \pi \) and vertex \( v \in V_i \), \( \sigma(\pi v) = \text{dec}(\text{mem}(\pi v), v) \) where \( \text{mem}(\pi) \) is defined by induction on the length of the finite play \( \pi \) by

\[
\text{mem}(v_0) = m_0 \quad \text{and} \quad \text{mem}(\pi v) = \text{up}(\text{mem}(\pi), v).
\]

We call size of the strategy the cardinal \(|M|\) of the memory it uses.

For all strategies \( \sigma_i \) for Player 1, \( \sigma_2 \) for Player 2, for all vertices \( v \), we let Play\((v, \sigma_i)\) be the outcome of \( \sigma_i \) (from \( v \)), i.e., the set of plays that conform to \( \sigma_i \) and that start in \( v \). Similarly, we let Play\((v, \sigma_1, \sigma_2)\) be the outcome of \( \sigma_1 \) and \( \sigma_2 \), i.e., the unique play that conforms to \( \sigma_1 \) and \( \sigma_2 \) and that starts in \( v \). Then, for all vertices \( v \), we let Val\((v, \sigma_1, \sigma_2)\) = \( P(\text{Play}(v, \sigma_1, \sigma_2)) \) be the value of the profile of strategies \( (\sigma_1, \sigma_2) \) over \( v \).
We can now define the objectives of the two players. Intuitively, the objectives of Player 1 and Player 2 are respectively to minimize and maximize the payoff. Formally, we let $\text{Val}_G(v, \sigma_1)$ and $\text{Val}_G(v, \sigma_2)$ be the respective values of strategies $\sigma_1$ (of Player 1) and $\sigma_2$ (of Player 2), defined as:

$$\text{Val}_G(v, \sigma_1) = \sup_{\sigma_2} \text{Val}_G(v, \sigma_1, \sigma_2) = \sup_{\pi \in \text{Play}(v, \sigma_1)} P(\pi)$$

$$\text{Val}_G(v, \sigma_2) = \inf_{\sigma_1} \text{Val}_G(v, \sigma_1, \sigma_2) = \inf_{\pi \in \text{Play}(v, \sigma_2)} P(\pi).$$

Finally, for all vertices $v$, we let $\overline{\text{Val}}_G(v) = \sup_\sigma \text{Val}_G(v, \sigma_2)$ and $\underline{\text{Val}}_G(v) = \inf_\sigma \text{Val}_G(v, \sigma_1)$ be the lower and upper values of $v$ respectively. We may easily show that $\underline{\text{Val}}_G \leq \overline{\text{Val}}_G$. We say that strategies $\sigma_1^*$ of Player 1 and $\sigma_2^*$ of Player 2 are optimal if, for all vertices $v$: $\text{Val}_G(v, \sigma_1^*) = \overline{\text{Val}}_G(v)$ and $\text{Val}_G(v, \sigma_2^*) = \underline{\text{Val}}_G(v)$ respectively. We say that a game $G$ is determined if for all vertices $v$, its lower and upper values are equals. In that case we write $\text{Val}_G(v) = \overline{\text{Val}}_G(v) = \underline{\text{Val}}_G(v)$ and refer to it as the value of $v$. In the game is clear from the context, we may drop the index $G$ of all previous values. As a consequence of Martin’s theorem [10], any game equipped with one of the value mappings introduced above is determined.

**Theorem 1.** Total-payoff, mean-payoff and $\lambda$-discounted-payoff games, as well as their reachability variants, are determined.

**Proof.** Consider a quantitative game $G = (V, E, \omega, P)$ and a vertex $v \in V$. We will prove the determinacy result by using the Borel determinacy result of [10]. Indeed, for an integer $M$, consider $\text{Win}_M$ to be the set of infinite plays with a payoff less than or equal to $M$. Payoff mappings $\text{RTP}$, $\text{RMP}$ and $\text{RDP}_\lambda$ are easily shown to be Borel measurable since the set of plays with finite payoff is then a countable union of open sets of plays. For $\text{TP}$, $\text{MP}$, $\text{DP}_\lambda$, it is shown by considering a pointwise limit of Borel measurable functions. Therefore, for all these payoff mappings $P$, we know that $\text{Win}_M$ is a Borel set, so that the qualitative game defined over the graph $(V, E, \omega)$ with winning condition $\text{Win}_M$ is determined. We now use this preliminary result to show our determinacy result.

We first consider cases where lower or upper values are infinite. Suppose first that $\text{Val}(v) = -\infty$. We have to show that $\overline{\text{Val}}(v) = -\infty$ too. Let $M$ be an integer. From $\text{Val}(v) < M$, we know that for all strategy $\sigma_2$ of Player 2, there exists a strategy $\sigma_1$ for Player 1, such that $P(\text{Play}(v, \sigma_1, \sigma_2)) \leq M$. In particular, Player 2 has no winning strategy in the qualitative game equipped with $\text{Win}_M$ as a winning condition. By the previous determinacy result, we know that Player 1 has a winning strategy in that case, i.e., a strategy $\sigma_1$ such that every strategy $\sigma_2$ of Player 2 verifies $P(\text{Play}(v, \sigma_1, \sigma_2)) \leq M$. This exactly means that $\overline{\text{Val}}(v) \leq M$. Since this holds for every value $M$, we get that $\overline{\text{Val}}(v) = -\infty$. The proof goes exactly in a symmetrical way to show that $\overline{\text{Val}}(v) = +\infty$ implies $\overline{\text{Val}}(v) = -\infty$.

Consider then the case where both $\overline{\text{Val}}(v)$ and $\underline{\text{Val}}(v)$ are finite values. Suppose that $\overline{\text{Val}}(v) < \underline{\text{Val}}(v)$ and consider a real number $r$ strictly in-between those two values. From $r < \overline{\text{Val}}(v)$, we deduce that Player 1 has no winning strategy from $v$
in the qualitative game with winning condition \( \text{Win}_r \). Identically, from \( \text{Val}(v) < r \), we deduce that Player 2 has no winning strategy from \( v \) in the same game. This contradicts the determinacy of this qualitative game. Hence, \( \text{Val}(v) = \overline{\text{Val}}(v) \). □

2.3 Finding vertices with value \(+\infty\)

We show that, under certain conditions on the payoff \( \mathbf{P} \), the classical attractor technique (used for qualitative reachability, see [12]) allows us to determine when a vertex \( v \) has value \(+\infty\) in a game with the corresponding reachability payoff mapping \( \mathbf{RP} \). More precisely, let \( \mathcal{G} = (V, E, \omega, T_{\mathbf{RP}}) \) be a reachability quantitative game. We suppose that the payoff \( T_{\mathbf{RP}} \) verifies that for all play \( \pi = v_0v_1\cdots \), \( T_{\mathbf{RP}}(\pi) = +\infty \) if and only if \( v_i \notin T \) for all \( i \geq 0 \), i.e., \( \pi \) does not reach the target. We let \( V_{+\infty} = \{ v \in V \mid \text{Val}(v) = +\infty \} \), and we recall the definition of the attractor of a set of vertices \( T \), as the sequence \( \text{Attr}_0(T), \ldots, \text{Attr}_i(T), \ldots \)

\[
\text{Attr}_0(T) = T \\
\text{for all } i \geq 0 : \text{Attr}_{i+1}(T) = \text{Attr}_i(T) \cup \{ v \in V_1 \mid \exists (v, v') \in E \text{ } v' \in \text{Attr}_i(T) \} \\
\cup \{ v \in V_2 \mid \forall (v, v') \in E \text{ } v' \in \text{Attr}_i(T) \}.
\]

It is easy to see that, in a graph with \( n \) vertices, this sequence converges in at most \( n \) steps, i.e., for all \( k \geq 1 : \text{Attr}_k(T) = \text{Attr}_{k+1}(T) \). We denote by \( \text{Attr}(T) \) the set \( \text{Attr}_n(T) \). It is well-known that Player 1 has a memoryless strategy to ensure reaching a vertex in \( T \) from a vertex \( v \) if and only if \( v \in \text{Attr}(T) \). Hence:

**Proposition 2.** In all reachability quantitative game \( \mathcal{G} = (V, E, \omega, T_{\mathbf{RP}}) \) such that for all play \( \pi, T_{\mathbf{RP}}(\pi) = +\infty \) if and only if \( \pi \) does not reach the target \( T \), we have \( V_{+\infty} = V \setminus \text{Attr}(T) \). This set can be computed in polynomial time.

**Proof.** Let us first assume that \( v \in \text{Attr}(T) \), in this case, we know that Player 1 has a memoryless winning strategy from \( v \) to reach \( T \) (qualitative objective) [12]. Since the game is finite, payoffs are bounded, and this implies that \( \text{Val}(v) \) is finite, and thus that \( v \notin V_{+\infty} \).

Assume now that \( v \notin \text{Attr}(T) \), in this case, we know that Player 2 has a (memoryless) winning strategy from \( v \) to avoid \( T \) (qualitative objective) [12]. This implies that \( \text{Val}(v) \) is infinite, and thus that \( v \in V_{+\infty} \). □

The previous result applies for reachability payoffs \( \mathbf{RTP}, \mathbf{RMP}, \) and \( \mathbf{RDP}_\lambda \) for \( \lambda \in (0,1) \). Hence, in the following, when considering a reachability quantitative game \( \mathcal{G} = (V, E, \omega, T_{\mathbf{RP}}) \), we always assume that the set of vertices is equal to the attractor of \( T \) and thus that no vertex has value \(+\infty\).

3 Prologue on mean-payoff and discounted-payoff games

We start by considering reachability mean-payoff and discounted-payoff games. We achieve efficient algorithms for those games by introducing logarithmic space
reductions to the purely quantitative versions, for which pseudo-polynomial time algorithms exist. This is to be contrasted with total payoff games, where, to the best of our knowledge, no pseudo polynomial time algorithm exists in the literature, even in the purely quantitative case. The definition of such an efficient algorithm for total payoff games (both in the reachability and in the purely quantitative cases) is the core contribution of the next sections of the paper.

3.1 Reachability mean-payoff games

We show in this section an equivalence result between reachability mean-payoff games, and mean-payoff games. More precisely we show that when we fix the initial vertex in the games (i.e., when we are only interested in computing the value of a given vertex), reachability mean-payoff games and mean-payoff games are equivalent.

Let $G = (V, E, \omega, T-RMP)$ be a reachability mean-payoff game. Without loss of generality, we assume that $T$ is a singleton $T = \{t\}$ and that the only edge going out of the target is a loop $(t, t)$. Recall that we assumed $V$ to be the attractor of $t$.

For all vertices $v^* \in V$, we construct the mean-payoff game $M(G, v^*) = (V', E', \omega', MP)$ as follows. The set of vertices $V'$ equals $V$, but the partition may differ, namely $v^* \in V_i$ if and only if $t \in V_i'$. Next, the edges are the same as in the initial game except that (i) we remove the loop $(t, t)$; and (ii) we replace each edge of the form $(v^*, v)$ by $(t, v)$. Formally:

$$E' = E \setminus \{(t, t)\} \cup \{(t, v) \mid (v^*, v) \in E\},$$

and for all $v_1, v_2$ with $v_1 \neq t$, $\omega'(v_1, v_2) = \omega(v_1, v_2)$ and $\omega'(t, v_2) = \omega(v^*, v_2)$. Let us show that the value of $v^*$ in $G$ is equal to the value of $t$ in $M(G, v^*)$ (Corollary 6). In order to do that we prove both inequalities separately.

Example 3. Take the game $G$ depicted in Fig. 1. In this game the value of vertex $v_1$ is 0, as Player 2 will always prefer go to $v_3$ (going to $v_2$ enables Player 1 to secure a cost of $-4/3$), and from there Player 1 will go to $v_4$ and loop for a long time before reaching the target, ensuring a value arbitrarily close to 0. Notice that from $v_1$ there is no strategy which has value 0 thus there are no optimal strategies. On the other hand the value of $v_2$ is $-4/2 = -2$ because Player 1 goes
to \(v_3\) and then reaches the target (in two steps). Notice that the choice of what to play on \(v_3\) depends on the vertex from which the game started, indeed when starting in \(v_1\), Player 1 better plays \(v_3\) from \(v_1\); and starting in \(v_2\), she better goes directly to the target \(t\). This property does not hold in (purely quantitative) mean-payoff games where optimal strategies do not depend on the initial vertex. This explains why we need to construct one equivalent game per initial vertex.

We have depicted \(\mathcal{M}(G, v_1)\) and \(\mathcal{M}(G, v_2)\) in Fig. 1.

Lemma 4. \(\text{Val}_c(v^*) \leq \text{Val}_{\mathcal{M}(G, v^*)}(t)\).

**Proof.** In this proof we write \(m = \text{Val}_{\mathcal{M}(G, v^*)}(t)\). First we introduce some notations. Let \(\pi\) be a play in \(G\) starting from \(v^*\). We define the play \(\mathcal{M}(\pi)\) in \(\mathcal{M}(G, v^*)\) as follows: if \(\pi = v^*v_1\cdots v_k\), then \(\mathcal{M}(\pi) = (tv_1\cdots v_k)^\omega\), (in this case \(\text{MP}(\mathcal{M}(\pi)) = \text{RMP}(\pi)\)), otherwise, we have \(\pi = v^*v_1v_2\cdots\), with \(v_i \neq t\), and we let \(\mathcal{M}(\pi) = tv_1v_2\cdots\) (in this case \(\text{MP}(\mathcal{M}(\pi)) = \text{MP}(\pi)\)).

Let \(\sigma_1\) be a memoryless strategy of Player 1 in \(\mathcal{M}(G, v^*)\), we define the (non necessarily memoryless) strategy \(\overrightarrow{\sigma_1}\) in \(G\) as follows: \(\overrightarrow{\sigma_1}(v^*) = \sigma_1(t)\), \(\overrightarrow{\sigma_1}(v^*v_1\cdots v_k) = \sigma_1(v_k)\) if \(v_k \neq t\), \(\overrightarrow{\sigma_1}(v^*v_1\cdots v_k t) = t\). Notice that for all play \(\pi\) starting from \(v^*\) in \(G\), \(\pi \in \text{Play}(v^*, \overrightarrow{\sigma_1})\) if and only if \(\mathcal{M}(\pi) \in \text{Play}(t, \sigma_1)\).

Consider now \(\sigma_1\) to be a memoryless optimal strategy for Player 1 in \(\mathcal{M}(G, v^*)\) (that exists by [13]). For all \(\varepsilon > 0\), we will define a strategy \(\sigma_1^\varepsilon\) in \(G\) such that all play starting from \(v^*\) that conforms to \(\sigma_1^\varepsilon\) has value less than or equal to \(m + \varepsilon\). Intuitively \(\sigma_1^\varepsilon\) consists in playing like \(\sigma_1\) for a long time getting close to \(m\), and then either the play has ended and we show that the value is at most \(m\) or Player 1 switches strategy to reach the target in less than \(|V|\) steps, ensuring to stay very close to \(m\) (i.e., at most \(m + \varepsilon\)).

Notice first that, since \(V\) is the attractor of \(t\), there exists a memoryless strategy \(\sigma_1^\text{Attr}\) such that for all vertex \(v\), every play in \(\text{Play}(v, \sigma_1^\text{Attr})\) reaches \(t\) after at most \(|V|\) steps. We let \(\ell = \left\lceil \frac{2|V|m}{\varepsilon} \right\rceil\) intuitively \(\ell\) is the minimum length of a prefix such that switching to a reachability strategy ensures that the average value will not increase more than \(\frac{\varepsilon}{2}\). Finally we say that a prefix \(\pi\) is **switchable** if \(|\pi| \geq \ell\) and \(\text{MP}(\pi) \leq m + \frac{\varepsilon}{2}\).

We can now define the strategy \(\sigma_1^\varepsilon\). If \(\pi = v_1\cdots v_k\) contains a switchable prefix, and \(v_k \neq t\), define \(\sigma_1^\varepsilon(\pi) = \sigma_1^\text{Attr}(\pi)\), otherwise define \(\sigma_1^\varepsilon(\pi) = \overrightarrow{\sigma_1}(\pi)\).

First we show that all plays \(\pi\) in \(\text{Play}(v^*, \sigma_1^\varepsilon)\) reaches \(t\). Assume by contradiction that \(\pi\) does not contain a switchable prefix and does not reach \(t\). Therefore for all prefixes \(v^*v_1\cdots v_k\) of \(\pi\) such that \(k \geq \ell\), \(\text{MP}(v^*v_1\cdots v_k) > m + \frac{\varepsilon}{2}\), thus \(\text{MP}(\pi) \geq m + \frac{\varepsilon}{2}\) and \(\text{RMP}(\pi) \geq m + \frac{\varepsilon}{2}\). Since no prefix is switchable, \(\pi\) conforms to \(\overrightarrow{\sigma_1}\), thus \(\mathcal{M}(\pi)\) conforms to \(\sigma_1\) in \(\mathcal{M}(G, v^*)\), and \(\text{MP}(\mathcal{M}(\pi)) \geq m + \frac{\varepsilon}{2}\), which raises a contradiction since \(\mathcal{M}(\pi) \in \text{Play}(v^*, \sigma_1)\) with \(\sigma_1\) an optimal strategy. Then there exists \(k\) such that \(\pi = v^*v_1\cdots v_k\), \(v_k \neq t\), \(\pi_2 = v^*v_1\cdots v_k\) a switchable prefix. Thus \(v_k\pi_2 \in \text{Play}(v^*, \sigma_1^\text{Attr})\) hence \(t\) is reached.

Now let \(\pi = v^*v_1\cdots v_k t^\omega \in \text{Play}(v^*, \sigma_1^\varepsilon)\) with \(v_i \neq t\) for all \(i\). Assume that \(v^*v_1\cdots v_k\) does not contain a prefix that is switchable. Therefore \(\pi\) conforms to \(\overrightarrow{\sigma_1}\), and \(\mathcal{M}(\pi)\) conforms to \(\sigma_1\), so that \(\text{RMP}(\pi) = \text{MP}(\mathcal{M}(\pi)) \leq m\).
Assume that \( v^*v_1 \cdots v_k \) contain a prefix \( v^*v_1 \cdots v_j \) that is switchable (with \( j \geq \ell \)). Thus for all \( j < i < k \), we have \( v_{i+1} = \sigma^1_{\text{Attr}}(v_i) \) and \( t = \sigma^1_{\text{Attr}}(v_k) \), then \( \pi_2 = v_jv_{j+1} \cdots \) conforms to \( \sigma^1_{\text{Attr}} \), which ensures that \( k < j + |V| \). Then the following holds:

\[
\text{RMP}(\pi) = \frac{j\text{MP}(v^*v_1 \cdots v_j) + \text{TP}(v_j \cdots v_k \tau)}{k + 1}.
\]

By the definition of switchable prefixes, \( \text{MP}(v^*v_1 \cdots v_j) \leq m + \varepsilon/2 \). Moreover, \( \text{TP}(v_j \cdots v_k \tau) \leq W|V| \) since \( k - j + 1 \leq |V| \). Using that \( \ell \leq j \leq k + 1 \), we obtain

\[
\text{RMP}(\pi) \leq m + \varepsilon/2 + \frac{W|V|}{\ell} \leq m + \varepsilon.
\]

Hence, every play of \( \text{Play}(v^*, \sigma^1_1) \) has a reachability mean-payoff bounded by \( m + \varepsilon \), so that \( \text{Val}_\mathcal{G}(v^*) \leq m + \varepsilon \). Since this holds for every \( \varepsilon > 0 \), we obtain \( \text{Val}_\mathcal{G}(v^*) \leq m = \text{Val}_{\mathcal{M}(\mathcal{G}, v^*)}(t) \). 

\[\square\]

**Lemma 5.** \( \text{Val}_\mathcal{G}(v^*) \geq \text{Val}_{\mathcal{M}(\mathcal{G}, v^*)}(t) \).

**Proof.** Let \( \sigma_1 \) be a strategy of Player 1 in \( \mathcal{G} \) such that \( \text{Val}_\mathcal{G}(v^*, \sigma_1) \neq +\infty \). We construct a strategy \( \sigma'_1 \) in \( \mathcal{M}(\mathcal{G}, v^*) \) such that for all plays starting from \( t \) that conforms to \( \sigma'_1 \), we have \( \text{Val}_{\mathcal{M}(\mathcal{G}, v^*)}(t, \sigma'_1) \leq \text{Val}_\mathcal{G}(v^*, \sigma_1) \). Intuitively, \( \sigma'_1 \) consists in playing like \( \sigma_1 \) and then the play goes back to \( t \) the strategy is reinitialized and played again. Therefore, a play from \( t \) in \( \mathcal{M}(\mathcal{G}, v^*) \) that conforms to \( \sigma'_1 \) corresponds to a succession of finite plays in \( \mathcal{G} \) that conforms to \( \sigma_1 \), that will ensure a mean-payoff bounded by \( \text{Val}_\mathcal{G}(v^*, \sigma_1) \).

We define the strategy \( \sigma'_1 \) as follows. Let \( \pi = tv_1 \cdots v_k \) be a history in \( \mathcal{M}(\mathcal{G}, v^*) \) starting in \( t \), and let \( j \) be the last position in \( \pi \) such that \( v_j = t \); we define \( \sigma'_1(\pi) = \sigma_1(v^*v_{j+1} \cdots v_k) \).

Now we consider \( \pi \in \text{Play}_{\mathcal{M}(\mathcal{G}, v^*)}(t, \sigma'_1) \). Let us show that the target \( t \) occurs infinitely often in \( \pi \). Assume, by means of contradiction, that it is not the case, i.e., that \( \pi = \pi'^*tv_1v_2 \cdots \) with \( v_i \neq t \) for all \( i \), where \( \pi'^* \) is a finite play. Then, we have for all \( i \) that \( v_i = \sigma_1(v^*v_1 \cdots v_{i-1}) \). Thus \( v^*v_1v_2 \cdots \) conforms to \( \sigma_1 \) and does not reach \( t \), which is a contradiction since \( \text{Val}_\mathcal{G}(v^*, \sigma_1) \neq +\infty \).

Therefore, \( \pi \) is of the form \( \pi = tv_1^1 \cdots v_{i_1}^1 t^1v_1^2 \cdots v_{i_2}^2 \tau \cdots \) such that for all \( i \) and for all \( j \leq \ell_i \), \( v_j^i \neq t \). Moreover, for all \( i \), for all \( j \leq \ell_i \): we have \( v_j^i = \sigma_1(v^*v_1 \cdots v_{i-1}) \). Also, for all \( i \), \( t = \sigma_1(v^*v_1 \cdots v_{i}^i) \). Thus \( v^*v_1 \cdots v_{i}^i \tau^i \) conforms to \( \sigma_1 \), and hence:

\[
\text{MP}_{\mathcal{M}(\mathcal{G}, v^*)}(tv_1^1 \cdots v_{\ell_i}^1, t) \overset{\text{(1)}}{=} \text{MP}_{\mathcal{G}}(v^*v_1^1 \cdots v_{\ell_i}^i, t)
\overset{\text{(2)}}{=} \text{RMP}(v^*v_1^1 \cdots v_{\ell_i}^i, t^i)
\leq \text{Val}_\mathcal{G}(v^*, \sigma_1) \quad \text{(\( \bullet \))}
\]

where (1) comes by construction of \( \mathcal{M}(\mathcal{G}, v^*) \) and (2) from the definition of \( \text{RMP} \).
Recall that for all \( j \), \( \pi[j] \) denotes the prefix of \( \pi \) of size \( j \). Furthermore remark that (*) for all history \( v_0 \cdots v_k \), \( \text{TP}(v_0 \cdots v_k) = k\text{MP}(v_0 \cdots v_k) \). By the definition of mean-payoff:

\[
\text{MP}(\pi) = \liminf_{j \to +\infty} \text{MP}(\pi[j])
\]

\[
\leq \liminf_{n \to +\infty} \text{MP}(tv_1^1 \cdots v_t^1 \cdots v_n^0 \cdots v_n^0 t)
\]

\[
\leq \liminf_{n \to +\infty} \frac{\text{TP}(tv_1^1 \cdots v_t^1 \cdots v_n^0 \cdots v_n^0 t)}{\sum_{i=1}^{n}(\ell_i + 1)}
\]

\[
\leq \liminf_{n \to +\infty} \frac{\sum_{i=1}^{n}(\ell_i + 1)\text{MP}(tv_1^1 \cdots v_t^1 t)}{\sum_{i=1}^{n}(\ell_i + 1)}
\]

\[
\leq \liminf_{n \to +\infty} \frac{\sum_{i=1}^{n}(\ell_i + 1)\text{Val}(v^*, \sigma_1)}{\sum_{i=1}^{n}(\ell_i + 1)}
\]

\[
\leq \liminf_{n \to +\infty} \text{Val}(v^*, \sigma_1)
\]

Since this holds for every play \( \pi \in \text{Play}_M(G, \omega)(t, \sigma_1') \), we know that

\[
\text{Val}_M(G, \omega)(t, \sigma_1') \leq \text{Val}(v^*, \sigma_1)
\]

which allows us to conclude. \( \square \)

As a corollary of the two previous lemmas, we obtain:

**Corollary 6.** \( \text{Val}(v^*) = \text{Val}_M(G, \omega)(t) \).

Using this reduction, we may apply algorithms solving mean-payoff games in order to solve reachability mean-payoff games. In particular, thanks to the value iteration algorithm of [13] that runs in pseudo-polynomial time, we obtain a pseudo-polynomial time algorithm that computes the optimal values and optimal or \( \epsilon \)-optimal strategies in reachability mean-payoff games.

For the sake of completeness, let us finally show that mean-payoff games and reachability mean-payoff are indeed as hard as each other. We establish this result by introducing a reduction from mean-payoff games to reachability mean-payoff games. Let \( G = (V, E, \omega, \text{MP}) \) be a mean-payoff game and let \( t \) be a vertex in \( A \). We define the reachability mean-payoff game \( H(G, v^*) \) by \( H(G, v^*) = (V \cup \{v^*\}, E', \omega', \{t\} \cdot \text{RMP}) \) with \( E' = (E \cap (V \setminus \{t\})) \times V \cup \{(v^*, v) \mid (t, v) \in E \} \cup \{(t, t)\} \) and \( \omega'(v, v') = \omega(v, v') \) if \( (v, v') \in E \), \( \omega'(v^*, v) = \omega(t, v), \omega(t, t) = 0 \).

Noticing that \( G = M(H(G, t), v^*) \), we obtain, using the previous corollary, that

**Theorem 7.** \( \text{Val}_{H(G, v^*)}(v^*) = \text{Val}(t) \).
Proof. $\text{Val}_G(t) = \mathcal{M}(H(G, t), v^*) = \text{Val}_{H(G, v^*)}(v^*)$ by the remark above and Corollary 6.

Hence, we can solve mean-payoff games by reducing them to reachability mean-payoff games.

3.2 Reachability discounted-payoff games

A reduction in both directions can be shown for discounted-payoff as done before for mean-payoff. We shortly sketch the reductions. Let $G = (V, E, \omega, T)$ be a reachability $\lambda$-discounted game with $T$ as target. As before, without loss of generality we can assume that $T$ is a singleton $\{t\}$, that the only outgoing edge from $t$ is $(t, t)$ and that $\omega(t, t) = 0$. Consider the $\lambda$-discounted game $G' = (V, E, \omega, \text{DP}_\lambda)$.

**Theorem 8.** For all vertices $v$, $\text{Val}_G(v) = \text{Val}_{G'}(v)$.

**Proof (Sketch).** To show that $\text{Val}_G(v) \leq \text{Val}_{G'}(v)$ simply notice that given an optimal strategy $\sigma_1$ for Player 1 in $G'$ (optimal memoryless strategies exist, e.g., by [7]) and an integer $n$, one can define the strategy $\sigma^n_1$ in $G$ that consists in playing $n$ times $\sigma_1$ and then playing a memoryless reachability strategy (that exists since we play in the attractor of the target). Since after $n$ steps, the weights are discounted by a factor $\lambda^n$. One can easily show that $\text{Val}_G(v, \sigma^n_1) - \lambda^n W \leq \text{Val}_{G'}(v, \sigma^n_1) \leq \text{Val}_G(v, \sigma_1) + \lambda^n W$.

Thus the sequence of values $(\text{Val}_G(v, \sigma^n_1))_{n \geq 1}$ converges towards $\text{Val}_{G'}(v, \sigma_1) = \text{Val}_G(v)$, which shows that $\text{Val}_G(v)$ is an upper bound of $\text{Val}_{G'}(v)$.

On the other hand, to show that $\text{Val}_G(v) \leq \text{Val}_{G'}(v)$ it suffices to remark that any strategy $\sigma_1$ in $G$ that ensures a value less than some value $m$ can be applied in $G'$ ensuring the same value. □

This reduction permits to apply the classical value iteration algorithm existing for discounted-payoff games (that runs in pseudo-polynomial time, see [13] or Appendix 3 for a detailed explanation) to solve reachability discounted-payoff games in pseudo-polynomial time, more precisely, polynomially in the size of the graph and exponential in a representation of the discount factor $\lambda$.

Once again, let us show that reachability discounted-payoff games are indeed as hard as discounted-payoff games. For that, consider a $\lambda$-discounted game $G = (V, E, \omega, \text{DP}_\lambda)$. We informally define a reachability discounted-payoff game $G'$ as follows. The game $G'$ simulates the game $G$ in which after every move, Player 1 can choose to leave the game and to go to the target while paying a cost higher than anything he would pay by staying infinitely in the game, e.g., $W + 1$.

**Theorem 9.** For all vertices $v$, $\text{Val}_G(v) = \text{Val}_{G'}(v)$.
Proof (sketch). Inequality $\text{Val}_G(v) \leq \text{Val}_{G'}(v)$ is shown as in the previous theorem.

To show that $\text{Val}_G(v) \geq \text{Val}_{G'}(v)$, take a strategy $\sigma_1$ in $G'$, and define the strategy $\sigma_1'$ in $G$ as playing like $\sigma_1$ until it tries to reach the target, and then pick any of the successors indifferently. Because of the choice of the weight $W + 1$ of the path following the choice to exit the game, one can show that $\text{Val}_G(v, \sigma_1') \leq \text{Val}_{G'}(v, \sigma_1)$.

$$\Box$$

4 Efficient algorithms for reachability total-payoff games

In this section, we focus on reachability total-payoff games. A first attempt could be to consider reducing those games to total-payoff games (without reachability): this is the path we have followed previously for mean-payoff and discounted-payoff. However, this seems useless for the total-payoff case, since no efficient algorithm is known to solve total-payoff games. Instead, we produce an ad-hoc efficient algorithm, and will apply it in the next section to solve total-payoff games through a reduction.

Reachability total-payoff games were studied in [5, Thm 8], where NP \cap co-NP complexities were obtained, and in [9] for the special case of non-negative weights, where a polynomial time algorithm inspired by Dijkstra’s algorithm was given. We now focus on practical algorithms for solving these games, and, in particular, computing the values and optimal strategies, if any. We summarize our results in the following theorem, that is proved in the remaining of the section.

**Theorem 10.** Let $G = (V, E, \omega, T\text{-RTP})$ be a reachability total-payoff game.

1. Determining if the value $\text{Val}(v)$ of the game in a vertex $v$ is $-\infty$ is in NP \cap co-NP, can be decided in pseudo-polynomial time\footnote{Here, pseudo-polynomial means polynomial in the size of the graph where weights are encoded in unary.} and is as hard as solving mean-payoff games.

2. In case the value $\text{Val}(v)$ is finite for every vertex $v$, optimal strategies exist for both players. Moreover, Player 2 has optimal memoryless strategies, whereas Player 1 has optimal finite-memory strategies. Finally, values $\text{Val}(v)$, as well as optimal strategies for both players, can be computed in pseudo-polynomial time.

It has to be noticed as a corollary of the first item that the problem of determining if the value of a vertex is at most $K$, for $K$ given as an input, is also NP \cap co-NP. However, if we restrict our attention to the class of games where every vertex has a finite value, we do not know whether the complexity drops down, or not, to PTIME.

It has to be underpinned that, even if every vertex $v$ has a finite value $\text{Val}_A(v) \in \mathbb{R}$, memoryless optimal strategies may not exist for Player 1, as shown in the next example.
Example 11. Consider the reachability total-payoff game played on the weighted graph of Fig. 2, with \( W \) a positive integer. Here Player 1 vertices are shown as circles and Player 2 as boxes, and weights shown on each edge. Moreover, we consider \( v_3 \) to be the target. This example shows that even when every vertex \( v \) has a finite value \( \text{Val}(v) \in \mathbb{R} \), memoryless optimal strategies may not exist for Player 1. Indeed, in vertex \( v_1 \) the value achievable by Player 1 is \(-W\), but there is no memoryless strategy of Player 1 to achieve it.

4.1 Finding vertices with value \(-\infty\)

Let us turn our attention to a procedure allowing to determine whether a vertex has value \(-\infty\), i.e., compute the set \( V_{-\infty} = \{ v \in V \mid \text{Val}(v) = -\infty \} \). This is more complex a priori, since the objective of Player 1 is to reach the final vertices, so that \( \text{Val}(v) = -\infty \) if for all constant \( M \in \mathbb{R} \), Player 1 has a strategy \( \sigma_1 \) such that \( \text{Val}(v, \sigma_1) < M \). Thus, it is a priori not possible to reduce this question to a qualitative reachability question anymore. Instead, we now link this problem to mean-payoff games. We use mean-payoff games, and in particular, the result that they are positionally determined, i.e., there exists a profile of memoryless strategies \( (\sigma_{1}^*, \sigma_{2}^*) \) so that the mean-payoff obtained from a vertex \( v \) following this profile is exactly the value of \( v \) in the game [4]. Finally, those values can be computed in pseudo-polynomial time.

Let us now consider the mean-payoff game \( G' = \langle V, E, \omega, \text{MP} \rangle \) over the same graph as \( G \). Then:

**Proposition 12.** For all reachability total-payoff game \( G = \langle V, E, \omega, \text{T-RTP} \rangle \): \( \text{Val}_{G'}(v) = -\infty \) if and only if \( \text{Val}_{G'}(v) < 0 \), where \( G' \) is the mean-payoff game obtained by the above construction.

**Proof.** If \( \text{Val}_{G'}(v) < 0 \), the outcome starting in \( v \) and following the profile \( (\sigma_{1}^*, \sigma_{2}^*) \) of memoryless strategies necessarily starts with a finite prefix, with a total cost \( w \), and then loops in a cycle with a total cost \( w' \) less than 0. For every \( M > 0 \), we construct a strategy \( \sigma_1 \) that ensures in \( G \) a cost less than or equal to \(-M\): this will prove that \( \text{Val}_{G}(v) = -\infty \). Since in every vertex of \( G' \), Player 1 has a strategy in \( G \) to reach the final vertices, there exists \( w'' \) such that Player 1 can reach from any vertex of \( G' \) the target with a cost at most \( w'' \). The strategy \( \sigma_1 \) of Player 1 is then to follow \( \sigma_1^* \) until the accumulated cost is less than \(-M - w''\), at which point it follows its strategy to reach the target. Notice that strategy is a...
finite-memory strategy that requires a memory of size \( \log_2(M + |w'| + |w| + |w'|) \), enabling to store the binary encoding of the current value of the cost.

Reciprocally, if \( \text{Val}_2(v) = -\infty \), consider \( M = |V| \times \max |\omega| \) and a strategy \( \sigma^M_1 \) of Player 1 ensuring a cost less than \(-M\), i.e., such that \( \text{Val}(v, \sigma^M_1) < -M \). Consider the finitely-branching tree built from \( \mathcal{G} \) by unfolding the game from vertex \( v \) and resolving the choices of Player 1 with strategy \( \sigma^M_1 \). Each branch of this tree corresponds to a possible strategy of Player 2. Since this strategy generates a finite cost, we are certain that every such branch leads to a vertex of \( T \). If we trim the tree at those vertices, we finally obtain a finite tree. Now, for a contradiction, suppose the optimal memoryless strategy \( \sigma^*_2 \) of Player 2 ensures a nonnegative mean-payoff, that is, \( \text{Val}_2(v, \sigma^*_2) \geq 0 \). Consider the branch of the previous tree where Player 2 follows strategy \( \sigma^*_2 \). Since this finite branch has cost less than \(-M = -|V| \times \max |\omega| \), we know for sure that there is two occurrences of the same vertex \( v' \) with an in-between cost less than 0; otherwise, by removing all nonnegative cycles, we obtain a play without repetition of vertices, henceforth of length bounded by \(|V|\), and therefore of cost at least \(-M \). Suppose that \( v' \in V_2 \). Then, Player 1 has a strategy \( \sigma_1 \) to ensure a negative mean-payoff \( \text{Val}_2(v, \sigma_1) < 0 \); indeed, it simply modifies its strategy so that it always follows his choices made in the negative cycle starting in \( v' \), ensuring that, against the optimal strategy \( \sigma^*_2 \) of Player 2, he gets a mean-payoff being the cost of the cycle. This is a contradiction since Player 2 is supposed to have a strategy ensuring a nonnegative mean-payoff from \( v \). Hence, \( v' \in V_1 \). But the same contradiction appears in that case since Player 1 can force that it always stays in the negative cycle by modifying his strategy. Finally, we have proved that Player 2 cannot have a memoryless strategy ensuring a nonnegative mean-payoff from \( v \). By memoryless determinacy of the mean-payoff games, this ensures that Player 1 has a memoryless strategy ensuring a negative mean-payoff from \( v \). □

Conversely, we can use mean-payoff games to obtain a lower bound on the complexity of determining whether a vertex has value \(-\infty \). We reduce mean-payoff games to reachability total-payoff games as follows. Let \( \mathcal{G} = (V, E, \omega, \text{MP}) \) be a mean-payoff game. Without loss of generality, we may suppose that the graph of the game is bipartite, in the sense that \( E \subseteq V_1 \times V_2 \cup V_2 \times V_1 \). The problem we are interested in is to decide whether \( \text{Val}_2(v) < 0 \) for a given vertex \( v \). We now construct a reachability total-payoff game \( \mathcal{G}' = (V', E', \omega', T^\ast\text{RTP}) \) from \( \mathcal{G} \). The only difference is the presence of a fresh target vertex \( v_f \) on top of vertices of \( V \): \( V' = V \cup \{v_f\} \) with \( T' = \{v_f\} \). Edges of \( A' \) are given by \( E' = E \cup \{(v, v_f) \mid v \in V_1\} \cup \{(v_f, v_f)\} \). Weights of edges are given by: \( \omega'(v, v') = \omega(v, v') \) if \( (v, v') \in E \), and \( \omega'(v_f, v_f) = \omega'(v_f, v_f) = 0 \).

**Proposition 13.** For all mean-payoff game \( \mathcal{G} = (V, E, \omega, \text{MP}) \), and for all \( v \in V \), \( \text{Val}_2(v) < 0 \) if and only if \( \text{Val}_2(v) = -\infty \), where \( \mathcal{G}' \) is the reachability total-payoff game obtained from \( \mathcal{G} \) by the above construction.

**Proof.** In \( \mathcal{G}' \), all values are different from \(+\infty\), since Player 1 plays at least every two steps, and has the capability to go to the target vertex with weight 0. Hence,
by Proposition \[\text{12}\] we have that for every vertex \(v \in V', \Val_{G'}(v) = -\infty\) if and only if \(\Val_{G'}(v) < 0\).

To conclude, we prove that for all vertex \(v \in V\), \(\Val_G(v) < 0\) if and only if \(\Val_G(v) < 0\). If \(\Val_G(v) < 0\), by mapping the memoryless optimal strategies of \(\mathcal{G}\) into \(\mathcal{G}'\), we directly obtain that \(\Val_{G'}(v) \leq \Val_G(v) < 0\). Reciprocally, if \(\Val_{G'}(v) < 0\), we can project a profile of memoryless optimal strategies over vertices of \(\mathcal{G}\); the play obtained from \(v\) in \(\mathcal{G}\) is then the projection of the play obtained from \(v\) in \(\mathcal{G}'\), with the same cost. Hence, \(\Val_{G'}(v) \leq \Val_G(v) < 0\). \(\square\)

To conclude, from Proposition \[\text{12}\] and Proposition \[\text{13}\] and using pseudo-polynomial time algorithms solving mean-payoff games \[\text{14}\], we obtain:

**Corollary 14.** In reachability total-payoff game \(\mathcal{G}\), we can compute the set \(V_{-\infty}\) in pseudo-polynomial time. Moreover, mean-payoff games are polynomial time reducible to the problem of deciding whether a vertex belongs to \(V_{-\infty}\).

This ends the proof of item 1 in Theorem \[\text{10}\].

### 4.2 Computing finite values

Let us now turn our attention to the computation of the values that are different from \(+\infty\), \(-\infty\). First, we observe that removing the vertices in \(V_{-\infty} \cup V_{+\infty}\) from a reachability total-payoff game does not affect the values of the other vertices. More precisely, let \(\mathcal{G} = (V, E, \omega, T, RTP)\) be a reachability total-payoff game, and let \(\mathcal{G}' = (V', E', \omega', T', RTP)\), where \(V' = V \setminus (V_{-\infty} \cup V_{+\infty})\), \(E' = E \cap (V' \times V')\), \(\omega'\) is the projection of \(\omega\) on \(V'\) and \(T' = T \cap V'\). Then, for all \(v \in V'\), \(\Val_{G'}(v) = \Val_{G}(v) \in \{-\infty, +\infty\}\).

Since we can compute the sets \(V_{+\infty}\) and \(V_{-\infty}\) in all reachability total-payoff game, we can always effectively remove those vertices from the graph before computing the finite values of the remaining vertices. From now on, we will assume a reachability total-payoff game \(\mathcal{G} = (V, E, \omega, T, RTP)\) such that \(\Val_{G}(v) \notin \{-\infty, +\infty\}\) for all \(v \in V\). Moreover, since the cost of an outcome is computed only up to the first occurrence of a vertex \(v \in T\), we assume that the only outgoing edge of each vertex in \(T\) is a self-loop of weight \(0\), i.e., for all \((v, v') \in E\): \(v \in T\) implies \(v' = v\) and \(\omega(v, v') = 0\).

Let us now introduce some notations. For a play \(\pi = v_0v_1 \cdots v_i \cdots\), we let \(\RTP^{\leq i}(\pi) = \RTP(\pi)\) in case \(v_k \in T\) for some \(k \leq i\), and \(\RTP^{\leq i}(\pi) = +\infty\) otherwise. Let \(\sigma_1\) and \(\sigma_2\) be strategy for each player and let \(v \in V\) be a vertex. Then, for all \(i \geq 0\), we let \(\Val^{\leq i}\) be the function such that for all \(v \in V\):

\[
\Val^{\leq i}(v) = \inf_{\sigma_1} \sup_{\sigma_2} \RTP^{\leq i}(\text{Play}(v, \sigma_1, \sigma_2)).
\]

Observe that for all \(v \in V\), for all \(i \geq 0\), and for all strategies \(\sigma_1\) and \(\sigma_2\):

\[
\RTP^{\leq i}(\text{Play}(v, \sigma_1, \sigma_2)) \geq \RTP(\text{Play}(v, \sigma_1, \sigma_2)).
\]
Indeed, if the objective $T$ is reached within $i$ steps, then both plays will have the same payoff. Otherwise, $\text{RTP}^\leq_i(\text{Play}(v, \sigma_1, \sigma_2)) = +\infty$. Thus, for all $i \geq 1$ and $v \in V$:

$$\text{Val}^\leq_i(v) \geq \text{Val}(v) = \text{Val}(v)$$

which can be rewritten as

$$\text{Val}^\leq_i \geq \text{Val} = \text{Val}.$$

Let us now consider the sequence $(\text{Val}^\leq_i)_{i \geq 0}$. We first give an alternative definition of this sequence permitting to show its convergence.

**Lemma 15.** For all $i \geq 1$, for all $v \in V$:

$$\text{Val}^\leq_i(v) = \begin{cases} 
\min_{v' \in E(v)} (\omega(v, v') + \text{Val}^{\leq i-1}(v')) & \text{if } v \in V_1 \ \setminus \ T \\
\max_{v' \in E(v)} (\omega(v, v') + \text{Val}^{\leq i-1}(v')) & \text{if } v \in V_2 \ \setminus \ T \\
0 & \text{if } v \in T
\end{cases}$$

**Proof.** The lemma can be established by showing that $\text{Val}^\leq_i(v)$ can be computed by solving a finite game played on a finite tree of depth $i$ (i.e., by applying a backward induction). We adopt the following notation for labeled unordered trees. A leaf is denoted by $(v)$, where $v \in V$ is the label of the leaf. A tree with root labeled by $v$ and subtrees $A_1, \ldots, A_n$ is denoted by $(v, \{A_1, \ldots, A_n\})$. Then, for each $v \in V$ and $i \geq 0$, we define $A^i(v)$ as follows:

$$A^0(v) = (v)$$

for all $i \geq 1$ : $A^i(v) = (v, \{A^{i-1}(v_j) \mid (v, v_j) \in E\})$

Now, let us further label those trees by a value in $\mathbb{Z} \cup \{+\infty\}$ thanks to the function $\lambda$. For all tree of the form $A^0(v) = (v)$, we let

$$\lambda(A^0(v)) = \begin{cases} 
0 & \text{if } v \in T \\
+\infty & \text{if } v \not\in T
\end{cases}$$

For all tree of the form $A^i(v) = (v, \{A^{i-1}(v_1), \ldots, A^{i-1}(v_m)\})$ (for some $i \geq 1$), we let

$$\lambda(A^i(v)) = \begin{cases} 
\min_{1 \leq j \leq m} (\omega(v, v_j) + \lambda(A^{i-1}(v_j))) & \text{if } v \in V_1 \ \setminus \ T \\
\max_{1 \leq j \leq m} (\omega(v, v_j) + \lambda(A^{i-1}(v_j))) & \text{if } v \in V_2 \ \setminus \ T \\
0 & \text{if } v \in T
\end{cases}$$

(1)

Clearly, for all $v \in V$, for all $i \geq 0$, the branches of $A^i(v)$ correspond to all the possible finite plays $\text{Play}(v, \sigma_1, \sigma_2)[i]$, i.e., there is a branch for each possible strategy profile $(\sigma_1, \sigma_2)$. Thus, $\lambda(A^i(v)) = \text{Val}^{\leq i}(v)$ for all $i \geq 0$, which permits us to conclude from $\blacksquare$. 

16
In the following, $\mathcal{F}$ denotes the function $\mathbb{Z}^V_\infty \rightarrow \mathbb{Z}^V_\infty$ that maps every vector $x \in \mathbb{Z}^V_\infty$ to

$$
\mathcal{F}(x)(v) = \begin{cases} 
\min_{v' \in E(v)} \left( \omega(v, v') + x(v') \right) & \text{if } v \in V_1 \setminus T \\
\max_{v' \in E(v)} \left( \omega(v, v') + x(v') \right) & \text{if } v \in V_2 \setminus T \\
0 & \text{if } v \in T 
\end{cases}
$$

We have just shown that for all $i \geq 1$, $\text{Val}^{<i} = \mathcal{F}(\text{Val}^{<i-1})$. Notice that $\mathcal{F}$ is a monotonous operator over a complete lattice $\mathbb{Z}^V_\infty$. By Knaster-Tarski’s theorem, we know that fixed points of $\mathcal{F}$ form a complete lattice; in particular, $\mathcal{F}$ admits a greatest fixed point. Moreover, by Kleene fixed point theorem, using the Scott-continuity of $\mathcal{F}$ (that holds by composition of continuous mappings), this greatest fixed point can be obtained as the limit of the non-increasing sequence of iterates $(\mathcal{F}(x_0))_{i \geq 0}$ starting in the maximal vector $x_0$ defined by $x_0(v) = +\infty$ for all $v \in V$. Observe that $\text{Val}^{<0}(v) = 0$ for all $v \in T$ and $\text{Val}^{<0}(v') = +\infty$ for all $v' \not\in T$, so that $\text{Val}^{<0} = \mathcal{F}(x_0)$. Therefore, we know that

**Proposition 16.** The sequence $(\text{Val}^{<i})_{i \geq 0}$ is a non-increasing sequence (i.e., $\text{Val}^{<i} \succeq \text{Val}^{<i+1}$, for all $i \geq 0$) that converges towards the greatest fixed point of $\mathcal{F}$.

**Remark 17.** Notice that, at this point, it would not be too difficult to show that $\text{Val}$ is a fixed point of operator $\mathcal{F}$. However, it is more difficult to show that it is the greatest fixed point of $\mathcal{F}$, and to deduce directly properties over optimal strategies in the reachability total-payoff game. Instead, we use the sequence $(\text{Val}^{<i})_{i \geq 0}$ to obtain more interesting results on $\text{Val}$. Notice also that replacing the starting vector $x_0$ by any vector larger than the greatest fixed point also leads to the convergence of the Kleene sequence towards the greatest fixed point.

We now study refined properties of the sequence $(\text{Val}^{<i})_{i \geq 0}$, namely its stationarity, the speed of convergence and the interpretation of its value in terms of the original reachability total-payoff game.

Let us now characterize how the $\text{Val}^{<i}$ will evolve over the $|V| + 1$ first steps. The next lemma states that, for each node $v$, the sequence $\text{Val}^{<0}(v)$, $\text{Val}^{<1}(v)$, $\ldots$, $\text{Val}^{<k}(v)$, $\ldots$, $\text{Val}^{<|V|}(v)$ is of the form

$$
+\infty, +\infty, \ldots, +\infty, a_{k+1}, a_{k+2}, \ldots, a_{|V|}
$$

where $k$ is the step at which $v$ has been added to the attractor, and each value $a_i$ is finite and bounded:

**Lemma 18.** Let $v \in V$ be a vertex and let $0 \leq k \leq |V|$ be such that $v \in \text{Attr}_k(T) \setminus \text{Attr}_{k-1}(T)$ (assuming $\text{Attr}_{-1}(T) = \emptyset$). Then, for all $0 \leq j \leq |V|$: (i) $j < k$ implies $\text{Val}^{<j}(v) = +\infty$ and (ii) $j \geq k$ implies $\text{Val}^{<j}(v) \leq j \times W$.
Proof. We prove the property for all vertices \( v \), by induction on \( j \).

**Base case:** \( j = 0 \). We consider two cases. Either \( v \in T \). In this case, \( k = 0 \), and we must show that \( \underline{\text{Val}}_{\leq 0}(v) \leq 0 \times W = 0 \), which is true by definition of \( \underline{\text{Val}}_{\leq 0} \).

Or \( v \not\in T \). In this case, \( k > 0 \), and we must show that \( \underline{\text{Val}}_{\leq 0}(v) = +\infty \), which is true again by definition of \( \underline{\text{Val}}_{\leq 0} \).

**Inductive case:** \( j = \ell \geq 1 \). Let us assume that the lemma holds for all \( v \), for all values of \( j \) up to \( \ell - 1 \), and let us show that it holds for all \( v \), and for \( j = \ell \).

Let us fix a vertex \( v \), and its associated value \( k \). We consider two cases.

1. First, assume \( k > \ell \). In this case, we must show that \( \underline{\text{Val}}_{\leq \ell}(v) = +\infty \).

   \[
   \underline{\text{Val}}_{\leq \ell}(v) = \min_{(v', v) \in E} \left( \omega(v, v') + \underline{\text{Val}}_{\leq \ell - 1}(v') \right) \quad \text{(Lemma 15)}
   \]

   \[
   = +\infty
   \]

2. Second, assume \( k \leq \ell \). In this case, we must show that \( \underline{\text{Val}}_{\leq \ell}(v) \leq \ell \times W \).

   As in the previous item, we consider two cases:

   (a) In the case where \( v \in V_1 \), we let \( \overline{v} \) be a vertex such that \( \overline{v} \in \text{Attr}_{k - 1}(T) \) and \( (v, \overline{v}) \in E \). Such a vertex exists by definition of the attractor. By induction hypothesis, \( \underline{\text{Val}}_{\leq \ell - 1}(\overline{v}) \leq \ell \times W \). Then:

   \[
   \underline{\text{Val}}_{\leq \ell}(v) = \min_{(v', v) \in E} \left( \omega(v, v') + \underline{\text{Val}}_{\leq \ell - 1}(v') \right) \quad \text{(Lemma 15)}
   \]

   \[
   \leq \omega(v, \overline{v}) + \underline{\text{Val}}_{\leq \ell - 1}(\overline{v})
   \]

   \[
   \leq \omega(v, \overline{v}) + (\ell - 1) \times W \quad \text{(Ind. Hyp.)}
   \]

   \[
   \leq W + (\ell - 1) \times W
   \]

   \[
   = \ell \times W
   \]

   (b) In the case where \( v \in V_2 \), we know that all successors \( v' \) of \( v \) belong to \( \text{Attr}_{k - 1}(T) \) by definition of the attractor. By induction hypothesis, for
all successors $v'$ of $v$: $\Val^{\leq \ell -1}(v') \leq \ell \times W$. Hence:

$$\Val^{\leq \ell}(v) = \max_{(v,v') \in E} \left( \omega(v,v') + \Val^{\leq \ell -1}(v') \right) \quad \text{(Lemma 15)}$$

$$\leq \max_{(v,v') \in E} (W + (\ell - 1) \times W) \quad \text{(Ind. Hyp.)}$$

$$= \ell \times W. \quad \Box$$

In particular, this allows us to conclude that, after $|V|$ steps, all values are bounded by $|V| \times W$:

**Corollary 19.** For all $v \in V$: $\Val^{\leq |V|}(v) \leq |V| \times W$.

The next step is to show that the sequence stabilises after a bounded number of steps:

**Lemma 20.** The sequence $\Val^{\leq 0}, \ldots, \Val^{\leq i}, \ldots$ stabilises after at most $(2|V| - 1) \times W \times |V| + |V|$ steps.

**Proof.** We first show that if Player 1 can secure, from some vertex $v$, a payoff less than $-((|V| - 1) \times W$, i.e., $\Val(v) < -((|V| - 1) \times W$, then it can secure an arbitrarily small payoff from that vertex, i.e., $\Val(v) = -\infty$, which contradicts our hypothesis that the value is finite. Hence, let us suppose that there exists a strategy $\sigma_1$ for Player 1 such that $\Val(v, \sigma_1) < -((|V| - 1) \times W$. Let $G'$ be the mean-payoff game introduced before Proposition 12. We will show that $\Val_{G'}(v) < 0$, which permits to conclude that $\Val_{G'}(v) = -\infty$ by Proposition 12. Let $\sigma_2$ be a memoryless strategy of Player 2. By hypothesis, we know that $\RTP(\Play(v, \sigma_1, \sigma_2)) < -((|V| - 1) \times W$. This ensures the existence of a cycle with negative cost in the play $\Play(v, \sigma_1, \sigma_2)$; otherwise, we could iteratively remove every possible nonnegative cycle of the history before reaching $T$ (hence reducing the cost of the play) and obtain a play without cycles before reaching $T$ with a cost less than $-((|V| - 1) \times W$, which is impossible (since it should be of length at most $|V| - 1$ to cross at most one occurrence of each vertex). Consider the first negative cycle in the play. After the first occurrence of the cycle, we let Player 1 chooses its actions like in the cycle. By this way, we can construct another strategy $\sigma'^*_1$ for Player 1, verifying that for every memoryless strategy $\sigma_2$ of Player 2, we have $\MP(\Play(v, \sigma'^*_1, \sigma_2))$ being the cost of the negative cycle in which the play finishes. Since for mean-payoff games, memoryless strategies are sufficient for Player 2, we deduce that $\Val_{G'}(v) < 0$.

This reasoning permits to prove that at every step $i$, $\Val^{\leq i}(v) \geq \Val(v) \geq -((|V| - 1) \times W + 1$ for all vertex $v$. Recall from Corollary 15 that, after $|V|$ steps in the sequence, all vertices are assigned a value smaller that $|V| \times W$. Moreover, the sequence is non-increasing by Property 16. In summary, for all $k \geq 0$ and for all vertices $v$:

$$-(|V| - 1) \times W + 1 \leq \Val^{\leq |V|+k}(v) \leq |V| \times W$$

Hence, in the worst case a strictly decreasing sequence will need $(2|V| - 1) \times W \times |V|$ steps to reach the lowest possible value where all vertices are assigned
−(|V|−1)×W+1 from the highest possible value where all vertices are assigned |V|×W. Thus, taking into account the |V| steps to reach a finite value on all vertices, the sequence stabilizes in at most $(2|V|−1)×W×|V|+|V|$ steps. □

Let us thus denote by $\overline{\text{Val}}^\leq$ the value obtained when the sequence $(\overline{\text{Val}}^\leq_i)_{i\geq0}$ stabilizes. We know from a previous discussion that $\overline{\text{Val}}^\leq$ is the greatest fixed point of operator $F$. We are now ready to prove that this value is the actual value of the game:

**Proposition 21.** For all reachability total-payoff game: $\overline{\text{Val}}^\leq = \text{Val}$.

**Proof.** We already know that $\overline{\text{Val}}^\leq \geq \text{Val}$. Let us show that $\overline{\text{Val}}^\leq \leq \text{Val}$. Let $v \in V$ be a vertex. Since $\text{Val}(v)$ is finite, there exists a strategy $\sigma_1$ for Player 1 that realises this value, i.e.,

$$\text{Val}(v) = \sup_{\sigma_2} \text{RTP(Play}(v, \sigma_1, \sigma_2)).$$

Notice that this holds because the values are integers, enducing that the infimum in the definition of $\overline{\text{Val}}^\leq(v) = \text{Val}(v)$ is indeed reached.

Let us build a tree $A_{\sigma_1}$ unfolding all possible plays from $v$ against $\sigma_1$. $A_{\sigma_1}$ has a root labeled by $v$. If a tree node is labeled by a Player 1 vertex $v$, this tree node has a unique child labeled by $\sigma_1(v)$. If a tree node is labeled by a Player 2 vertex $v$, this tree node has one child per successor $v'$ of $v$ in the graph, labeled by $v'$. We proceed this way until we encounter a node labeled by a vertex from $T$ in which case this node is a leaf. $A_{\sigma_1}$ is necessarily finite. Otherwise, by König’s Lemma, it has one infinite branch that never reaches $T$. From that infinite branch, one can extract a strategy $\sigma_2$ for Player 2 such that $\text{RTP(Play}(v, \sigma_1, \sigma_2)) = +\infty$, hence $\text{Val}(v) = +\infty$, which contradicts the hypothesis. Assume the tree has depth $m$. Then, $A_{\sigma_1}$ is a subtree of the tree $A$ obtained by unfolding all possible plays up to length $m$ (as in the proof of Lemma 15). In this case, it is easy to check that the value labeling the root of $A_{\sigma_1}$ after applying backward induction is larger than or equal to the value labeling the root of $A$ after applying backward induction. The latter is $\text{Val}(v)$ while the former is $\overline{\text{Val}}_{\sigma_1}^{\leq m}(v)$, by Lemma 15, so that $\text{Val}(v) \geq \overline{\text{Val}}_{\sigma_1}^{\leq m}(v)$. By Property 16 we finally obtain $\text{Val}(v) \geq \overline{\text{Val}}(v)$. □

As a corollary of this lemma and Property 16 we obtain:

**Corollary 22.** $\text{Val}$ is the greatest fixed point of $F$.

This permits to obtain a value iteration algorithm, described in Algorithm 1 that computes optimal values. Notice that we do not suppose that every vertex has a finite value, which is justified in the theorem below proving the correctness of the algorithm, as well as its complexity. A crucial argument is given in the following lemma:

**Lemma 23.** If the Kleene sequence $(F^i(x_0))_{i\geq0}$ is initiated with a vector of values $x_0$ that is greater or equal to the optimal value vector $\text{Val}$, then the sequence converges at least as fast as before towards the optimal value vector.
Algorithm 1: Value iteration for reachability total-payoff games

**Input**: Reachability total-payoff game \((V, E, \omega, T-\text{RTP})\), \(W\) greatest weight in absolute value in the graph

1. **foreach** \(v \in T\) do \(X(v) := 0\)
2. **foreach** \(v \in V \setminus T\) do \(X(v) := +\infty\)
3. repeat
4. \(X_{\text{pre}} := X\)
5. **foreach** \(v \in V_1 \setminus T\) do \(X(v) := \min_{v' \in E(v)} (\omega(v, v') + X_{\text{pre}}(v'))\)
6. **foreach** \(v \in V_2 \setminus T\) do \(X(v) := \max_{v' \in E(v)} (\omega(v, v') + X_{\text{pre}}(v'))\)
7. **foreach** \(v \in V \setminus T\) such that \(X(v) < -(|V| - 1) \times W\) do \(X(v) := -\infty\)
8. until \(X = X_{\text{pre}}\)
9. return \(X\)

**Proof.** This is a trivial consequence of Remark 17 and Corollary 22 saying that \(\text{Val}\) is the greatest fixed point of \(\mathcal{F}\).

**Theorem 24.** If a reachability total-payoff game \((V, E, \omega, T-\text{RTP})\) is given as input (possibly with values \(+\infty\) or \(-\infty\), the value iteration algorithm for reachability total-payoff games (Algorithm 1) outputs the vector \(\text{Val}\) of optimal values, after at most \((2|V| - 1) \times W \times |V| + 2|V|\) iterations.

**Proof.** Let us first suppose that values of every vertices are finite. Then, we can easily prove by induction that at the beginning of the \(j\)th step of the loop, \(X\) is equal to the vector \(\text{Val}^j\) and that the condition of line 14 has never been fulfilled. Hence, by Lemma 23 after at most \((2|V| - 1) \times W \times |V| + |V|\) iterations, all values are found correctly in that case.

Suppose now that there exist vertices with value \(+\infty\). Those vertices will remain at their initial value \(+\infty\) during the whole computation, and hence do not interfere with the rest of the computation.

Finally, consider that the game contains vertices with value \(-\infty\). We know that optimal values of vertices of values different from \(-\infty\) are at least \(-((|V| - 1) \times W + 1)\) so that, if the value of a vertex reaches an integer below \(-((|V| - 1) \times W\), we are sure that its value is indeed \(-\infty\), which proves correct the line 14 of the algorithm. This update may cost at most one step per vertices, which in total adds at most \(|V|\) iterations. Moreover, by Lemma 23 dropping the value to \(-\infty\) does not harm the correction for the other vertices (it may only speed the convergence of their values).

**Example 25.** Consider the reachability total-payoff game of Fig. 2. The successive values for vertices \((v_1, v_2)\) (as the target, value of \(v_3\) is always 0) computed by the value iteration algorithm are the following: \((+\infty, +\infty), (0, +\infty), (0, -1), (-1, -1), (-1, -2), (-2, -2), \ldots, (-W + 1, -W), (-W, -W)\). This requires \(2W\) steps to converge.
4.3 Computing optimal strategies

The computation of $\text{Val}^k$ permits to find the values of the game, as shown in Proposition 21. We now show how to deduce from this computation optimal strategies for both players in case of finite values. Infinite values have been considered in Section 2.3 and 4.1.

Strategies of Player 1

We have already seen an example in Fig. 2 of a game where Player 1 may need memory in an optimal strategy, i.e., where Player 1 has an optimal strategy, but no memoryless optimal strategies. Reciprocally, as a consequence of the previous work, we first show that, for vertices with finite value, Player 1 has always a finite-memory optimal strategy.

**Proposition 26.** In all reachability total-payoff game with only vertices of finite values, Player 1 has a finite-memory optimal strategy.

**Proof.** We explain how to reconstruct from the fixpoint computation an optimal strategy $\sigma^*_1$ for Player 1. Let $k$ be the integer such that $\text{Val}^k + 1 = \text{Val}^k$. For every step $\ell \leq k$ of the fixpoint computation, we define a strategy $\sigma^*_1$. For every finite play $\pi$ ending in a vertex $v$ of Player 1, if the length of $\pi$ is $i < \ell$, we let $\sigma^*_1(\pi) = \arg\min_{v' \in E(v)} (\omega(v, v') + \text{Val}^{\ell-1}(v'))$, and otherwise, we let $\sigma^*_1(\pi) = \arg\min_{v' \in E(v)} (\omega(v, v') + \text{Val}^0(v'))$.

Notice that the argmin operator may select any possible vertex as long as it selects one with minimum value. We also let $\sigma^*_1 = \sigma^*_k$ for every $\ell > k$, as well as $\sigma^*_1 = \sigma^*_k$. Notice that $\sigma^*_1$ is a finite-memory strategy, since it only requires to know the last vertex and the length of the prefix up to $k$.

We now prove that $\text{Val}(v, \sigma^*_1) = \sup_{\sigma^*_2} \text{RTP}(\text{Play}(v, \sigma^*_1, \sigma^*_2)) \leq \text{Val}^k(v)$ for all vertex $v$, which proves that $\sigma^*_1$ is an optimal strategy since $\text{Val}^k(v) = \text{Val}(v)$ by Proposition 21. To do so, we first show by induction on $\ell$ that

$$\text{RTP}^{\leq \ell}(\text{Play}(v, \sigma^*_1, \sigma^*_2)) \leq \text{Val}^{\leq \ell}(v) \quad (2)$$

holds for every strategy $\sigma^*_2$ of Player 2 and $v \in V$. This permits to conclude since, from $\sigma^*_1 = \sigma^*_k$ for every $\ell \geq k$, we can deduce:

$$\lim_{\ell \to \infty} \sup_{\sigma^*_2} \text{RTP}^{\leq \ell}(\text{Play}(v, \sigma^*_1, \sigma^*_2)) = \sup_{\sigma^*_2} \text{RTP}(\text{Play}(v, \sigma^*_1, \sigma^*_2))$$

and $\text{Val}^k$ is a stationary sequence converging towards $\text{Val}^k$. The proof by induction goes as follows. In case $\ell = 0$, either $v \in T$ and both terms of (2) are equal
to 0, or \(v \notin T\) and both terms of (2) are equal to \(+\infty\). Supposing now that the property holds for an index \(\ell\), let us prove it for \(\ell + 1\). For that, we consider a strategy \(\sigma_2\) of Player 2. In case \(v \in T\), we have

\[
\text{RTP}^{\leq \ell}(\text{Play}(v, \sigma_1^\ell, \sigma_2)) = 0 = \text{Val}^{\leq \ell}(v).
\]

We now consider the case \(v \notin T\). Let \(v'\) be the second vertex in \(\text{Play}(v, \sigma_1^{\ell+1}, \sigma_2)\). From the definition of \(\sigma_1^{\ell+1}\), \(\text{Play}(v, \sigma_1^{\ell+1}, \sigma_2)[\ell + 1]\) is the concatenation of \(v\) and \(\text{Play}(v', \sigma_1^\ell, \sigma_2)[\ell]\). Hence,

\[
\text{RTP}^{\leq \ell+1}(\text{Play}(v, \sigma_1^{\ell+1}, \sigma_2)) = \omega(v, v') + \text{RTP}^{\leq \ell}(\text{Play}(v', \sigma_1^\ell, \sigma_2)).
\]

By induction hypothesis, we obtain that

\[
\text{RTP}^{\leq \ell+1}(\text{Play}(v, \sigma_1^{\ell+1}, \sigma_2)) \leq \omega(v, v') + \text{Val}^{\leq \ell}(v').
\] (3)

Now, consider the two following cases.

- If \(v \in V_1 \setminus T\), we have \(v' = \sigma_1^{\ell+1}(v)\), so that, in case \(\ell + 1 \leq k\):
  \[
  \omega(v, v') + \text{Val}^{\leq \ell}(v') \leq \min_{v'' \in V_1 \setminus T} \left( \omega(v, v'') + \text{Val}^{\leq \ell}(v'') \right).
  \]

Using (3) and Lemma \[15\] we obtain

\[
\text{RTP}^{\leq \ell+1}(\text{Play}(v, \sigma_1^{\ell+1}, \sigma_2)) \leq \text{Val}^{\leq \ell+1}(v).
\]

In case \(\ell + 1 > k\), we indeed have \(\text{Val}^{\leq \ell} = \text{Val}^{\leq \ell+1} = \text{Val}^{\leq k}\), so that we conclude similarly.

- If \(v \in V_2 \setminus T\), we have \(v' = \sigma_2(v)\) and
  \[
  \omega(v, v') + \text{Val}^{\leq \ell}(v') \leq \max_{v'' \in V_2 \setminus T} \left( \omega(v, v'') + \text{Val}^{\leq \ell}(v'') \right).
  \]

Once again using (3) and Lemma \[15\] we obtain

\[
\text{RTP}^{\leq \ell+1}(\text{Play}(v, \sigma_1^{\ell+1}, \sigma_2)) \leq \text{Val}^{\leq \ell+1}(v).
\]

This concludes the induction proof.

Notice that the proof of Proposition \[26\] together with the statement of Lemma \[20\] imply that a memory of size pseudo-polynomial for the strategy of Player 1 is sufficient. Before stating the result for Player 2, we informally refine this result in order to find a strategy of Player 1 having more structural properties. It will be composed of two memoryless strategies \(\sigma_1^\star\) and \(\sigma_1^\text{Attr}\); the game will start with Player 1 following \(\sigma_1^\star\), and at some point, determined by the weight of the current history, Player 1 will switch to strategy \(\sigma_1^\text{Attr}\) which is an attractor strategy, i.e., a strategy that reaches the target in less than \(|V|\) steps, regardless of the weights along this path. Intuitively the strategy \(\sigma_1^\star\) ensures either to reach the
target with optimal value, or to go in cycles of negative weights. The only chance for Player 2 of having a greater value than the optimal is to go infinitely through these cycles without reaching the target. But if it does so, the total-payoff will decrease and at some point the value will be so low, that the cost of calling the attractor strategy will leave the total-payoff smaller than the optimal value. Let us formalise this construction.

For the sake of exposure, we present the construction when all values are finite, but such construction can be applied with few changes when some vertices have value $-\infty$ or $+\infty$. We start by defining a memoryless strategy $\sigma^*_i$ that has some good properties (stated in proposition 28). Let $X^i$ denote the value of variable $X$ after $i$ iteration of the loop of Algorithm 1 and let $X^0(v) = +\infty$ for all $v \in V$. We have seen that the sequence $X^0 \succ X^1 \succ X^2 \succ \cdots$ is stationary at some point, equal to $\text{Val}$. For all vertex $v \in V_1 \setminus \{t\}$, let $i_v > 0$ be the first index such that $X^v = \text{Val}(v)$. Fix a vertex $v' \neq t$ such that $X^v = \omega(v, v') + X^{v-1}(v')$ (that exists by definition) and define $\sigma^*_i(v) = v'$. The following lemma states that the vertex $v'$ already reached its final value at step $i - 1$.

**Lemma 27.** For all vertex $v \in V_1 \setminus \{t\}$, $X^{i-1}(\sigma^*_i(v)) = \text{Val}(\sigma^*_i(v))$.

**Proof.** Let $v' = \sigma^*_i(v)$. By contradiction assume that $X^{i-1}(v') > \text{Val}(v')$. Note that there exists $j > i$ such that $X^j(v') = \text{Val}(v')$. By definition,

$$\text{Val}(v) \leq X^j(v) \leq \omega(v, v') + X^{j-1}(v') = \omega(v, v') + \text{Val}(v')$$

$$< \omega(v, v') + X^{i-1}(v') = X^i(v) = \text{Val}(v),$$

which raises a contradiction. \qed

We can state the properties of $\sigma^*_i$: intuitively one can see it as an almost perfect strategy, in the sense that it is memoryless, if it reaches the target, then the value obtained is optimal, and if it does not reach the target then the total-payoff of the history will decrease as the game goes on. The only problem is one cannot ensure that we reach the target.

**Proposition 28.** For all vertex $v$, and for all play $\pi = v_1v_2 \cdots \in \text{Play}(v, \sigma^*_i)$,

1. if there exists $i < j$ such that $v_i = v_j$, then $\text{TP}(v_i \cdots v_j) < 0$,
2. if $\pi$ reaches $t$ then $\text{TP}(\pi) \leq \text{Val}(v)$.

**Proof.** Let us prove (1), take a cycle $v_i \cdots v_j$ with $v_j = v_i$. Notice that at least one vertex of this cycle belongs to Player 1, since, otherwise, Player 2 would have a strategy to obtain a value $+\infty$ for vertex $v_1$, which contradicts the hypothesis on the game. Hence, for the sake of the explanation, we suppose that $v_i \in V_1$. Let us also suppose that $i_{v_i}$ is maximal among $\{i_{v_i} \mid i \leq \ell < j, v_\ell \in V_1\}$. The following extends straightforwardly to the case where this maximal vertex of Player 1 is not $v_i$. We prove by induction over $i < \ell \leq j$ that

$$X^{i_{v_i}}(v_i) \geq \text{TP}(v_i \cdots v_\ell) + X^{i_{v_i}-1}(v_\ell).$$
The base case comes from the fact that since \( v_i \in V_1 \) we have \( \sigma_i^\star(v_i) = v_{i+1} \) thus \( X^{i_o} = \omega(v_i, v_{i+1}) + X^{i_o-1}(v_{i+1}) \). For the inductive case, let us consider \( i < \ell < j \) such that \( X^{i_o}(v_i) \geq TP(v_i \cdots v_\ell) + X^{i_o-1}(v_\ell) \) and let us prove it for \( \ell + 1 \).

If \( v_\ell \in V_2 \), by definition of \( X^{i_o} \), we have

\[
X^{i_o}(v_\ell) = \max_{(v_\ell, v') \in E} \omega(v_\ell, v') + X^{i_o-1}(v') \geq \omega(v_\ell, v_{\ell+1}) + X^{i_o-1}(v_{\ell+1}).
\]

In case \( v_j \in V_1 \), by maximality of \( i_o \), we have

\[
X^{i_o}(v_\ell) = X^{i_o}(v_\ell) = \omega(v_\ell, v_{\ell+1}) + X^{i_o-1}(v_{\ell+1})
\]

using that the sequence \( X^0, X^1, X^2, \ldots \) is non-increasing.

Hence, in all cases, we have

\[
X^{i_o}(v_\ell) \geq \omega(v_\ell, v_{\ell+1}) + X^{i_o-1}(v_{\ell+1})
\]

Using again that \( X^0, X^1, X^2, \ldots \) is non-decreasing, we obtain

\[
X^{i_o-1}(v_\ell) \geq X^{i_o}(v_\ell) \geq \omega(v_\ell, v_{\ell+1}) + X^{i_o-1}(v_{\ell+1}).
\]

Injecting this into the induction hypothesis, we have

\[
X^{i_o}(v_i) \geq TP(v_i \cdots v_\ell) + \omega(v_\ell, v_{\ell+1}) + X^{i_o-1}(v_{\ell+1})
\]

\[
= TP(v_i \cdots v_\ell) + X^{i_o-1}(v_{\ell+1})
\]

which concludes the proof by induction. In particular, for \( \ell = j \), as \( v_i = v_j \) we obtain that

\[
X^{i_o}(v_i) \geq X^{i_o-1}(v_i) + TP(v_i \cdots v_j).
\]

and as, by definition of \( i_o \), we have \( X^{i_o}(v_i) < X^{i_o-1}(v_i) \), we necessarily have \( TP(v_1 \cdots v_j) < 0 \).

To prove (2) we decompose \( \pi = v_1 \cdots v_k t^\omega \) with for all \( i, v_i \neq t \). We prove by decreasing induction on \( i \) that \( TP(v_i \cdots v_k t^\omega) \leq Val(v) \). If \( i = k + 1 \), \( TP(t^\omega) = 0 = Val(t) \). If \( i < k \), by induction we have \( TP(v_{i+1} \cdots v_k t^\omega) = Val(v_{i+1}) \) thus \( TP(v_i \cdots v_k t^\omega) = \omega(v_i, v_i(i + 1)) + Val(v_{i+1}) \). If \( v_i \in V_1 \) then \( v_{i+1} = \sigma_i^\star(v) \) and \( Val(v_i) = \omega(v_i, v_i(i + 1)) + Val(v_{i+1}) = TP(v_i \cdots v_k t^\omega) \). If \( v_i \in V_2 \), then \( TP(v_i \cdots v_k t^\omega) = \omega(v_i, v_i(i + 1)) + Val(v_{i+1}) \leq max_{(v_i, v') \in E} \omega(v_i, v_i(i + 1)) + Val(v_{i+1}) = Val(v_i) \).

Next, let \( \sigma_1^{\text{Attr}} \) be memoryless strategy induced by the computation of the attractor: notice that it is possible to construct it directly from the value iteration computation by mapping a vertex \( v \) to one vertex from which \( v \) is first discovered (i.e., its value is first set to a real value different from \( +\infty \)). This strategy ensures to reach the target after at most \( |V| \) steps, thus for all \( v \), \( Val(v, \sigma_1^{\text{Attr}}) \leq W(|V| - 1) \).

25
Before defining the strategy \( \bar{\sigma}_1 \), we introduce the notion of switchable history as follows. A history \( v_1 \cdots v_k \) is switchable if \( v_k \in V_1 \) and \( \text{TP}(v_1 \cdots v_k) \leq \text{Val}(v_1) - \text{Val}(v_k, \sigma_1^\text{Attr}) \). Intuitively a strategy is switchable, if by switching to the attractor strategy, we ensure to get an optimal value.

We define the strategy \( \bar{\sigma}_1 \) as follows: for all history \( v_1 \cdots v_k \) with \( v_k \in V_1 \):

\[
\bar{\sigma}_1(v_1 \cdots v_k) = \begin{cases} 
\sigma_1^\text{Attr}(v_k) & \text{if } v_1 \cdots v_i \text{ is switchable for some } i \\
\bar{\sigma}_1^*(v_k) & \text{otherwise}
\end{cases}
\]

One can easily show that for all play \( v_1v_2 \cdots \) that conforms to \( \bar{\sigma}_1 \) either \( v_1 \cdots v_k \) conforms to \( \sigma_1^\text{Attr} \) if no history is switchable, or there exists \( k \) such that \( v_1 \cdots v_k \) conforms to \( \sigma_1^* \), \( v_1 \cdots v_k \) is a switchable history, and \( v_k v_{k+1} \cdots \) conforms to \( \sigma_1^\text{Attr} \). The following proposition states that the strategy \( \bar{\sigma}_1 \) is optimal.

**Proposition 29.** For all \( v \), \( \text{Val}(v, \bar{\sigma}_1) = \text{Val}(v) \).

**Proof.** Let \( v \in V \) and \( \pi = v_1v_2 \cdots \in \text{Play}(v, \bar{\sigma}_1) \), let us show that \( \text{RTP}(\pi) \leq \text{Val}(v) \). Assume first that there exists \( k \) such that \( v_1 \cdots v_k \) conforms to \( \sigma_1^* \), \( v_1 \cdots v_k \) is a switchable history, and \( v_k v_{k+1} \cdots \) conforms to \( \sigma_1^\text{Attr} \). Thus \( \pi \) reaches the target as \( v_k v_{k+1} \cdots \) does, and \( \text{RTP}(\pi) = \text{TP}(v_1 \cdots v_k) + \text{TP}(v_k v_{k+1} \cdots) \). As \( v_1 \cdots v_k \) is a switchable history, we have \( \text{TP}(v_1 \cdots v_k) \leq \text{Val}(v_1) - \text{Val}(v_k, \sigma_1^\text{Attr}) \), thus \( \text{RTP}(\pi) \leq \text{Val}(v_1) - \text{Val}(v_k, \sigma_1^\text{Attr}) + \text{TP}(v_k v_{k+1} \cdots) \leq \text{Val}(v_1) \).

Assume now that \( v_1v_2 \cdots \) does not contains a switchable prefix and thus \( \pi \) conforms to \( \sigma_1^* \). If \( \pi \) reaches \( t \) then by Proposition 28, \( \text{RTP}(\pi) \leq \text{Val}(v) \). To conclude, let us prove that \( \pi \) reaches \( t \), and by contradiction assume that this is not the case.

First we prove by induction on \( k \), that (\( \ast \)) for all play \( v_1' \cdots v_k' \) that conforms to \( \sigma_1^* \) and does not reach \( t \), \( \text{TP}(v_1' \cdots v_k) \leq \frac{(|V| - 1)W - k}{|V|} \).

If \( k < |V| \) the results is straightforward. For the inductive case, let \( k \geq |V| \) and \( v_1' \cdots v_k' \) that conforms to \( \sigma_1^* \) and does not reach \( t \). Then as \( k \geq |V| \), there exists \( i < j \) such that \( v_i = v_j \). Thus as \( v_1' \cdots v_i' v_{i+1}' \cdots v_k' \) is a play of size \( k - |V| \) that conforms to \( \sigma_1^* \) and does not reach \( t \), we know by induction hypothesis that \( \text{TP}(v_1' \cdots v_i' v_{i+1}' \cdots v_k') \leq (|V| - 1)W - \frac{k}{|V|} = (|V| - 1)W - \frac{1}{|V|} + 1 \).

Furthermore by Proposition 28 we have that \( \text{TP}(v_1' \cdots v_i') - \frac{1}{|V|} \leq -1 \). Thus

\[
\text{TP}(v_1' \cdots v_k') = \text{TP}(v_1' \cdots v_i') + \text{TP}(v_i' \cdots v_j') + \text{TP}(v_j' \cdots v_k')
\]

\[
= \text{TP}(v_1' \cdots v_j' v_{j+1}' \cdots v_k') + \text{TP}(v_i' \cdots v_j') \leq (|V| - 1)W - \frac{k}{|V|},
\]

which concludes the proof of (\( \ast \)).

Now we can get back to raising a contradiction, and for that we show that there exists a switchable history in \( v_1v_2 \cdots \). Take \( k \) be the least index greater that \( 3|V|(|V| - 1)W \) such that \( v_k \in V_1 \) (we know that there exists one, otherwise the vertices \( v_j \) with \( j \geq 3|V|(|V| - 1)W \) would not be in the attractor of \( t \)). We have

\[
\text{TP}(v_1 \cdots v_k) \leq (|V| - 1)W - \frac{k}{|V|} \leq -2(|V| - 1)W \leq \text{Val}(v_1) + \text{Val}(v_k, \sigma_1^\text{Attr}).
\]
Algorithm 2: Computation of optimal strategy for both players in value iteration algorithm for reachability total-payoff games

Input: Reachability total-payoff game \((V, E, \omega, T)\-\text{RTP}\), \(W\) greatest weight in absolute value in the graph

\begin{algorithmic}
\State \textbf{foreach} \(v \in T\) \textbf{do} \(X(v) := 0\)
\State \textbf{foreach} \(v \in V \setminus T\) \textbf{do} \(X(v) := +\infty\)
\Repeat
\State \(X_{\text{pre}} := X\)
\ForAll {\(v \in V_1 \setminus T\)}
\State \(X(v) := \min_{v' \in E(v)} (\omega(v, v') + X_{\text{pre}}(v'))\)
\EndFor
\If {\(X(v) \neq X_{\text{pre}}(v)\)}
\State \(\sigma^*_1(v) := \arg\min_{v' \in E(v)} (\omega(v, v') + X_{\text{pre}}(v'))\)
\EndIf
\If {\(X_{\text{pre}}(v) = +\infty\)}
\State \(\sigma^\text{Attr}_1(v) = \sigma^*_1(v)\)
\EndIf
\EndRepeat
\ForAll {\(v \in V_2 \setminus T\)}
\State \(X(v) := \max_{v' \in E(v)} (\omega(v, v') + X_{\text{pre}}(v'))\)
\EndFor
\If {\(X(v) \neq X_{\text{pre}}(v)\)}
\State \(\sigma^*_2(v) := \arg\max_{v' \in E(v)} (\omega(v, v') + X_{\text{pre}}(v'))\)
\EndIf
\ForAll {\(v \in V \setminus T\)}
\If {\(X(v) < -(|V| - 1) \times W\)} \(X(v) := -\infty\)
\EndIf
\EndFor
\Until {\(X = X_{\text{pre}}\)}
\Return {\(X\)}
\end{algorithmic}

Thus \(v_1 \cdots v_k\) is a switchable prefix, which raises a contradiction. \(\square\)

Notice that \(\tilde{\sigma}_1\) may be more easily implementable than a more general finite-memory strategy, in particular, we may encode the current total-payoff in binary, hence saving some space. We give in Algorithm 2 a way to compute strategies \(\sigma^\text{Attr}_1\) and \(\sigma^*_1\).

Strategies of Player 2 While we have already shown that optimal strategies for Player 1 might require memory, let us show that Player 2 always has a memoryless optimal strategy. This asymmetry stems directly from the asymmetric definition of the game – while Player 1 has the double objective of reaching \(T\) and minimising its cost, Player 2 aims at avoiding \(T\), and if not possible, maximising the cost.

Proposition 30. In all reachability total-payoff game, Player 2 has a memoryless optimal strategy.

Proof. For vertices with value \(+\infty\), we already know a memoryless optimal strategy for Player 2, namely any strategy that remains outside the attractor of the target vertices. For vertices with value \(-\infty\), all strategies are equally bad for Player 2.

We now explain how to define a memoryless optimal strategy \(\sigma^*_2\) for Player 2 in case of a graph containing only finite values. For every finite play \(\pi\) ending in
a vertex \( v \in V_2 \) of Player 2, we let
\[
\sigma^*_2(\pi) = \arg\max_{v' \in E(v)} \left( \omega(v, v') + \text{Val}(v') \right).
\]
This is clearly a memoryless strategy. Let us prove that it is optimal for Player 2, that is, for every vertex \( v \in V \), and every strategy \( \sigma_1 \) of Player 1
\[
\text{RTP}(\text{Play}(v, \sigma_1, \sigma^*_2)) \geq \text{Val}(v).
\]
In case \( \text{Play}(v, \sigma_1, \sigma^*_2) \) does not reach the target set of vertices, the inequality holds trivially. Otherwise, we let \( \text{Play}(v, \sigma_1, \sigma^*_2) = v_0v_1 \cdots v_\ell \cdots \) with \( \ell \) the least position such that \( v_\ell \in T \). If \( \ell = 0 \), i.e., \( v = v_0 \in T \), we have
\[
\text{RTP}(\text{Play}(v, \sigma_1, \sigma^*_2)) = 0 = \text{Val}(v).
\]
Otherwise, let us prove by induction on \( 0 \leq i \leq \ell \) that
\[
\text{RTP}(v_{\ell-i} \cdots v_\ell) \geq \text{Val}(v_{\ell-i}).
\]
This will permit to conclude since
\[
\text{RTP}(\text{Play}(v, \sigma_1, \sigma^*_2)) = \text{RTP}(v_0v_1 \cdots v_\ell) \geq \text{Val}(v_0) = \text{Val}(v).
\]
The base case \( i = 0 \) corresponds to the previous case where the starting vertex is in \( T \). Supposing that the property holds for index \( i \), let us prove it for \( i + 1 \). We have
\[
\text{RTP}(v_{\ell-i-1} \cdots v_\ell) = \omega(v_{\ell-i-1}, v_{\ell-i}) + \text{RTP}(v_{\ell-i} \cdots v_\ell).
\]
By induction hypothesis, we have
\[
\text{RTP}(v_{\ell-i-1} \cdots v_\ell) \geq \omega(v_{\ell-i-1}, v_{\ell-i}) + \text{Val}(v_{\ell-i}). \tag{4}
\]
We now consider two cases:
- If \( v_{\ell-i-1} \in V_2 \setminus T \), then \( v_{\ell-i} = \sigma^*_2(v_0v_1 \cdots v_{\ell-i-1}) \), so that by definition of \( \sigma^*_2 \):
  \[
  \omega(v_{\ell-i-1}, v_{\ell-i}) + \text{Val}(v_{\ell-i}) = \max_{v' \in V \setminus \{v_{\ell-i-1}, v_{\ell-i}\}} \left( \omega(v_{\ell-i-1}, v') + \text{Val}(v') \right).
  \]
  Using Corollary 22 and (4), we obtain
  \[
  \text{RTP}(v_{\ell-i-1} \cdots v_\ell) \geq \text{Val}(v_{\ell-i-1}).
  \]
- If \( v_{\ell-i-1} \in V_1 \setminus T \), then
  \[
  \omega(v_{\ell-i-1}, v_{\ell-i}) + \text{Val}(v_{\ell-i}) \geq \min_{v' \in V \setminus \{v_{\ell-i-1}, v_{\ell-i}\}} \left( \omega(v_{\ell-i-1}, v') + \text{Val}(v') \right).
  \]
  Once again using Corollary 22 and (4), we obtain
  \[
  \text{RTP}(v_{\ell-i-1} \cdots v_\ell) \geq \text{Val}(v_{\ell-i-1}).
  \]
This concludes the proof. \( \square \)

This strategy \( \sigma^*_2 \) can directly be computed along the execution of the value iteration algorithm. This is done in Algorithm 2.
4.4 Optimal plays

While Player 1 may require memory to force Player 2 optimally, the optimal plays built from the previous optimal strategies are indeed non-looping, i.e., do not visit two occurrences of the same vertex.

Proposition 31. In all reachability total-payoff game with finite values, the optimal play \( \text{Play}(v, \sigma_1, \sigma_2^*) \) is non-looping, i.e. never visits twice the same vertex before reaching \( T \).

Proof. Player 2 follows a memoryless strategy. Hence, each loop in the optimal play, if any, is controlled by Player 1. Then, if this loop has a cost at least 0, Player 1 has no incentive to take it, and would indeed gain at not entering this loop. If this loop has a negative cost, then Player 1 would have incentive to repeat a great number of times this loop, to obtain value \( -\infty \), which contradicts the hypothesis on finite values. \( \square \)

4.5 Acceleration techniques

In this section, we propose two acceleration techniques, that can be coupled, in order to speed up the value iteration algorithm. The first one is based on a DAG-decomposition of the graph, and the second is refinement relying on Proposition 31.

Let \( G = (V, E, \omega, T\text{-RTP}) \) be a total-payoff game, and suppose that every vertex \( v_f \) of the target set of vertices \( T \) is absorbing, i.e., \((v_f, v) \in E \) implies \( v = v_f \) and \( \omega(v_f, v_f) = 0 \).

We now suppose given a decomposition of the graph similar to arboreal decompositions introduced in [5], or DAG-decomposition of [1]. Precisely, a function \( c : V \to \mathbb{N} \), mapping each vertex to its component index, is called a DAG-decomposition if it verifies the following conditions:

1. \( c(V) = \{0, 1, \ldots, p\} \) with \( p \in \mathbb{N} \);
2. \( c^{-1}(0) = T \);
3. if \((v, v') \in E\) then \( c(v) \geq c(v') \).

Notice that this implies the acyclicity of the graph with vertices \( c^{-1}(q) \) (for \( 0 \leq q \leq p \)) and edges \((c^{-1}(q_1), c^{-1}(q_2))\) (if there exists \( v_i \in c^{-1}(q_i) \) \( i \in \{1, 2\} \)) such that \((v_1, v_2) \in E\). However, we do not require each subgraph with vertices \( c^{-1}(q) \) to be a strongly connected component, even though it is possible. The directed acyclic graph (DAG) induced by \( c \) defines a topological order over some sets of vertices of the graph, that allows us to decompose the computation performed by the value iteration. The idea is to compute the value iteration algorithm successively on each \( c^{-1}(q) \) by increasing order over \( q \). The intuitive reason is that the value of vertices in such a component only depends on the values of vertices in \( c^{-1}(0, \ldots, q) \) (since there is no path from vertices in \( c^{-1}(q) \) to vertices in \( c^{-1}(r) \) with \( r > q \), by the DAG property). The formal argument is given in Lemma 23. Hence, when computing the values of vertices in \( c^{-1}(q) \), we only compute the value iteration algorithm for vertices in \( c^{-1}(0, \ldots, q) \). We start with initial value being \(+\infty\) for vertices of \( c^{-1}(q) \), and the previously
where quick computations of these sets are possible. The greatest weight optimal values, after at most \(\sum_S\) compute a finite set of possible optimal values are the one of non-looping paths reaching the target (by approximation of such sets of weights, we use it. Below are detailed some classes of vertices of \(c^{-1}\{0, \ldots, q-1\}\)). Since this is above the greatest fixed point, the previous lemma permits us to conclude to the correctness of this acceleration technique.

The second acceleration consists in studying more precisely each component \(c^{-1}(q)\). Having already computed the optimal values \(\text{Val}(v)\) of vertices \(v \in c^{-1}(0, \ldots, q-1)\), we ask an oracle \(\mathcal{O}(q, (\text{Val}(v))_{v \in c^{-1}(0, \ldots, q-1)})\) to pre-compute a finite set \(S_v \in \mathbb{R} \cup \{-\infty, +\infty\}\) of possible optimal values for each vertex \(v \in c^{-1}(q)\). One way to construct such a set \(S_v\) is to consider that possible optimal values are the one of non-looping paths reaching the target (by Proposition 31). Hence, if an oracle has a quick way to compute a finite over-approximation of such sets of weights, we use it. Below are detailed some classes where quick computations of these sets are possible.

Then, during the computation of the value iteration for \(c^{-1}(\{0, \ldots, q\})\), we now update the value by jumping to the next value possible in \(S_v\), rather than the classical update: this results in changes in lines 14 and 15 of Algorithm 1. Once again, the correctness of this acceleration method comes from a repeated use of Lemma 1. Moreover, for component \(c^{-1}(q)\), the number of iterations is bounded by \(\sum_{v \in c^{-1}(q)}|S_v|\) since each vertex will take at most \(|S_v|\) values during the computation.

**Theorem 32.** If a reachability total-payoff game \((V, E, \omega, T\text{-RTP})\) is given as input (possibly with values +\(\infty\) and −\(\infty\)), the accelerated value iteration algorithm for reachability total-payoff games (Algorithm 3) outputs the vector \(\text{Val}\) of optimal values, after at most \(\sum_{v \in V \setminus T}|S_v|\) iterations.

In particular, if \(|S_v|\) is polynomial with respect to the graph (notice that the greatest weight \(W'\) could still be very large) for every vertex \(v\), we obtain a polynomial time algorithm.
Example 33. For the reachability total-payoff game of Fig. 2, we let $c(v_1) = c(v_2) = 1$. Moreover, non-looping paths from $v_1$ (respectively, $v_2$) to the target $v_3$ have weights $-W$ or 0 (respectively, $-W$ or $-1$). In particular, in the computation of the value iteration algorithm, we start from value $(0, -1)$ and directly jump to $(-W, -1)$, and then to $(-W, -W)$, reaching in only 3 steps the convergence.

As an example of a possible non trivial class of graphs for which we obtain a polynomial algorithm by Theorem 32, consider a fixed positive integer $L$ and the set of graphs such that every component $c^{-1}(q)$ uses at most $L$ distinct weights. Then, using again the fact that optimal values are obtained by following non-looping paths, we can consider $S_v$ to be given, for every vertex $v \in c^{-1}(q)$ by

$$S_v = \{k_1w_1 + \cdots + k_Lw_L + w \mid k_1, \ldots, k_L \geq 0, k_1 + \cdots + k_L \leq |c^{-1}(q)| - 1\}$$

where $w_1, \ldots, w_L$ are the weights of edges in $c^{-1}(q)$, and $w$ is a weight of the edge exiting the component added to the optimal value of the vertex previously computed. The size of $S_v$ is bounded by $O(|c^{-1}(q)| \times |V| \times \frac{(|c^{-1}(q)|-1)^{L-1}}{L(L-1)!})$ by [11, Corollary 15.1], where factor $|c^{-1}(q)| \times |V|$ comes from the choice for $w$. This implies that our accelerated value iteration algorithm terminates over such graphs in at most $O\left(\frac{|V|^{L+2}}{L(L-1)!}\right)$ steps. Notice that no restriction is done on the size of weights in the graph.

As a particular case of the previous example, notice that we can deal with graphs where DAG-decomposition with components of bounded size exist (without restrictions on the sets of weights).

5 Efficient algorithms for total-payoff games

We now turn our attention to total-payoff games (without reachability). Our first aim is to show a reduction from total-payoff games to reachability total-payoff games. The obtained game will have size pseudo-polynomial in the size of the original total-payoff game. However, exploiting the efficient algorithm for reachability total-payoff game studied in the previous section, we will be able to obtain a pseudo-polynomial algorithm to solve total-payoff games. Afterwards, we will show how to apply some acceleration techniques in order to not construct the pseudo-polynomial weighted graph, but instead directly run a value iteration algorithm over the original graph: this will result in a much more efficient algorithm, in terms of space complexity. Finally, we will study how to deduce memoryless optimal strategies for both players from our pseudo-polynomial computation.

5.1 Reduction to reachability total-payoff games

We provide a transformation from a total-payoff game $G = (V, E, \omega, TP)$ to a reachability total-payoff game $G^K$ such that the values in $G^K$ are the same as
the values in $G$ (as formalised below). Intuitively the game $G^K$ simulates the following: the players play in $G$, but after each move, Player 1 may propose that the game stops, going into a fresh target $t$; Player 2 can then accept, in which case we reach the target, or refuse, in which case the game continues. After $K$ refusals ($K$ will later be chosen as pseudo-polynomial in the description of the total-payoff game), Player 2 is forced to accept. Structurally $G^K$ consists of a sequence of copies of $G$ along with some new states that we now describe formally.

We let $t$ be a fresh vertex, and, for all $n \geq 1$, we define the reachability total-payoff game $G^n = (V^n, E^n, \omega^n, \{t\}$-$\text{RTP}$) as follows. The sets of vertices are given by

$$V_1^n = \{(v, j) \mid v \in V_1, 1 \leq j \leq n\} \cup \{(\text{in}, v, j) \mid v \in V, 1 \leq j \leq n\} \cup \{t\}$$
$$V_2^n = \{(v, j) \mid v \in V_2, 1 \leq j \leq n\} \cup \{(\text{ex}, v, j) \mid v \in V, 1 \leq j \leq n\}$$

Vertices of the form $(\text{in}, v, j)$ are called interior vertices, and vertices of the form $(\text{ex}, v, j)$ are called exterior vertices. Edges are defined by

$$E^n = \{(v, j), (v', j') \mid (v, v') \in E, 1 \leq j \leq n\}$$
$$\cup \{(\text{in}, v, j), (v, j) \mid v \in V, 1 \leq j \leq n\}$$
$$\cup \{(\text{ex}, v, j), (\text{ex}, v, j) \mid v \in V, 1 \leq j \leq n\}$$
$$\cup \{(\text{ex}, v, j), (v, j - 1) \mid v \in V, 1 < j \leq n\}$$

while weights of edges are defined, for all $(v, v') \in E$ and $1 \leq j \leq n$, by

$$\omega^n((v, j), (v', j)) = \omega(v, v'), \quad \omega^n((\text{ex}, v, j), t) = 0,$$
$$\omega^n((\text{in}, v, j), (v, j)) = 0, \quad \omega^n((\text{ex}, v, j + 1), (v, j)) = 0,$$
$$\omega^n((\text{in}, v, j), (\text{ex}, v, j)) = 0.$$

Example 34. We consider the weighted graph of Fig. 2. The corresponding reachability total-payoff game $G^3$ is depicted in Fig. 3. Indeed, in the following, we could remove interior vertices $(\text{in}, v, j)$ with $v \in V_1$ by putting the choice to stop or continue directly in vertex $(v, j)$. For readability and symmetry reasons, we did not consider this option in the general definition.

The goal of this section is to show that, for $n$ sufficiently large, values of $G$ and $G^n$ are (mostly) the same (see Corollary 40 for a formal statement).

With each finite path in $G^n$, we associate a finite path in $G$, obtained by looking at the sequence of vertices of $V$ appearing inside the vertices of the history. Formally, the projection of a finite path $\pi$ is the sequence $\text{proj}(\pi)$ of vertices of $G$ inductively defined by $\text{proj}(\varepsilon) = \varepsilon$ and for all finite path $\pi$, $v \in V$ and $1 \leq j \leq n$:

$$\text{proj}((\text{in}, v, j)\pi) = \text{proj}(\pi), \quad \text{proj}((\text{ex}, v, j)\pi) = v, \quad \text{proj}((\text{ex}, v, j)) = \varepsilon,$$
$$\text{proj}((\text{ex}, v, j + 1)(v, j)\pi) = \text{proj}((v, j)\pi) = v \text{proj}(\pi).$$

32
Then notice that if $(w, w)$ we associate a vertex $f$ either $\pi_k$ hold trivially. Otherwise, $\pi_k$.

The proof of 2 is a direct consequence of 1. With all vertex $\pi_3$. For all finite path $\pi_3$.

In particular, notice that in the case of a play with prefix $(\text{ex}, v, j)t$, the rest of the play is entirely composed of target vertices $t$, since $t$ is a sink state. For instance, the projection of the history $(v_1, 3)(\text{in}, v_2, 3)(\text{ex}, v_2, 3)(v_2, 2)(\text{in}, v_3, 2)(v_3, 2)(\text{ex}, v_3, 2)t$ of the game $\mathcal{G}^3$ of Fig. 3 is given by $v_1v_2v_3v_3$.

The following lemma relates plays of $\mathcal{G}^n$ with their projection in $\mathcal{G}$, comparing their total-payoff.

**Lemma 35.** The projection mapping satisfies the following properties.
1. If $\pi$ is a finite path in $\mathcal{G}^n$ then $\text{proj}(\pi)$ is a finite in $\mathcal{G}$.
2. If $\pi$ is a play in $\mathcal{G}^n$ that does not reach the target, then $\text{proj}(\pi)$ is a play in $\mathcal{G}$.
3. For all finite path $\pi$, $\text{TP}(\pi) = \text{TP}(\text{proj}(\pi))$.

**Proof.** The proof of 2 is a direct consequence of 1. With all vertex $w \in V^n \setminus \{t\}$, we associate a vertex $f(w)$ as follows:

$$f(v, j) = f(\text{in}, v, j) = f(\text{ex}, v, j) = v.$$ 

Then notice that if $(w, w') \in E^n$ with $w, w' \neq t$, then either $f(w) = f(w')$ or $(f(w), f(w')) \in E$. We now prove 2 and 3 inductively on the size of the partial play $\pi = w_1 \cdots w_k$ of $\mathcal{G}^n$, along with the fact that

4. if $\text{proj}(\pi) \neq \emptyset$ and $w_1 \neq t$ then the first vertex of $\text{proj}(\pi)$ is $f(w_1)$.

If $k = 0$, then $\pi = \text{proj}(\pi) = \varepsilon$ are histories with the same total-payoff. If $k = 1$, either $\text{proj}(\pi) = \varepsilon$ or $\pi = (v, j)$ and $\text{proj}(\pi) = v$; in both cases, the properties hold trivially. Otherwise, $k \geq 2$ and we distinguish several possible prefixes:

- If $\pi = (\text{in}, v, j)\pi'$, then $\text{proj}(\pi) = \text{proj}(\pi')$. Hence, 2 holds by induction hypothesis. If $\text{proj}(\pi)$ is non-empty, so is $\text{proj}(\pi')$. Moreover, the first vertex of $\pi'$ is either $(v, j)$ or $(\text{ex}, v, j)$, so that we can show 4 by induction hypothesis.

Finally, the previous remark shows that the first edge of $\pi$ has necessarily weight 0, so that, $\text{TP}(\pi) = \text{TP}(\pi')$, and 3 also holds by induction hypothesis.
− If \( \pi = (v, j)\pi' \), then \( \text{proj}(\pi) = v \text{proj}(\pi') \) so that 4 holds directly. Moreover, \( \pi' \) is a non-empty history so that \( \pi' = (\text{in}, v', j)\pi'' \) with \( (v, v') \in E \), and \( \text{proj}(\pi') = \text{proj}(\pi'\prime) \). By induction, \( \text{proj}(\pi') \) is a history in \( A \), and it starts with \( v' \) (by 4). Since \( (v, v') \in E \), this shows that \( \text{proj}(\pi) \) is a history. Moreover, \( \text{TP}(\pi) = \omega^\pi((v, j), (\text{in}, v', j)) + \text{TP}(\pi') = \omega(v, v') + \text{TP}(\pi') \). By induction hypothesis, we have \( \text{TP}(\pi') = \text{RTP}(\text{proj}(\pi')) \). Moreover, \( \text{RTP}(\text{proj}(\pi)) = \omega(v, v') + \text{RTP}(\text{proj}(\pi')) \) which concludes the proof of 3.

− If \( \pi = (\text{ex}, v, j)(v, j - 1)\pi' \) then \( \text{proj}(\pi) = v \text{proj}(\pi') = \text{proj}((v, j - 1)\pi') \); this allows us to conclude directly by using the previous case.

− Otherwise, \( \pi = (\text{ex}, v, j)\pi' \), and then \( \text{proj}(\pi) = v \) is a history with total-payoff 0, like \( \pi \), and 4 holds trivially.

The next lemma states that when playing memoryless strategies, one can bound the total-payoff of all prefixes of plays.

**Lemma 36.** Let \( v \in V \), and \( \sigma_1 \) (respectively, \( \sigma_2 \)) be a memoryless strategy for Player 1 (respectively, Player 2) in the total-payoff game \( G \), such that \( \text{Val}(v, \sigma_1) \neq +\infty \) (respectively, \( \text{Val}(v, \sigma_2) = \neq -\infty \)). Then for all history \( \pi \in \text{Play}(v, \sigma_1) \) (respectively, \( \pi \in \text{Play}(v, \sigma_2) \)), \( \text{TP}(\pi) \leq (|V| - 1) \times W \) (respectively, \( \text{TP}(\pi) \geq -(|V| - 1) \times W \)).

**Proof.** We prove the part for Player 1, the other case is similar. The proof proceeds by induction on the size of a partial play \( \pi = v_1 \cdots v_k \) with \( v_1 = v \). If \( k \leq |V| \) then \( \text{TP}(\pi) = \sum_{i=1}^{k-1} \omega(v_i, v_{i+1}) \leq (k - 1) \times W \leq (|V| - 1) \times W \). If \( k \geq |V| + 1 \) then there exists \( i < j \) such that \( v_i = v_j \). Assume by contradiction that \( \text{TP}(v_i \cdots v_j) > 0 \). Then the play \( \pi' = v_1 \cdots v_i \cdots v_j \cdots v_k \) conforms to \( \sigma_1 \) and \( \text{TP}(\pi') = +\infty \) which contradicts \( \text{Val}(v, \sigma_1) \neq +\infty \). Therefore \( \text{TP}(v_i \cdots v_j) \leq 0 \). We have \( \text{TP}(\pi) = \text{TP}(v_1 \cdots v_i) + \text{TP}(v_i \cdots v_j) + \text{TP}(v_j+1 \cdots v_k) \), and since \( v_i = v_j \), \( v_1 \cdots v_i v_{i+1} \cdots v_k \) is a history starting from \( v \) that conforms to \( \sigma_1 \), and by induction hypothesis \( \text{TP}(v_1 \cdots v_i v_{i+1} \cdots v_k) \leq (|V| - 1) \times W \). Then \( \text{TP}(\pi) = \text{TP}(v_1 \cdots v_i v_{i+1} \cdots v_k) + \text{TP}(v_i \cdots v_j) \leq \text{TP}(v_1 \cdots v_i v_{i+1} \cdots v_k) \leq (|V| - 1) \times W \). \( \Box \)

This permits to know bounded finite values \( \text{Val}(v) \) of vertices \( v \) of the game:

**Corollary 37.** For all \( v \in V \), \( \text{Val}(v) \in [- (|V| - 1) \times W, (|V| - 1) \times W] \) ⊃ \{−∞, +∞\}.

**Proof.** From [3], we know that total-payoff games are positionally determined, i.e., there exists two memoryless strategies \( \sigma_1, \sigma_2 \) such that for all \( v \), \( \text{Val}(v) = \text{Val}(v, \sigma_1) = \text{Val}(v, \sigma_2) \). Assume that \( \text{Val}(v) \in \mathbb{R} \). Then since \( \text{Val}(v, \sigma_2) = \text{Val}(v) \neq -\infty \), Lemma 36 shows that all history \( \pi \) that conforms to \( \sigma_2 \) verifies \( \text{TP}(\pi) \geq -(|V| - 1) \times W \), therefore \( \text{Val}(v) \geq -(|V| - 1) \times W \). One can similarly prove that \( \text{TP}(\pi) \leq (|V| - 1) \times W \). \( \Box \)

We now compare values in both games. A first lemma shows, in particular, that \( \text{Val}_{G_n}(v, n) \leq \text{Val}_G(v) \), in case \( \text{Val}_G(v) \neq +\infty \).
Lemma 38. For all $m \in \mathbb{Z}$, $v \in V$, and $n \geq 1$, if $\Val G(v) \leq m$ then $\Val G^m(v, n) \leq m$.

Proof. By hypothesis and using the memoryless determinacy of \cite{7}, there exists a memoryless strategy $\sigma_1$ for Player 1 in $G$ such that $\Val G(v, \sigma_1) \leq m$. Let $\sigma^n_1$ be the strategy in $G^n$ defined, for all history $\pi$, vertex $v'$ and $1 \leq j \leq n$, by

$$\sigma^n_1(\pi(v', j)) = (\text{in}, \sigma_1(\text{proj}(\pi)v'), j),$$

$$\sigma^n_1(\pi(\text{in}, v', j)) = \begin{cases} (v', j) & \text{if } TP(\pi(\text{in}, v', j)) \geq m + 1, \\ (\text{ex}, v', j) & \text{if } TP(\pi(\text{in}, v', j)) \leq m. \end{cases}$$

Intuitively $\sigma^n_1$ simulates $\sigma_1$, and asks to leave the copy when the current total-payoff is less than or equal to $m$. Notice that, by construction of $\sigma^n_1$, $\text{proj}(\pi)$ conforms to $\sigma_1$, if $\pi$ conforms to $\sigma^n_1$.

As a first step, if a play $\pi \in \text{Play}((v, n), \sigma^n_1)$ encounters the target then its value is at most $m$. Indeed, it is of the form $\pi = \pi'(\text{in}, v', j)(\text{ex}, v', j)t^\omega$, and since it conforms to $\sigma^n_1$ we have

$$\text{RTP}(\pi) = TP(\pi'(\text{in}, v', j)(\text{ex}, v', j)t) = TP(\pi'(\text{in}, v', j)) \leq m.$$

Then, assume, by contradiction, that there exists a play $\pi \in \text{Play}((v, n), \sigma^n_1)$ that does not encounter the target. Then, this means that Player 1 does not ask $n+1$ times the ability to exit in $\pi$ (since on the $(n+1)$th time that we jump in an exterior vertex, Player 2 is forced to go to the target). In particular, there exists $0 \leq j \leq n$ such that $\pi$ is of the form $\pi'(v_1, j)(\text{in}, v_2, j)(\text{in}, v_3, j)\cdots(v_k, j)(\text{in}, v_k, j)\cdots$. Since for all $i$, $\sigma^n_1(\pi'(v_i, j)) = (v_i, j)$ we have that $TP(\pi'(v_i, j)(\text{in}, v_2, j)\cdots(\text{in}, v_k, j)) \geq m + 1$. Therefore, since any prefix of $\text{proj}(\pi)$ is the projection of a prefix of $\pi$, Lemma 35 shows that $TP(\text{proj}(\pi)) \geq m + 1 > m$, which raises a contradiction since $\text{proj}(\pi)$ conforms to $\sigma_1$ and $\Val(v, \sigma_1) \leq m$. Hence every play that conforms to $\sigma^n_1$ encounters the target, and, hence, has value at most $m$. This implies that $\Val G^m(v, n) \leq m$. \hfill $\square$

We now turn to the other comparison between $\Val G(v, n)$ and $\Val G(v)$. Since $\Val G(v)$ can be infinite in case the target is not reachable, we have to be more careful. In particular, we show that $\Val G(v, n) \geq \min(\Val G(v), (|V| - 1) \times W + 1)$ holds for large values of $n$. In the following, we let $K = |V| \times (2(|V| - 1) \times W + 1)$.

Lemma 39. For all $m \leq (|V| - 1)W + 1$, $k \geq K$, and vertex $v$, if $\Val G(v) \geq m$ then $\Val G^m(v, k) \geq m$.

Proof. By hypothesis and using the memoryless determinacy of \cite{7}, there exists a memoryless strategy $\sigma_2$ for Player 2 in $G$ such that $\Val G(v, \sigma_2) \geq m$. Let $\sigma^n_2$ be the strategy in $G^n$ defined, for all history $\pi$, vertex $v'$ and $1 \leq j \leq n$, by

$$\sigma^n_2(\pi(v, j)) = (\text{in}, \sigma_2(\text{proj}(\pi)v), j),$$

$$\sigma^n_2(\pi(\text{ex}, v, j)) = \begin{cases} (v, j - 1) & \text{if } TP(\pi) \leq m - 1 \text{ and } j > 1, \\ t & \text{otherwise.} \end{cases}$$

35
Intuitively $\sigma_2^m$ simulates $\sigma_2$, and accepts to go to the target when the current total-payoff is greater than or equal to $m$.

By construction of $\sigma_2^m$, if $\pi$ conforms to $\sigma_2^m$, then $\proj(\pi)$ conforms to $\sigma_2$. From the structure of the weighted graph, we know that for every play $\pi$ of $G^k$, there exists $1 \leq j \leq k$ such that $\pi$ is of the form $\pi_k(\ex, v_k, k)\pi_{k-1}(\ex, v_{k-1}, k-1) \cdots \pi_j(\ex, v_j, j)\pi'$ verifying that: there are no occurrences of exterior vertices in $\pi_k, \ell \leq j$, all vertices in $\pi_\ell$ belong to the $\ell$-th copy of $G$; either $\pi' = t^\omega$ or all vertices of $\pi'$ belong to the $(j+1)$th copy of $G$ (in which case, $j < k$).

We now show that, in $G^k$, $\RTP(\pi) \geq m$ for all play $\pi \in \Play((v, k), \sigma_2^m)$. There are three cases to consider.

1. If $\pi$ does not reach the target, then $\RTP(\pi) = +\infty \geq m$.
2. If $\pi = \pi_k(\ex, v_k, k) \cdots \pi_j(\ex, v_j, j)t^\omega$ and $j > 1$ then,
   \[ \sigma_2^m(\pi_k(\ex, v_0, k) \cdots \pi_j(\ex, v_j, j)) = t. \]
   Thus, using Lemma 33, $\RTP(\pi) = \TP(\pi_k(\ex, v_k, k) \cdots \pi_j(\ex, v_j, j)t) = \TP(\proj(\pi_k(\ex, v_k, k) \cdots \pi_j(\ex, v_j, j)t)) \geq \Val(\pi, \sigma_2) \geq m$.
3. If $\pi = \pi_k(\ex, v_k, k) \cdots \pi_1(\ex, v_1, 1)t^\omega$ assume by contradiction that $\TP(\pi_k(\ex, v_k, k) \cdots \pi_1) \leq m - 1$.

Otherwise, we directly obtain $\RTP(\pi) \geq m$.

Let $v^*$ be a vertex of $A$ that occurs at least $N = [K/V] = 2(|V| - 1) \times W + 1$ times in the sequence $v_1, \ldots, v_k$: such a vertex exists, since otherwise $K \leq (N - 1) \times |V|$ which contradicts the fact that $(N - 1) \times |V| < K$. Let $j_1 > \cdots > j_N$ be a sequence of indices such that $v_{j_i} = v^*$ for all $i$. We give a new decomposition of $\pi$:

\[ \pi = \pi'_1(v_{j_1}, j_1) \cdots \pi'_N(v_{j_N}, j_N)\pi'_{N+1}. \]

Since $\pi$ conforms to $\sigma_2^m$ and according to the assumption, we have that for all $i$,

\[ \TP(\pi'_1(v_{j_1}, j_1) \cdots \pi'_i) \leq m - 1. \]

We consider two cases.

(a) If there exists $\pi'_i$ such that $\TP(\pi_i') \leq 0$ then, let $\proj(\pi'_i) = u_1 \cdots u_\ell$ with $u_1 = u_\ell = v^*$, since $\pi'_i$ conforms to $\sigma_2^m$, $\proj(\pi'_i)$ conforms to $\sigma_2$. Therefore the play

\[ \tilde{\pi} = \proj(\pi'_1(v_{j_1}, j_1) \cdots \pi'_i(v_{j_i}, j_i))(u_1 \cdots u_{\ell-1})^\omega \]

conforms to $\sigma_2$. Furthermore, using again Lemma 33,

\[ \TP(\tilde{\pi}) = \liminf_{n \to +\infty} \left( \TP(\pi'_1(v_{j_1}, i_1) \cdots \pi'_i(v_{j_i}, j_i)) + n\TP(u_1 \cdots u_\ell) \right) \]

and since $\TP(u_1 \cdots u_\ell) = \TP(\pi'_i) \leq 0$, we have

\[ \TP(\tilde{\pi}) \leq \TP(\pi'_1(v_{j_1}, i_1) \cdots \pi'_i(v_{j_i}, j_i)) \leq m - 1. \]

Thus $\tilde{\pi}$ is a play starting from $v$ that conforms to $\sigma_2$ but whose total-payoff is strictly less than $m$, which raises a contradiction. 
(b) If for all $\pi'_i$, $\text{TP}(\pi'_i) \geq 1$ (notice that it is implied by $\text{TP}(\pi'_i) > 0$). From Lemma 36 since $\text{Val}_G(v, \sigma_2) \geq m \neq -\infty$, we know that $\text{TP}(\text{proj}(\pi'_i)) \geq -|V| - 1 \times W$. From Lemma 35 $\text{TP}(\pi'_i) \geq ((|V| - 1) \times W + N$. Therefore

$$\text{TP}(\pi'_i(v, j_1, i_1) \cdots \pi'_N) \geq ((|V| - 1) \times W + 1 \geq m$$

which contradicts the assumption that $\text{TP}(\pi'_i(v, j_1, i_1) \cdots \pi'_N) < m$.

We have shown that $\text{RTP}(\pi) \geq m$ for all play $\pi \in \text{Play}((v, k), \sigma^n_2)$, which implies $\text{Val}_G^K((v, k), \sigma^n_2) \geq m$. Therefore $\text{Val}_G^K((v, k), \sigma^n_2) \geq m$. 

From the two previous lemma, we are ready to relate precisely values in $\mathcal{G}$ and $\mathcal{G}^k$.

**Corollary 40.** For all vertex $v$, for all $k \geq K$,

- if $\text{Val}_G(v) \neq +\infty$ then $\text{Val}_G(v) = \text{Val}_{G^k}((v, k))$;
- if $\text{Val}_G(v) = +\infty$ then $\text{Val}_{G^k}((v, k)) \geq (|V| - 1) \times W + 1$.

**Proof.** Let $v \in V$.

- If $\text{Val}_G(v) = -\infty$, then for all $m$, $\text{Val}_G(v) \leq m$. Thus, by Lemma 38 $\text{Val}_{G^k}(v, k) \leq m$. Therefore $\text{Val}_{G^k}(v, k) = -\infty$.
- If $\text{Val}_G(v) = m \in [-\infty, W]$ then, $m \text{Val}_G(v) \geq m$. Thus, by Lemma 38 and 39 $m \leq \text{Val}_G((v, k)) \geq m$. Therefore $\text{Val}_{G^k}(v, k) = m$.
- If $\text{Val}_G(v) = +\infty$, then $\text{Val}_G(v) \geq (|V| - 1) \times W + 1$. Thus, by Lemma 39 $\text{Val}_{G^k}((v, k)) \geq (|V| - 1) \times W + 1$. Therefore $\text{Val}_{G^k}((v, k)) \geq (|V| - 1) \times W + 1$. □

Notice that in Corollary 40 since we have seen that $\text{Val}_G(v)$ can never be greater than $(|V| - 1) \times W + 1$, we indeed have the following result: for all vertex $v \in V$,

- $\text{Val}_G(v) \neq +\infty$ if and only if $\text{Val}_G(v) = \text{Val}_{G^k}((v, k))$;
- $\text{Val}_G(v) = +\infty$ if and only if $\text{Val}_{G^k}((v, K)) \geq (|V| - 1) \times W + 1$.

To understand well how the values are evolving in the copies of $\mathcal{G}$, we present the following theorem that gives more information on vertices of value $+\infty$.

**Proposition 41.** Let $v \in V$ such that $\text{Val}_G(v) = +\infty$, and $\sigma_2$ a memoryless strategy for Player 2 in $\mathcal{G}$ such that $\text{Val}_G(v, \sigma_2) = +\infty$. Then the following holds.

1. For all history $v_1 \cdots v_k \in \text{Play}(v, \sigma_2)$, if there exists $i < j$ such that $v_i = v_j$ then $\text{TP}(v_1 \cdots v_j) \geq 1$.
2. For all $m \in \mathbb{N}, k \geq m|V| + 1$ and $v_1 \cdots v_k \in \text{Play}(v, \sigma_2)$, $\text{TP}(v_1 \cdots v_k) \geq m - (|V| - 1)W$.
3. For all $m \in \mathbb{N}$ and for all $k \geq (m + (|V| - 1)W)|V| + 1$, $\text{Val}_G(v, k) \geq m$. 

37
We first study in detail the computation performed in lines 4 to 12, transforming 
moreover to process through the steps of the algorithm, without a chance to 
reachability total-payoff game. Notice that this a priori requires to construct the game 
\(G\). Hence, we study a refined version of this algorithm, in order to avoid the 
compliance to \(\omega\). By induction hypothesis, as \(v_{i} < j\) for all \(i \leq j\), \(v_{i} \in V\), then \(v_{i} \in V\), then from Corollary \(40\) we know that 
for all \(k \geq K\), \(\text{Val}_{G_{k}}(v, k) = \text{Val}_{G}(v)\). If \(\text{Val}_{G}(v) = +\infty\), then we know from 
Proposition \(41\) that the sequence \((\text{Val}_{G_{j}}(v, j))_{j \in \mathbb{N}}\) converges towards \(+\infty\). \(\square\)

One can finally show that \(\text{Val}_{G}\) is the limit of the sequence \((\text{Val}_{G_{j}}(v, j))_{v \in V})_{j \in \mathbb{N}}\).

### Theorem 42

The sequence \((\text{Val}_{G_{j}}(v, j))_{v \in V})_{j \in \mathbb{N}}\) converges towards \(\text{Val}_{G}\).

**Proof.** Let \(v \in V\). If \(\text{Val}_{G}(v) < +\infty\), then from Corollary \(40\) we know that 
for all \(k \geq K\), \(\text{Val}_{G_{k}}(v, k) = \text{Val}_{G}(v)\). If \(\text{Val}_{G}(v) = +\infty\), then we know from 
Proposition \(41\) that the sequence \((\text{Val}_{G_{j}}(v, j))_{j \in \mathbb{N}}\) converges towards \(+\infty\). \(\square\)

#### 5.2 Value iteration algorithm for total-payoff games

As a first solution, we may now apply the value iteration algorithm on the 
reachability total-payoff game \(G^{K}\) in order to solve it and map the values back 
in game \(G\). Notice that this a priori requires to construct the game \(G^{K}\), and 
moreover to process through the \(K\) steps of the algorithm, without a chance to 
be able to stop before.

Hence, we study a refined version of this algorithm, in order to avoid the 
construction of \(G^{K}\) and make it possible for the algorithm to stop as soon as 
possible. For that, we consider the value iteration algorithm for total-payoff 
games presented in Algorithm \(4\) and show its correctness and complexity. To 
ease our explanations, \(Y^{j}\) will denote the value of variable \(Y\) at the beginning of the 
jth iteration of the external loop, starting with \(Y^{0}(v) = -\infty\) for all vertex \(v\). 
We first study in detail the computation performed in lines \(4\) to \(12\) transforming 
a vector \(Y\) into another vector, to which we then apply the operation of line \(13\).
Algorithm 4: A value iteration algorithm for total-payoff games

Input: Total-payoff game $\mathcal{G} = \langle V, E, \omega, TP \rangle$

1. foreach $v \in V$ do $Y(v) := -\infty$
2. repeat
3.   $Y_{\text{pre}} := Y$
4.   foreach $v \in V$ do $Y(v) := \max(0, Y(v))$
5.   foreach $v \in V$ do $X(v) := +\infty$
6. repeat
7.   $X_{\text{pre}} := X$
8.   foreach $v \in V_1$ do $X(v) := \min_{v' \in E(v)} \omega(v, v') + \min(X_{\text{pre}}(v'), Y(v'))$
9.   foreach $v \in V_2$ do $X(v) := \max_{v' \in E(v)} \omega(v, v') + \min(X_{\text{pre}}(v'), Y(v'))$
10. foreach $v \in V$ such that $X(v) < -(|V| - 1) \times W$ do $X(v) := -\infty$
11. until $X = X_{\text{pre}}$
12. $Y := X$
13. foreach $v \in V$ such that $Y(v) > (|V| - 1) \times W$ do $Y(v) := +\infty$
14. until $Y = Y_{\text{pre}}$
15. return $Y$

Let us define an extended game. To the original total-payoff game $\mathcal{G} = \langle V, E, \omega, TP \rangle$ and to every vector $Y \in \mathbb{Z}_\infty^V$, we associate the reachability total-payoff game $\mathcal{G}_Y = \langle V', E', \omega_Y, \{t\}$-RTP $\rangle$ as follows. The sets of vertices are given by

$$V_1' = V_1 \uplus \{(\text{in}, v) \mid v \in V\} \uplus \{t\} \quad \text{and} \quad V_2' = V_2.$$

As in the previous section, vertices of the form $(\text{in}, v)$ are called interior vertices. Edges are defined by

$$E' = \{(v,(\text{in},v')) \mid (v, v') \in E\} \uplus \{(\text{in}, v), v \in V\} \uplus \{(t, t)\}$$

while weights of edges are defined, for all $(v, v') \in E$, by

$$\omega_Y(v, (\text{in}, v')) = \omega_Y(v, v'), \quad \omega_Y((\text{in}, v), t) = \max(0, Y(v)), \quad \omega_Y((\text{in}, v), v) = 0.$$

Notice that only the weight function depends on $Y$. For instance, the game $\mathcal{G}_Y$ associated with the total-payoff game $\mathcal{G}$ of Fig. 2 resembles a copy of $\mathcal{G}$ in the game $\mathcal{G}_j$ of the previous section. More, precisely, from the values $(\text{Val}_{\mathcal{G}_j}(v, j))_{v \in V}$ in the $j$th copy, we can
deduce the values in the \((j + 1)\)th copy by an application of operator \(H\):

\[
(\text{Val}_{G^{j+1}}(v, j+1))_{v \in V} = H((\text{Val}_{G^j}(v, j))_{v \in V}).
\]

Although the 0th copy is not defined, we abuse the notation and set \(\text{Val}_{G^0}(v, 0) = -\infty\), which still coforms to the above equality. Furthermore, due to the structure of the game \(G^j\) notice that for all \(j \leq j'\), \(\text{Val}_{G^j}(v, j) = \text{Val}_{G^{j'}}(v, j)\).

Notice the absence of exterior vertices \((ex, v', j)\) in game \(G_Y\), replaced by the computation of the maximum between 0 and \(X(v')\) on the edge towards the target. Before proving the correctness of Algorithm 4, we prove several interesting properties of operator \(H\).

**Proposition 43.** \(H\) is a monotonic operator.

**Proof.** For every vector \(Y \in \mathbb{Z}^V_{\infty}\), let \(F_Y\) be the operator associated with the reachability total payoff game as defined in Section 4, i.e., for all \(X \in \mathbb{Z}^{V'}_{\infty}\), and for all \(v_1 \in V'\)

\[
F_Y(X)(v_1) = \begin{cases} 
\min_{v_2 \in E'(v_1)} (\omega_Y(v_1, v_2) + X(v_2)) & \text{if } v \in V' \setminus \{t\} \\
\max_{v_2 \in E'(v_1)} (\omega_Y(v_1, v_2) + X(v_2)) & \text{if } v \in V \\
0 & \text{if } v_1 = t.
\end{cases}
\]

We know from Corollary 22 that \(\text{Val}_{G^j}\) is the greatest fixed point of \(F_Y\). Consider now two vectors \(Y, Y' \in \mathbb{Z}^V_{\infty}\) such that \(Y \preceq Y'\).

First, notice that for all \(X \in \mathbb{Z}^{V'}_{\infty}\):

\[
F_Y(X) \preceq F_{Y'}(X). \tag{5}
\]

Indeed, to get the result it suffices to notice that for all \(v_1, v_2 \in V', \omega_Y(v_1, v_2) \leq \omega_{Y'}(v_1, v_2)\).

Consider then the vector \(X_0\) defined by \(X_0(v_1) = +\infty\) for all \(v_1 \in V'\). From (5), we have that \(F_Y(X_0) \preceq F_{Y'}(X_0)\), then a simple induction shows
that for all \( i \), \( F^Y_i(X_0) \preceq F^Y_{i+1}(X_0) \). Thus, since \( \text{Val}_{G_Y} \) (respectively, \( \text{Val}_{G_Y'} \)) is the greatest fixed point of \( F_Y \) (respectively, \( F_Y' \)), we have \( \text{Val}_{G_Y} \preceq \text{Val}_{G_Y'} \). As a consequence \( \mathcal{H}(Y) \preceq \mathcal{H}(Y') \).

Notice that \( \mathcal{H} \) may not be Scott-continuous, as shown in the following example.

**Example 44.** Let \( G \) be the total-payoff game containing one vertex \( v \) of Player 1 and a self loop of weight \(-1\) (as depicted in Fig. 5). For all \( x < +\infty \), in the reachability total-payoff game \( G_x \), \( v \) has value \(-\infty \), indeed one can take the loop an arbitrary number of times before reaching the target, ensuring a value arbitrary low. Therefore if we take an increasing sequence \( x_0 < x_1 < \cdots \) of real numbers, \( \mathcal{H}(x_i)(v) = -\infty \) for all \( i \), thus the limit of the sequence \( (\mathcal{H}(x_i))_{i \geq 0} \) is \(-\infty \). However the limit of the sequence \( (x_i)_{i \geq 0} \) is \(+\infty \) and \( \mathcal{H}(+\infty)(v) = +\infty \) (indeed any strategy has to go to the target, paying a cost \(+\infty \)). Thus \( \mathcal{H} \) is not continuous.

In particular, we may not use the Kleene sequence, as we have done for reachability total-payoff games, to conclude to the correctness of our algorithm. However, we will show that the sequence \( (Y_j)_{j \geq 0} \) indeed converges towards the least fixed point of \( \mathcal{H} \), that is the vector of values of the total-payoff game. We show that this vector is a pre-fixed point of \( \mathcal{H} \), and infers from it that it is the least fixed point of \( \mathcal{H} \). We start with a technical lemma that is useful in the subsequent proof.

**Lemma 45.** Let \( \sigma_1 \) be a strategy for Player 1 in \( G \), and \( \pi = v_1 \cdots v_i \) a history that conforms to \( \sigma_1 \). Then:

\[
\text{TP}(v_1 \cdots v_i) + \text{Val}_G(v_i) \preceq \text{Val}_G(v_1, \sigma_1).
\]

**Proof.** Let \( \sigma_2 \) be an optimal strategy for Player 2 and denote by \( v_i v_{i+1} v_{i+2} \cdots \) the play \( \text{Play}(v_i, \sigma_1, \sigma_2) \). Since \( \sigma_2 \) is optimal, \( \text{TP}(v_i v_{i+1} v_{i+2} \cdots) \preceq \text{Val}_G(v_i) \). Furthermore notice that \( v_1 \cdots v_i v_{i+1} \cdots \) conforms to the strategy \( \sigma_1 \), therefore \( \text{TP}(v_1 v_2 \cdots v_i v_{i+1} \cdots) \preceq \text{Val}_G(v_1, \sigma_1) \). Thus:

\[
\text{TP}(v_1 \cdots v_i) + \text{Val}_G(v_i) \preceq \text{TP}(v_1 \cdots v_i) + \text{TP}(v_i v_{i+1} \cdots) \\
\preceq \text{TP}(v_1 v_2 \cdots v_i v_{i+1} \cdots) \preceq \text{Val}_G(v_1, \sigma_1). \tag*{\Box}
\]

**Proposition 46.** \( \text{Val}_G \) is a pre-fixed point of \( \mathcal{H} \), i.e., \( \mathcal{H}(\text{Val}_G) \preceq \text{Val}_G \).
Proposition 47. Before the $j$th iteration of the external loop of Algorithm 4, we have $\text{Val}_{\mathcal{G}_v}(v, j) \leq Y^j(v) \leq \text{Val}_{\mathcal{G}}(v)$ for all vertex $v \in V$.

Proof. For $j = 0$, we have $Y^0(v) = -\infty = \text{Val}_{\mathcal{G}_v}(v, 0)$ for all vertex $v \in V$.

Suppose then that the invariant holds for $j \geq 0$. We know that $\text{Val}_{\mathcal{G}_v}(v, j + 1) = \mathcal{H}((\text{Val}_{\mathcal{G}_v}(v', j))_{v' \in V})$. Moreover, after the assignment of line 14, variable $Y$ contains $\mathcal{H}(Y^j)$. The operation performed on line 15 only
increases the values of vector $Y$, so that at the end of the $j$th iteration, we have $\mathcal{H}(Y^j) \preceq Y^{j+1}$. Since $\mathcal{H}$ is monotonous, and by the invariant at step $j$, we obtain

$$Val_G,j+1(v, j+1) = \mathcal{H}((Val_G,(v', j))_{v' \in V}) \preceq \mathcal{H}(Y^j) \preceq Y^{j+1}.$$  

Moreover, using again the monotony of $\mathcal{H}$ and Proposition 46, we have

$$H(Y^j) \preceq H(Val_G) \preceq Val_G.$$  

A closer look at line 13 shows that $H(Y^j)$ and $Y^{j+1}$ coincide over vertices $v$ such that $H(Y^j)(v) \leq (|V| - 1) \times W$, and otherwise $Y^{j+1}(v) = +\infty$. Hence, if $H(Y^j)(v) \leq (|V| - 1) \times W$, we directly obtain $Y^{j+1}(v) = H(Y^j)(v) \leq Val_G(v)$. Otherwise, we know that $Val_G(v) > (|V| - 1) \times W$. By Corollary 39 we know that $Val_G(v) = +\infty$, so that $Y^{j+1}(v) = +\infty = Val_G(v)$. In the overall, we have proved

$$Val_G,j+1(v, j+1) \preceq Y^{j+1}(v) \preceq Val_G(v) \quad \Box$$

We conclude that

**Theorem 48.** If a total-payoff game $G = \langle V, E, \omega, TP \rangle$ is given as input, the value iteration algorithm for total-payoff games (Algorithm 4) outputs the vector $Val_G$ of optimal values, after at most $K = |V| \times (2(|V| - 1) \times W + 1)$ external iterations.

**Proof.** For $j = K$ (remember that $K$ was defined in the previous section), the invariant of Proposition 47 becomes

$$Val_G,K(v, K) \preceq Y^K(v) \preceq Val_G(v)$$

for all vertices $v \in V$. Notice that the iteration may have stopped before iteration $K$, in which case the sequence $(Y^j)_{j \geq 0}$ may be considered as stationary. In case $Val_G(v) \neq +\infty$, Corollary 40 proves that $Val_G,K(v, K) = Val_G(v)$, so that we have $Y^K(v) \preceq Val_G(v)$. In case $Val_G(v) = +\infty$, Corollary 40 shows that $Val_G,K(v, K) > (|V| - 1) \times W$: by the operation performed at line 13 we obtain that $Y^K(v) = +\infty = Val_G(v)$.

Hence, $K = |V| \times (2(|V| - 1) \times W + 1)$ is an upper bound on the number of iterations before convergence of Algorithm 4 and moreover, at the convergence, the algorithm outputs the vector of optimal values of the total-payoff game.  \Box

Notice finally that acceleration methods presented in Algorithm 3 for reachability total-payoff games are still applicable in the setting of total-payoff games, both in the external and the internal loops. In particular, if we consider the class of total-payoff games $G$ such that, for all vector $X$, $G_X$ admits a decomposition in components (see Section 4.5) where each component uses only a bounded number of distinct weights (without taking into account the weights of $X$), we easily show that the accelerated version of our algorithm runs in polynomial time over this game.
5.3 Optimal strategies

We now conclude this section with the study of optimal strategies in the total-payoff game. We consider a total-payoff game $G$, and denote by $Y^*$ the vector $\text{Val}_G$ of optimal values. We then show how to lift optimal strategies in the reachability total-payoff game $G_{Y^*}$ to the total-payoff game $G$. For that purpose, let us suppose that every vertex of $G$ has value different from $-\infty$. Notice first that for vertices of value $+\infty$, Player 1 can play arbitrarily without harm. We now focus on vertices with finite values. Thanks to Proposition 26 and especially the discussion after this proposition, we know the existence of a couple a memoryless strategies $(\sigma^*_1, \sigma^{\text{Attr}}_1)$ permitting to reconstruct an optimal strategy for Player 1 in the game $G_{Y^*}$. We consider in particular the projection on $V$ of strategy $\sigma^*_1$ and denote by $\overline{\sigma}$ this memoryless strategy over $G$. Precisely, it is defined for all history $\pi$ and vertex $v \in V_1$ by $\overline{\sigma}(\pi v) = v'$ if $\sigma^*_1(v) = (\text{in}, v')$.

**Theorem 49.** Let $G$ be a total-payoff game where all vertices have a finite value. The memoryless strategy $\overline{\sigma}$ extracted from the game $G_{Y^*}$ as explained above is optimal for Player 1.

**Proof.** Without loss of generality, we can suppose that strategy $\sigma^{\text{Attr}}_1$ is only called in interior vertices. Indeed, notice that if the switch between the two strategies is performed in a vertex $v$ of $G_{Y^*}$, the attractor may choose to go in any vertex $(\text{in}, v')$ with $(v, v') \in E$. The switch has been possible because the total-payoff of the current history is at most $\text{Val}_{G_{Y^*}}(v_0) - \text{Val}(v, \sigma^{\text{Attr}}_1)$ (where $v_0$ is the initial vertex of the play): hence, taking a step further after $v$ does not violate this property. Since $\sigma^{\text{Attr}}_1$ can be taken arbitrary, we may also suppose that, in interior vertex $(\text{in}, v')$, the strategy is to go directly to the target, hence with a weight $\max(0, Y^*(v')) \geq 0$. In particular, notice that $\text{Val}((\text{in}, v'), \sigma^{\text{Attr}}_1) \geq 0$, so that, when the switch is performed the current history has a total-payoff at most $\text{Val}_{G_{Y^*}}(v_0)$ (where $v_0$ is again the initial vertex of the play).

Notice that if strategy $\sigma^*_1$ himself chooses to go to the target, because it leads to an optimal strategy in $G_{Y^*}$, we are sure that the total-payoff of the current history is at most $\text{Val}_{G_{Y^*}}(v_0) - \max(0, Y^*(v')) \leq \text{Val}_{G_{Y^*}}(v_0)$. Moreover, because the pair $(\sigma^*_1, \sigma^{\text{Attr}}_1)$ defines an optimal strategy (since all vertex has a value different from $+\infty$), every play starting in $v_0$ and following this strategy is reaching the target (either by $\sigma^*_1$ directly or because of the switch to $\sigma^{\text{Attr}}_1$).

Consider now a vertex $v_0$ of $G$ (of finite value), and any play $\pi = v_0v_1 \cdots \in \text{Play}_{G_{Y^*}}(v_0, \overline{\sigma})$. By injecting the play in $G_{Y^*}$, adding the passage through interior vertices, there must exist a position in the play where either a switch wants to be performed, or the target is ready to be reached by $\sigma^*_1$. Denote by $i_1$ the index corresponding in $\pi$. Then, we know that the history $\pi[i_1]$ has a total-payoff (equal to its mapping in the game $Y^*$) at most $\text{Val}_{G_{Y^*}}(v_0)$. Let us then consider the play $\pi_1$ extracted from $\pi$ starting from $v_{i_1+1}$.

\[ \text{The very same reasoning permits } \]

---

\[ \text{Notice that history } \pi[i_1] \text{ ends in vertex } v_{i_1}, \text{ but we start again from } v_{i_1+1} \text{ on purpose, in order to ensure that we are constructing an increasing sequence of positions in the play.} \]
to prove that there exists an index $i_2 \geq i_1 + 1$ such that $\pi_{i_2}$ has a total-payoff at most $\text{Val}_{G^\star}(v_{i_1+1})$. The history $\pi_{i_2}$ then has a payoff verifying

$$TP(\pi_{i_2}) = TP(\pi[i_1 + 1]) + TP(\pi_{i_2}) \leq TP(\pi[i_1 + 1]) + \text{Val}_{G^\star}(v_{i_1+1}).$$

By Lemma 45, knowing that the couple $(\sigma^*, \sigma^*_1)$ defines an optimal strategy in $G^\star$, and using that total-payoffs are transferred from $G$ to $G^\star$, this implies that

$$TP(\pi_{i_2}) \leq \text{Val}_{G^\star}(v_0).$$

By applying this reasoning inductively, we can construct an increasing sequence $(i_j)_{j \geq 1}$ such that for all $j$, $TP(\pi[i_j]) \leq \text{Val}_{G^\star}(v_0)$. This implies that $TP(\pi)$, as a limit inferior of total-payoffs of prefixes of $\pi$, is at most $\text{Val}_{G^\star}(v_0)$. By the proof of Theorem 15, since $v_0$ has a finite value in $G$, we know that $\text{Val}_{G^\star}(v_0) = \text{Val}_G(v_0)$. So that we have proved that every play from $v_0$ following $\pi$ has a total-payoff bounded above by $\text{Val}_G(v_0)$, so that $\text{Val}_G(v_0, \pi) \leq \text{Val}_G(v_0)$, which proves that $\pi$ is an optimal strategy for Player 1.

Notice that $\sigma^*$, and hence $\pi$, can be computed during the last iteration of the value iteration algorithm, as explained in Algorithm 2. Adaptations can be done in order to also deal with vertices of value $-\infty$. Indeed, in $G^\star$, there will be strategies for Player 1 allowing him to reach values as low as possible. In $G$, this turns to a strategy such that total-payoffs of prefixes of plays following this strategy can be made as low as possible, leading to a total-payoff $-\infty$.

A similar construction can also be done to compute the optimal strategy of Player 2. Note however that it is not correct to simply consider any optimal strategy of Player 2 in $G^\star$, and projecting it to $G$. A counter-example is given by the game of Fig. 6. Indeed, $v_1$ has optimal value $-1$, $v_2$ optimal value 1 and $v_3$ optimal value 0. In the reachability total-payoff game $G^\star$, a strategy for Player 2 going from $v_1$ to $(in, v_2)$ (no choices for the other vertices) would be optimal, but the strategy choosing to go from $v_1$ to $v_2$ in $G$ leads to the value $-2$ in $v_1$, and not $-1$.

References

1. Dietmar Berwanger, Anuj Dawar, Paul Hunter, Stephan Kreutzer, and Jan Obdrzálek. The DAG-width of directed graphs. *Journal of Combinatorial Theory, Series B*, 102(4):900–923, 2012.

2. Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdziński. Mean-payoff parity games. In *Proceedings of the 20th Annual Symposium on Logic in Computer Science (LICS’05)*, pages 178–187. IEEE Computer Society Press, 2005.
3. Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, 1979.

4. Jerzy Filar and Koos Vrieze. *Competitive Markov Decision Processes*. Springer, 1996.

5. Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. Quantitative languages defined by functional automata. In *Proceedings of the 23rd International Conference on Concurrency theory (CONCUR’12)*, volume 7454 of *Lecture Notes in Computer Science*, pages 132–146. Springer, 2012.

6. Thomas Martin Gawlitza and Helmut Seidl. Games through nested fixpoints. In *Proceedings of the 21st International Conference on Computer Aided Verification (CAV’09)*, volume 5643 of *Lecture Notes in Computer Science*, pages 291–305. Springer, 2009.

7. Hugo Gimbert and Wieslaw Zielonka. When can you play positionally? In *Proceedings of the 29th International Conference on Mathematical Foundations of Computer Science (MFCS’04)*, volume 3153 of *Lecture Notes in Computer Science*, pages 686–698. Springer, 2004.

8. Thor Johnson, Neil Robertson, P. D. Seymour, and Robin Thomas. Directed treewidth. *Journal of Combinatorial Theory, Series B*, 82:138–155, 2001.

9. Leonid Khachiyan, Endre Boros, Konrad Borys, Khaled Elbassioni, Vladimir Gurvich, Gabor Rudolf, and Jihui Zhao. On short paths interdiction problems: Total and node-wise limited interdiction. *Theory of Computing Systems*, 43:204–233, 2008.

10. Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.

11. Melvyn B. Nathanson. *Elementary Methods in Number Theory*, volume 195 of *Graduate Texts in Mathematics*. Springer, 2000.

12. Wolfgang Thomas. On the synthesis of strategies in infinite games. In *Symposium on Theoretical Aspects of Computer Science (STACS’95)*, volume 900 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1995.

13. Uri Zwick and Michael S. Paterson. The complexity of mean payoff games. *Theoretical Computer Science*, 158:343–359, 1996.
A Value iteration for discounted-payoff games running in pseudo-polynomial time

Let \( G = \langle V, E, \omega, \text{DP}_\lambda \rangle \) be a \( \lambda \)-discounted-payoff game with \( 0 < \lambda < 1 \) a rational.

Let \( v \) be a vertex of \( A \) and \( 0 < a \leq b \) integers such that \( \lambda = \frac{a}{b} \).

Lemma 50. For all vertex \( v \in V \), there exists \( N \in \mathbb{Z} \) such that \( \text{Val}(v) = \frac{N}{D} \), with \( D = b^{|V|} \prod_{j=1}^{|V|} (b^j - a^j) \).

Proof. By [13], there exist memoryless optimal strategies \( \sigma_1 \) and \( \sigma_2 \) for both players. Thus, if we let \( \pi = \text{Play}(v, \sigma_1, \sigma_2) \), we have \( \text{Val}(v) = \text{DP}_\lambda(\pi) \). Furthermore, we know that \( \pi \) is of the form:

\[
\pi = v_0 \cdots v_k(u_0 \cdots u_\ell)^\omega
\]

with \( 0 \leq k, \ell < |V| \).

Therefore the following holds:

\[
\text{Val}(v) = \text{DP}_\lambda(\pi)
\]

\[
= (1 - \lambda) \sum_{i=0}^{k-1} \lambda^i \omega(v_i, v_{i+1}) + \lambda^k \omega(v_k, u_0) +
\]

\[
\lambda^{k+1} \sum_{r=0}^{\infty} \lambda^{r(r+1)} \left( \sum_{i=0}^{\ell-1} \lambda^i \omega(u_i, u_{i+1}) + \lambda^\ell \omega(u_\ell, u_0) \right)
\]

\[
= \frac{b - a}{b} \sum_{i=0}^{k-1} \frac{b^{k-i} \omega(v_i, v_{i+1})}{b^k} + \frac{a^k}{b^k} \omega(v_k, u_0) +
\]

\[
\frac{\lambda^{k+1}}{1 - \lambda^{\ell+1}} \left( \sum_{i=0}^{\ell-1} \frac{b^{r-i} \omega(u_i, u_{i+1})}{b^\ell} + \frac{a^\ell}{b^\ell} \omega(u_\ell, u_0) \right)
\]

\[
= \frac{N_1}{b^{k+1}} + \frac{a^{k+1}b^{\ell+1}}{b^{k+1}(b^{\ell+1} - a^{\ell+1})} \frac{N_2}{b^\ell}
\]

\[
= \frac{N_1}{b^{k+1}} + \frac{a^{k+1}N_2}{b^k(b^{\ell+1} - a^{\ell+1})}
\]

\[
= \frac{N_3}{b^{k+1}(b^{\ell+1} - a^{\ell+1})}
\]

\[
= \frac{N}{b^{|V|} \prod_{j=1}^{|V|} (b^j - a^j)}
\]
where
\[
N_1 = (b - a) \left[ \sum_{i=0}^{k-1} b^k a^i \omega (v_i, v_{i+1}) + a^k w(v_k, u_0) \right],
\]
\[
N_2 = (b - a) \left[ \sum_{i=0}^{\ell-1} b^{\ell} a^i \omega (u_i, u_{i+1}) + a^\ell \omega (u_\ell, u_0) \right],
\]
\[
N_3 = N_1 (b^{\ell+1} - a^{\ell+1}) + a^{k+1} N_2,
\]
\[
N = N_3 b^{\lvert V \rvert - k} \prod_{1 \leq j \leq \lvert V \rvert, j \neq \ell + 1} (b^j - a^j)
\]
are integers.

Lemma 51. For all vertex \( v \in V \) and \( x \in \mathbb{R} \) such that \( \text{Val}(v) - \frac{1}{2D} < x < \text{Val}(v) + \frac{1}{2D} \),
\[
\text{Val}(v) = \left\lfloor D x + \frac{1}{2} \right\rfloor.
\]

Proof. We have
\[
\text{Val}(v) - \frac{1}{2D} < x < \text{Val}(v) + \frac{1}{2D}
\]
that implies
\[
\text{Val}(v) < x + \frac{1}{2D} < \text{Val}(v) + \frac{1}{D}.
\]
Multiplying by \( D \), we obtain
\[
D \text{Val}(v) < Dx + \frac{1}{2} < D \text{Val}(v) + 1.
\]
Since \( D \text{Val}(v) = N \) is an integer, this proves that
\[
D \text{Val}(v) = \left\lfloor Dx + \frac{1}{2} \right\rfloor
\]
which allows us to conclude. \( \square \)

Now define a functional \( \mathcal{F} : \mathbb{R}^V \to \mathbb{R}^V \) as follows: for all \( X \in \mathbb{R}^V \) and \( v \in V \),
\[
\mathcal{F}(X)(v) = \begin{cases} 
\min_{v' \in E(v)} \left( (1 - \lambda) \omega (v, v') + \lambda X(v') \right) & \text{if } v \in V_1 \\
\max_{v' \in E(v)} \left( (1 - \lambda) \omega (v, v') + \lambda X(v') \right) & \text{if } v \in V_2
\end{cases}
\]

By \cite{13}, we know that \( \text{Val} \) is the unique fixed point of \( \mathcal{F} \). Moreover, operator \( \mathcal{F} \) is \( \lambda \)-contracting, since
\[
\lVert \mathcal{F}(X) - \mathcal{F}(Y) \rVert \leq \lambda \lVert X - Y \rVert
\]
for every vectors \( X, Y \in \mathbb{R}^V \), where \( \lVert X \rVert = \sup_{v \in V} |X(v)| \).

For all \( i \), we inductively define \( \text{Val}^i \in \mathbb{R}^V \) as follows: \( \text{Val}^0(v) = 0 \) for all \( v \in V \) and \( \text{Val}^{i+1} = \mathcal{F}(\text{Val}^i) \). Let \( W \) be the maximal absolute value of the weights of the graph.
Lemma 52. For all $i$, $\|\text{Val}^i - \text{Val}\| \leq \lambda^i W$.

Proof. Since $F$ is $\lambda$-contracting and has $\text{Val}$ as a fixed point, we have $\|\text{Val}^i - \text{Val}\| = \|F(\text{Val}^{i-1}) - F(\text{Val})\| \leq \lambda \|\text{Val}^{i-1} - \text{Val}\|$.

By induction, knowing that $\text{Val}^0 = 0$, we have $\|\text{Val}^i - \text{Val}\| \leq \lambda^i \|\text{Val}\|$. Finally, using that $\text{Val}(v)$ is obtained as an infinite sum of weights smaller than $W$ is absolute value, discounted by $\lambda$, we know that $|\text{Val}(v)| \leq (1 - \lambda)W \sum_{i=0}^{\infty} \lambda^i \leq W$.

that allows us to conclude. $\square$

Lemma 53. For all $K \geq \frac{1}{-\log_2 \lambda} \left( \frac{|V|(|V|+3)}{2} \log(b) + \log(W) + 2 \right)$, $\|\text{Val}^K - \text{Val}\| < \frac{1}{2D}$.

Proof. First notice that:

$$D = b^{|V|} \prod_{j=1}^{|V|} (b^j - a^j) \leq b^{|V|} \prod_{j=1}^{|V|} b^j \leq b^{|V|} \prod_{j=1}^{|V|} b^{|V|} \leq b^{\frac{|V|(|V|+3)}{2}}$$

Thus $\frac{|V|(|V|+3)}{2} \log_2 b \geq \log_2 D$. Then,

$$K \geq \frac{1}{-\log_2 \lambda} \times \left( \log_2 D + \log_2 W + \log_2 4 \right) = \log_{1/\lambda}(4DW)$$

Hence,

$$\frac{1}{\lambda} \geq 4DW$$

so that

$$\lambda^K \leq \frac{1}{4DW}$$

and finally

$$\lambda^K W \leq \frac{1}{4D} < \frac{1}{2D}$$

which allows us to conclude by the previous lemma. $\square$

As a consequence of the previous lemmas the following algorithm computes the mapping $\text{Val}$.

1. Initialize $X$ as $X(v) = 0$ for all $v$.
2. Repeat $\left[ -\frac{1}{\log_2 \lambda} \left( \frac{|V|(|V|+3)}{2} \log_2 b + \log_2 W + 2 \right) \right]$ times $X := F(X)$.
3. For all $v \in V$, output $\text{Val}(v) = \frac{|DX(v)+\frac{1}{2}|}{D}$.

This ends up with an algorithm working in time polynomial in the size of the graph, and polynomial in $\lambda$ (since $-1/\log_2 \lambda \sim 1 - \lambda$ when $\lambda \to 1$).