Acta Informatica (2017) 54:85–125
DOI 10.1007/s00236-016-0276-z

ORIGINAL ARTICLE

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games

Thomas Brihaye[1] · **Gilles Geeraerts**[2] · **Axel Haddad**[1] · **Benjamin Monmege**[3]

Received: 30 November 2015 / Accepted: 2 July 2016 / Published online: 12 July 2016

Abstract Quantitative games are two-player zero-sum games played on directed weighted graphs. Total-payoff games—that can be seen as a refinement of the well-studied mean-payoff games—are the variant where the payoff of a play is computed as the sum of the weights. Our aim is to describe the first pseudo-polynomial time algorithm for total-payoff games in the presence of arbitrary weights. It consists of a non-trivial application of the value iteration paradigm. Indeed, it requires to study, as a milestone, a refinement of these games, called min-cost reachability games, where we add a reachability objective to one of the players. For these games, we give an efficient value iteration algorithm to compute the values and optimal strategies (when they exist), that runs in pseudo-polynomial time. We also propose heuristics to speed up the computations.

1 Introduction

Games played on graphs are nowadays a well-studied and well-established model for the computer-aided design of computer systems, as they enable *automatic synthesis* of systems

Part of this work was sponsored by EU FP7 project Cassting.

✉ Benjamin Monmege
benjamin.monmege@lif.univ-mrs.fr

Thomas Brihaye
thomas.brihaye@umons.ac.be

Gilles Geeraerts
gilles.geeraerts@ulb.ac.be

Axel Haddad
axel.haddad@umons.ac.be

1 Université de Mons, Mons, Belgium

2 Université libre de Bruxelles, Brussels, Belgium

3 CNRS, LIF, Aix-Marseille Univ, Marseille, France

that are *correct-by-construction*. Of particular interest are *quantitative games*, that allow one to model precisely *quantitative* parameters of the system, such as energy consumption. In this setting, the game is played by two players on a directed weighted graph, where the edge weights model, for instance, a cost or a reward associated with the moves of the players. Each vertex of the graph belongs to one of the two players who compete by moving a token along the graph edges, thereby forming an infinite path called a *play*. With each play is associated a real-valued *payoff* computed from the sequence of edge weights along the play. The traditional payoffs that have been considered in the literature include total-payoff [12], mean-payoff [9] and discounted-payoff [21]. In this quantitative setting, one player aims at maximising the payoff while the other tries to minimise it. So one wants to compute, for each player, the best payoff that he can guarantee from each vertex, and the associated optimal strategies (i.e. that guarantee the optimal payoff no matter how the adversary is playing).

Such quantitative games have been extensively studied in the literature. Their associated decision problems (*is the value of a given vertex above a given threshold?*) are known to be in NP∩co-NP . Mean-payoff games have arguably been best studied from the algorithmic point of view. A landmark is Zwick and Paterson's [21] pseudo-polynomial time (i.e. polynomial in the weighted graph when weights are encoded in unary) algorithm, using the *value iteration* paradigm that consists in computing a sequence of vectors of values that converges towards the optimal values of the vertices. After a fixed, pseudo-polynomial, number of steps, the computed values are precise enough to deduce the actual values of all vertices. Better pseudo-polynomial time algorithms have later been proposed, e.g., by Björklund and Vorobyov [1], Brim et al. [6], Comin and Rizzi [8], also achieving sub-exponential expected running time by means of randomisation.

In this paper, we focus on *total-payoff games*.[1] Given an infinite play π, we denote by $\pi[k]$ the prefix of π of length k, and by $\mathbf{TP}(\pi[k])$ the (finite) sum of all edge weights along this prefix. The *total-payoff* of π, $\mathbf{TP}(\pi)$, is the inferior limit of all those sums, i.e. $\mathbf{TP}(\pi) = \liminf_{k\to\infty} \mathbf{TP}(\pi[k])$. Compared to mean-payoff (and discounted-payoff) games, the literature on total-payoff games is less extensive. Gimbert and Zielonka [12] have shown that optimal memoryless strategies always exist for both players and the best algorithm to compute the values runs in exponential time [11], and consists in iteratively improving strategies. Other related works include *energy games* where one player tries to optimise its energy consumption (computed again as a sum), keeping the energy level always above 0. Note that it differs in essence from total-payoff games where no condition on the energy level is required: in particular, the optimal total-payoff could be negative, and even $-\infty$, and it is a priori not possible to simply lift all the weights by a constant to solve total-payoff games by solving a related energy games. Moreover, this difference makes difficult to apply techniques solving energy games in the case of total-payoff games. Probabilistic variants of total-payoff games have also been studied, but the weights are restricted to be non-negative [7].

We argue that the total-payoff objective is interesting as a *refinement* of the mean-payoff. Indeed, recall first that the total-payoff is finite if and only if the mean-payoff is null. Then, the computation of the total-payoff enables a finer, two-stage analysis of a game $\mathcal{G}$: (i) compute the mean payoff $\mathbf{MP}(\mathcal{G})$; (ii) subtract $\mathbf{MP}(\mathcal{G})$ from all edge weights, and scale the resulting weights if necessary to obtain integers. At that point, one has obtained a new game $\mathcal{G}'$ with null mean-payoff; (iii) compute $\mathbf{TP}(\mathcal{G}')$ to *quantify the amount of fluctuation around the mean-payoff* of the original game. Unfortunately, so far, no efficient (i.e. pseudo-polynomial time) algorithms for total-payoff games have been proposed, and straightforward adaptations of Zwick and Paterson's value iteration algorithm for mean-payoff do not work,

[1] Note that those games are different from *total-reward games* as studied in [20].

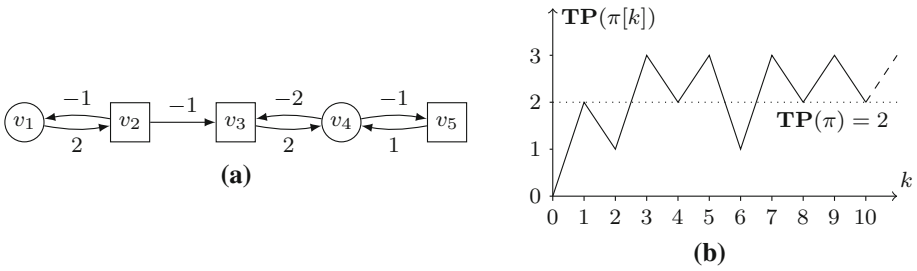

Fig. 1 **a** A total-payoff game, **b** the evolution of the partial sums in π

as we demonstrate at the end of Sect. 2. In the present article, we fill in this gap by introducing the first pseudo-polynomial time algorithm for computing the values in total-payoff games.

Our solution is a non-trivial value iteration algorithm that proceeds through nested fixed points (see Algorithm 2). A play of a total-payoff game is infinite by essence. We transform the game so that one of the players (the minimiser) must ensure a *reachability objective*: we assume that the game ends once this reachability objective has been met. The intuition behind this transformation, that stems from the use of an inferior limit in the definition of the total-payoff, is as follows: in each play π whose total-payoff is *finite*, there is a position ℓ in the play after which all the partial sums $\mathbf{TP}(\pi[i])$ (with $i \geqslant \ell$) will be larger than or equal to the total-payoff $\mathbf{TP}(\pi)$ of π, and infinitely often both will be equal. For example, consider the game depicted in Fig. 1a, where the maximiser player (henceforth called Max) plays with the round vertices and the minimiser (Min) with the square vertices. For both players, the optimal value when playing from v_1 is 2, and the play $\pi = v_1v_2v_3\,v_4v_5\,v_4v_3\,(v_4v_5)^\omega$ reaches this value [i.e. $\mathbf{TP}(\pi) = 2$]. Moreover, for all $k \geqslant 7$: $\mathbf{TP}(\pi[k]) \geqslant \mathbf{TP}(\pi)$, and infinitely many prefixes ($\pi[8]$, $\pi[10]$, $\pi[12]$, ...) have a total-payoff of 2, as shown in Fig. 1b.

Based on this observation, we transform a total-payoff game $\mathcal{G}$, into a new game that has *the same value as the original total-payoff game* but incorporates a reachability objective for Min. Intuitively, in this new game, we allow a new action for Min: after each play prefix $\pi[k]$, he can ask to *stop the game*, in which case the payoff of the play is the payoff $\mathbf{TP}(\pi[k])$ of the prefix. However, allowing Min to stop the game at every moment would not allow us to obtain the same value as in the original total-payoff game: for instance, in the example of Fig. 1a, Min could secure value 1 by asking to stop after $\pi[2]$, which is strictly smaller that the actual total-payoff (2) of the whole play π. So, we allow Max to *veto* to stop the game, in which case both must go on playing. Again, allowing Max to turn down all of Min's requests would be unfair, so we parametrise the game with a natural number K, which is the maximal number of vetoes that Max can play (and we denote by $\mathcal{G}^K$ the resulting game). For the *play* depicted in Fig. 1b, letting $K = 3$ is sufficient: trying to obtain a better payoff than the optimal, Min could request to stop after $\pi[0]$, $\pi[2]$ and $\pi[6]$, and Max can veto these three requests. After that, Max can safely accept the next request of Min, since the total payoff of all prefixes $\pi[k]$ with $k \geqslant 6$ are larger than or equal to $\mathbf{TP}(\pi) = 2$. Our key technical contribution is to show that *for all total-payoff games, there exists a finite, pseudo-polynomial, value of K such that the values in $\mathcal{G}^K$ and $\mathcal{G}$ coincide* (assuming all values are finite in $\mathcal{G}$: we treat the $+\infty$ and $-\infty$ values separately). Now, assume that, when Max accepts to stop the game (possibly because he has exhausted the maximal number K of vetoes), the game moves to a *target vertex*, and stops. By doing so, we effectively reduce the computation of the values in the total-payoff game $\mathcal{G}$ to the computation of the values in the total-payoff game $\mathcal{G}^K$ *with an additional reachability objective* (the target vertex) for Min.

In the following, such refined total-payoff games—where Min *must* reach a designated target vertex—will be called *min-cost reachability games* (*MCR games*). Failing to reach the target vertices is the worst situation for Min, so the payoff of all plays that do not reach the target is $+\infty$, irrespective of the weights along the play. Otherwise, the payoff of a play is the sum of the weights up to the first occurrence of the target. As such, this problem nicely generalises the classical shortest path problem in a weighted graph. In the one-player setting (considering the point of view of Min for instance), this problem can be solved in polynomial time by Dijkstra's and Floyd–Warshall's algorithms when the weights are non-negative and arbitrary, respectively. Khachiyan et al. [13] propose an extension of Dijkstra's algorithm to handle the two-player, non-negative weights case. However, in our more general setting (two players, arbitrary weights), this problem has, as far as we know, not been studied as such, except that the associated decision problem is known to be in NP $\cap$ co-NP [10]. A pseudo-polynomial time algorithm to solve a very close problem, called the *longest shortest path problem* (LSP) has been introduced by Björklund and Vorobyov [1] to eventually solve mean-payoff games. However, because of this peculiar context of mean-payoff games, their definition of the length of a path differs from our definition of the payoff and their algorithm can not be easily adapted to solve our MCR problem. Thus, as a second contribution, we show that a value iteration algorithm enables us to compute in pseudo-polynomial time the values of a MCR game. We believe that MCR games bear their own potential theoretical and practical applications.[2] Those games are discussed in Sect. 3. In addition to the pseudo-polynomial time algorithm to compute the values, we show how to compute optimal strategies for both players and characterise them: there is always a memoryless strategy for the maximiser player, but we exhibit an example (see Fig. 2) where the minimiser player needs (finite) memory. Those results on MCR games are exploited in Sect. 4 where we introduce and prove correct our efficient algorithm for total-payoff games.

Finally, we briefly present our implementation in Sect. 5, using as a core the numerical model-checker PRISM. This allows us to describe some heuristics able to improve the practical performances of our algorithms for total-payoff games and MCR games on certain subclasses of graphs.

2 Quantitative games with arbitrary weights

In this section, we formally introduce the game model we consider throughout the article.

We denote by $\mathbb{Z}$ the set of integers, and $\mathbb{Z}_\infty = \mathbb{Z} \cup \{-\infty, +\infty\}$. The set of vectors indexed by V with values in S is denoted by S^V. We let $\preccurlyeq$ be the pointwise order over $\mathbb{Z}_\infty^V$, where $x \preccurlyeq y$ if and only if $x(v) \leqslant y(v)$ for all $v \in V$.

2.1 Games played on graphs

We consider two-player turn-based games played on weighted graphs and denote the two *players* by Max and Min. A *weighted graph* is a tuple $\langle V, E, \omega \rangle$ where $V = V_{\mathsf{Max}} \uplus V_{\mathsf{Min}}$ is a finite set of vertices partitioned into the sets V_{Max} and V_{Min} of Max and Min respectively, $E \subseteq V \times V$ is a set of *directed edges*, $\omega \colon E \to \mathbb{Z}$ is the *weight function*, associating an integer weight with each edge. In our drawings, Max vertices are depicted by circles; Min vertices by rectangles. For every vertex $v \in V$, the set of successors of v with respect to E is

[2] An example of practical application would be to perform controller synthesis taking into account energy consumption. On the other hand, the problem of computing the values in certain classes of priced timed games has recently been reduced to computing the values in MCR games [3].

denoted by $E(v) = \{v' \in V \mid (v, v') \in E\}$. Without loss of generality, we assume that every graph is deadlock-free, i.e. for all vertices v, $E(v) \neq \emptyset$. Finally, throughout this article, we let $W = \max_{(v,v')\in E} |\omega(v, v')|$ be the greatest edge weight (in absolute value) in the game graph. A *finite play* is a finite sequence of vertices $\pi = v_0 v_1 \dots v_k \in V^*$ such that for all $0 \leqslant i < k$, $(v_i, v_{i+1}) \in E$. A *play* is an infinite sequence of vertices $\pi = v_0 v_1 \dots$ such that every finite prefix $v_0 \dots v_k$, denoted by $\pi[k]$, is a finite play.

The total-payoff of a finite play $\pi = v_0 v_1 \dots v_k$ is obtained by summing up the weights along π, i.e. $\mathbf{TP}(\pi) = \sum_{i=0}^{k-1} \omega(v_i, v_{i+1})$. In the following, we sometimes rely on the mean-payoff to obtain information about total-payoff objectives. The *mean-payoff* computes the average weight of π, i.e. if $k \geqslant 1$, $\mathbf{MP}(\pi) = \frac{1}{k} \sum_{i=0}^{k-1} \omega(v_i, v_{i+1})$, and $\mathbf{MP}(\pi) = 0$ when $k = 0$. These definitions are lifted to infinite plays as follows. The total-payoff of a play π is given by $\mathbf{TP}(\pi) = \liminf_{k\to\infty} \mathbf{TP}(\pi[k])$.[3] Similarly, the mean-payoff of a play π is given by $\mathbf{MP}(\pi) = \liminf_{k\to\infty} \mathbf{MP}(\pi[k])$. Tuples $\langle V, E, \omega, \mathbf{TP}\rangle$ and $\langle V, E, \omega, \mathbf{MP}\rangle$, where $\langle V, E, \omega\rangle$ is a weighted graph, are called *total-payoff* and *mean-payoff* games respectively.

2.2 Strategies and values

A *strategy* for Max (respectively, Min) in a game $\mathcal{G} = \langle V, E, \omega, \mathbf{P}\rangle$ (with $\mathbf{P}$ one of the previous payoffs), is a mapping $\sigma\colon V^* V_{\mathsf{Max}} \to V$ ($\sigma\colon V^* V_{\mathsf{Min}} \to V$) such that for all sequences $\pi = v_0 \dots v_k$ with $v_k \in V_{\mathsf{Max}}$ ($v_k \in V_{\mathsf{Min}}$), it holds that $(v_k, \sigma(\pi)) \in E$. A play or finite play $\pi = v_0 v_1 \dots$ conforms to a strategy σ of Max (respectively, Min) if for all k such that $v_k \in V_{\mathsf{Max}}$ ($v_k \in V_{\mathsf{Min}}$), we have that $v_{k+1} = \sigma(\pi[k])$. A strategy σ is *memoryless* if for all finite plays π, π', we have that $\sigma(\pi v) = \sigma(\pi' v)$ for all $v \in V$. A strategy σ is said to be *finite-memory* if it can be encoded in a deterministic Moore machine, $\langle M, m_0, \mathsf{up}, \mathsf{dec}\rangle$, where M is a finite set representing the memory of the strategy, with an initial memory content $m_0 \in M$, $\mathsf{up}\colon M \times V \to M$ is a memory-update function, and $\mathsf{dec}\colon M \times V \to V$ a decision function such that for every finite play π and vertex v, $\sigma(\pi v) = \mathsf{dec}(\mathsf{mem}(\pi v), v)$ where $\mathsf{mem}(\pi)$ is defined by induction on the length of the finite play π as follows: $\mathsf{mem}(v_0) = m_0$, and $\mathsf{mem}(\pi v) = \mathsf{up}(\mathsf{mem}(\pi), v)$. In this case, we say that $|M|$ is the *size* of the strategy.

For all strategies σ_{Max} and σ_{Min}, for all vertices v, we let $\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})$ be the outcome of σ_{Max} and σ_{Min}, defined as the unique play conforming to σ_{Max} and σ_{Min} and starting in v. Naturally, the objective of Max is to maximise its payoff. In this model of zero-sum game, Min then wants to minimise the payoff of Max. Formally, we let $\mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Max}})$ and $\mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Min}})$ be the respective values of the strategies, defined as (recall that $\mathbf{P}$ is either $\mathbf{TP}$ or $\mathbf{MP}$): $\mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Max}}) = \inf_{\sigma_{\mathsf{Min}}} \mathbf{P}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}}))$ and $\mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Min}}) = \sup_{\sigma_{\mathsf{Max}}} \mathbf{P}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}}))$. Finally, for all vertices v, we let $\underline{\mathsf{Val}}_{\mathcal{G}}(v) = \sup_{\sigma_{\mathsf{Max}}} \mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Max}})$ and $\overline{\mathsf{Val}}_{\mathcal{G}}(v) = \inf_{\sigma_{\mathsf{Min}}} \mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Min}})$ be respectively the *lower* and *upper values* of v. We may easily show that $\underline{\mathsf{Val}}_{\mathcal{G}} \preccurlyeq \overline{\mathsf{Val}}_{\mathcal{G}}$. We say that strategies $\sigma^\star_{\mathsf{Max}}$ of Max and $\sigma^\star_{\mathsf{Min}}$ of Min are optimal if, for all vertices v: $\mathsf{Val}_{\mathcal{G}}(v, \sigma^\star_{\mathsf{Max}}) = \underline{\mathsf{Val}}_{\mathcal{G}}(v)$ and $\mathsf{Val}_{\mathcal{G}}(v, \sigma^\star_{\mathsf{Min}}) = \overline{\mathsf{Val}}_{\mathcal{G}}(v)$ respectively. We say that a game $\mathcal{G}$ is *determined* if for all vertices v, its lower and upper values are equal. In that case, we write $\mathsf{Val}_{\mathcal{G}}(v) = \underline{\mathsf{Val}}_{\mathcal{G}}(v) = \overline{\mathsf{Val}}_{\mathcal{G}}(v)$, and refer to it as the *value* of v in $\mathcal{G}$. If the game is clear from the context, we may drop the index $\mathcal{G}$ from all previous notations. Mean-payoff and total-payoff games are known to be determined, with the existence of optimal memoryless strategies [12,21].

[3] Our results can easily be extended by substituting a lim sup for the lim inf. The lim inf is more natural since we adopt the point of view of the maximiser Max, where the lim inf is the *worst* partial sum seen infinitely often.

2.3 Previous works and contribution

Total-payoff games have been mainly considered as a refinement of mean-payoff games [12]. Indeed, if the mean-payoff value of a game is positive (respectively, negative), its total-payoff value is necessarily $+\infty$ ($-\infty$). When the mean-payoff value is 0 however, the total-payoff is necessarily different from $+\infty$ and $-\infty$, hence total-payoff games are particularly useful in this case, to refine the analysis of the game. Deciding whether the total-payoff value of a vertex is positive can be achieved in NP $\cap$ co-NP. Gawlitza and Seidl [11] refined the complexity to UP $\cap$ co-UP, and values are shown to be effectively computable solving nested fixed point equations with a strategy iteration algorithm working in exponential time in the worst case. Because of this strong relationship between mean- and total-payoff games, we can show that total-payoff games are, in some sense, as hard as mean-payoff games, for which the existence of a (strongly) polynomial time algorithm is a long-standing open question.

In this article, we improve on this state-of-the-art and introduce the first (to the best of our knowledge) pseudo-polynomial time algorithm for total-payoff games. In many cases, (e.g., mean-payoff games), a successful way to obtain such an efficient algorithm is the *value iteration paradigm*. Intuitively, value iteration algorithms compute successive approximations $x_0, x_1, \ldots, x_i, \ldots$ of the game value by restricting the number of turns that the players are allowed to play: x_i is the vector of optimal values achievable when the players play at most i turns. The sequence of values is computed by means of an operator $\mathcal{F}$, letting $x_{i+1} = \mathcal{F}(x_i)$ for all i. Good properties (Scott-continuity and monotonicity) of $\mathcal{F}$ ensure convergence towards its smallest or greatest fixed point (depending on the value of x_0), which, in some cases, is the value of the game.

Let us briefly explain why, unfortunately, a straightforward application of this approach fails with total-payoff games. In our case, the most natural operator $\mathcal{F}$ is such that $\mathcal{F}(x)(v) = \max_{v' \in E(v)}(\omega(v, v') + x(v'))$ for all $v \in V_{\mathsf{Max}}$ and $\mathcal{F}(x)(v) = \min_{v' \in E(v)}(\omega(v, v') + x(v'))$ for all $v \in V_{\mathsf{Min}}$. Indeed, this definition matches the intuition that x_N is the optimal value after N turns. Then, consider the example of Fig. 1a, limited to vertices $\{v_3, v_4, v_5\}$ for simplicity. Observe that there are two simple cycles with weight 0, hence the total-payoff value of this game is finite. Max has the choice between cycling into one of these two cycles. It is easy to check that Max's optimal choice is to enforce the cycle between v_4 and v_5, securing a payoff of -1 from v_4 (because of the lim inf definition of **TP**). Hence, the values of v_3, v_4 and v_5 are respectively 1, -1 and 0. In this game, we have $\mathcal{F}(x) = (2 + x(v_4), \max(-2 + x(v_3), -1 + x(v_5)), 1 + x(v_4))$, and the vector $(1, -1, 0)$ is indeed a fixed point of $\mathcal{F}$. However, it is neither the greatest nor the smallest fixed point of $\mathcal{F}$. Indeed, it is easy to check that, *if* x is a fixed point of $\mathcal{F}$, *then* $x + (a, a, a)$ is also a fixed point, for all constant $a \in \mathbb{Z} \cup \{-\infty, +\infty\}$. If we try to initialise the value iteration algorithm with value $(0, 0, 0)$, which could seem a reasonable choice, the sequence of computed vectors is: $(0, 0, 0), (2, -1, 1), (1, 0, 0), (2, -1, 1), (1, 0, 0), \ldots$ that is not stationary, and does not even contain $(1, -1, 0)$. Notice that $(-\infty, -\infty, -\infty)$ and $(+\infty, +\infty, +\infty)$ are fixed points, so that they do not allow us to find the correct answer too. Thus, it seems difficult to compute the actual game values with an iterative algorithm relying on the operator $\mathcal{F}$, as in the case of mean-payoff games.[4] Notice that, in the previous example, the Zwick and Paterson's algorithm [21] to solve mean-payoff games would easily conclude from the sequence above, since the vectors of interest are then $(0, 0, 0), (1, -0.5, 0.5), (0.33, 0, 0), (0.5, -0.25, 0.25), (0.2, 0, 0), \ldots$ indeed converging towards $(0, 0, 0)$, the mean-payoff values of this game.

[4] In the context of stochastic models like Markov decision processes, Strauch [17] already noticed that in the presence of arbitrary weights, the value iteration algorithm does not necessarily converge towards the accurate value: Puterman [16] gives a more detailed explanation in Ex. 7.3.3.

Instead, as explained in the introduction, we propose a different approach that consists in reducing total-payoff games to MCR games where Min must enforce a reachability objective on top of his optimisation objective. The aim of the next section is to study these games, and we reduce total-payoff games to them in Sect. 4.

3 Min-cost reachability games

In this section, we consider *MCR games*, a variant of total-payoff games where one player has a reachability objective that he must fulfil first, before minimising his quantitative objective (hence the name *min*-cost reachability). Without loss of generality, we assign the reachability objective to player Min, as this will make our reduction from total-payoff games easier to explain. Hence, when the target is not reached along a path, the payoff of this path shall be the worst possible for Min, i.e. $+\infty$. Formally, an MCR game is played on a weighted graph $\langle V, E, \omega\rangle$ equipped with a target set of vertices $T \subseteq V$. The payoff T-**MCR**(π) of a play $\pi = v_0 v_1 \dots$ is given by T-**MCR**$(\pi) = +\infty$ if the play avoids T, i.e. if for all $k \geqslant 0$, $v_k \notin T$, and T-**MCR**$(\pi) = \mathbf{TP}(\pi[k])$ if k is the least position in π such that $v_k \in T$. Lower and upper values are then defined as in Sect. 2.

Using an indirect consequence of Martin's theorem [15], we can show that MCR games are *determined*, i.e. that the upper and lower valus always coincide:

Theorem 1 *MCR games are determined.*

Proof Consider a quantitative game $\mathcal{G} = \langle V, E, \omega, \mathbf{P}\rangle$ and a vertex $v \in V$. We will prove the determinacy result by using the Borel determinacy result of [15]. First, notice that the payoff mapping T-**MCR** is Borel measurable since the set of plays with finite T-**MCR** payoff is a countable union of cylinders. Then, for an integer M, consider Win_M to be the set of plays with a payoff less than or equal to M. It is a Borel set, so that the qualitative game defined over the graph $\langle V, E, \omega\rangle$ with winning condition Win_M is determined. We now use this preliminary result to show our determinacy result.

We fix an MCR game and one of its vertices v, and first consider cases where either the lower or the upper values is infinite. Suppose first that $\underline{\mathsf{Val}}(v) = -\infty$. We have to show that $\overline{\mathsf{Val}}(v) = -\infty$ too. Let M be an integer. Since $\underline{\mathsf{Val}}(v) < M$, we know that for all strategies σ_{Max} of Max, there exists a strategy σ_{Min} for Min, such that $\mathbf{P}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})) \leqslant M$. In particular, Max has no winning strategy in the qualitative game equipped with Win_M as a winning condition, hence, by determinacy, Min has a winning strategy, i.e. a strategy σ_{Min} such that every strategy σ_{Max} of Max verifies $\mathbf{P}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})) \leqslant M$. This exactly means that $\overline{\mathsf{Val}}(v) \leqslant M$. Since this holds for every value M, we get that $\overline{\mathsf{Val}}(v) = -\infty$. The proof goes exactly in a symmetrical way to show that $\overline{\mathsf{Val}}(v) = +\infty$ implies $\underline{\mathsf{Val}}(v) = +\infty$.

Consider then the case where both $\overline{\mathsf{Val}}(v)$ and $\underline{\mathsf{Val}}(v)$ are finite values. For the sake of contradiction, assume that $\underline{\mathsf{Val}}(v) < \overline{\mathsf{Val}}(v)$ and consider a real number r strictly in-between those two values. From $r < \overline{\mathsf{Val}}(v)$, we deduce that Min has no winning strategy from v in the qualitative game with winning condition Win_r. Identically, from $\underline{\mathsf{Val}}(v) < r$, we deduce that Max has no winning strategy from v in the same game. This contradicts the determinacy of this qualitative game. Hence, $\underline{\mathsf{Val}}(v) = \overline{\mathsf{Val}}(v)$. □

Example 2 As an example, consider the MCR game played on the weighted graph of Fig. 2, where W is a positive integer and v_3 is the target. We claim that the values of vertices v_1 and v_2 are both $-W$. Indeed, consider the following strategy for Min: during each of the first W visits to v_2 (if any), go to v_1; else, go to v_3. Clearly, this strategy ensures that the target will

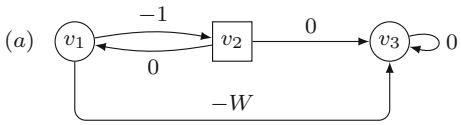

Fig. 2 An MCR game (with W a positive integer and, v_3 the target) where Min needs memory to achieve its optimal strategy

eventually be reached, and that either (i) edge (v_1, v_3) (with weight $-W$) will eventually be traversed; or (ii) edge (v_1, v_2) (with weight -1) will be traversed at least W times. Hence, in all plays following this strategy, the payoff will be at most $-W$. This strategy allows Min to secure $-W$, but he can not ensure a lower payoff, since Max always has the opportunity to take the edge (v_1, v_3) (with weight $-W$) instead of cycling between v_1 and v_2. Hence, Max's optimal choice is to follow the edge (v_1, v_3) as soon as v_1 is reached, securing a payoff of $-W$. The Min strategy we have just given is optimal, and there is *no optimal memoryless strategy* for Min. Indeed, always playing (v_2, v_3) does not ensure a payoff less than or equal to $-W$; and, always playing (v_2, v_1) does not guarantee to reach the target, and this strategy has thus value $+\infty$.

A remark on related work Let us note that [1] introduce the LSP and propose a pseudo-polynomial time algorithm to solve it. However, their definition has several subtle but important differences to ours, such as in the definition of the payoff of a play (equivalently, the length of a path). As an example, in the game of Fig. 2, the play $\pi = (v_1 v_2)^\omega$ (that never reaches the target) has length $-\infty$ in their setting, while, in our setting, $\{v_3\}$-**MCR**$(\pi) = +\infty$. A more detailed comparison of the two definitions is given in "Appendix". Moreover, even if a preprocessing would hypothetically allow one to use the LSP algorithm to solve MCR games, our solution (that has the same worst-case complexity as theirs) is simpler to implement, and we also introduce (see Sect. 5) heuristics that are only applicable to our value iteration solution.

As explained in the introduction of this section, we show how to *solve* those games, i.e. how to compute $\mathsf{Val}(v)$ for all vertices v in pseudo-polynomial time. This procedure will be instrumental to solving total-payoff games. Our contributions are summarised in the following theorem:

Theorem 3 *Let $\mathcal{G} = \langle V, E, \omega, T\text{-}\mathbf{MCR}\rangle$ be an MCR game.*

1. *For all $v \in V$, deciding whether $\mathsf{Val}(v) = +\infty$ can be done in polynomial time.*
2. *For $v \in V$, deciding whether $\mathsf{Val}(v) = -\infty$ is as hard as solving mean-payoff games, in* NP $\cap$ co-NP *and can be achieved in pseudo-polynomial time.*
3. *If $\mathsf{Val}(v) \neq -\infty$ for all vertices $v \in V$, then both players have optimal strategies. Moreover,* Max *always has a memoryless optimal strategy, while* Min *may require finite (pseudo-polynomial) memory in his optimal strategy.*
4. *Computing all values $\mathsf{Val}(v)$ (for $v \in V$), as well as optimal strategies (if they exist) for both players, can be done in (pseudo-polynomial) time $O(|V|^2|E|W)$.*

3.1 Finding vertices with value $+\infty$

To prove the first item of Theorem 3, it suffices to notice that vertices with value $+\infty$ are exactly those from which Min can not reach the target. Therefore the problem reduces to deciding the winner in a classical reachability game, that can be solved in polynomial time [19], using the classical *attractor* construction.

More precisely, let $\mathcal{G} = \langle V, E, \omega, T\text{-}\mathbf{MCR}\rangle$ be an MCR game. Notice that for all plays $\pi = v_0 v_1 \ldots$, T-**MCR**$(\pi) = +\infty$ if and only if $v_k \notin T$ for all $k \geqslant 0$, i.e. π avoids the target.

Then, let us show that the classical attractor technique [19] allows us to compute the set $V_{+\infty} = \{v \in V \mid \mathsf{Val}(v) = +\infty\}$. Recall that the attractor of a set T of vertices is obtained thanks to the sequence $\mathsf{Attr}_0(T), \ldots, \mathsf{Attr}_i(T), \ldots$ where: $\mathsf{Attr}_0(T) = T$; and for all $i \geqslant 0$:

$$\mathsf{Attr}_{i+1}(T) = \mathsf{Attr}_i(T) \cup \{v \in V_{\mathsf{Min}} \mid E(v) \cap \mathsf{Attr}_i(T) \neq \emptyset\} \cup \{v \in V_{\mathsf{Max}} \mid E(v) \subseteq \mathsf{Attr}_i(T)\}.$$

It is well-known that this sequence converges after at most $|V|$ steps to the set $\mathsf{Attr}(T)$ of all vertices from which Min has a memoryless strategy to ensure reaching T. Hence, under our hypothesis, $V_{+\infty} = V \backslash \mathsf{Attr}(T)$. This proves the first item of Theorem 3. Observe that we can safely *remove* from the game graph all vertices v such that $\mathsf{Val}(v) = +\infty$, without changing the values of the other vertices. Hence, we can, when need be, assume that the MCR games we consider contain no vertex with value $+\infty$, as they can be removed by this polynomial-time preprocessing.

In those games, one can construct in polynomial time a memoryless strategy, called *an attractor strategy*, ensuring to reach the target in less than $|V|$ steps from every vertex.

In the following, we assume that all vertices have a value different from $+\infty$. Indeed as described above one can detect in polynomial time the vertices with value $+\infty$ and remove them *without changing the values of the other vertices*.

3.2 Finding vertices with value $-\infty$

To prove the second item, we notice that vertices with value $-\infty$ are exactly those with a value <0 in the mean-payoff game played on the same graph. On the other hand, we can show that every mean-payoff game can be transformed (in polynomial time) into an MCR game such that a vertex has value <0 in the mean-payoff game if and only if the value of its corresponding vertex in the MCR game is $-\infty$. More precisely:

Proposition 4 *1. For all MCR games $\mathcal{G} = \langle V, E, \omega, T\text{-}\mathbf{MCR}\rangle$ where $\mathsf{Val}_{\mathcal{G}}(v) \neq +\infty$ for all v, for all vertices v of $\mathcal{G}$, $\mathsf{Val}_{\mathcal{G}}(v) = -\infty$ if and only if $\mathsf{Val}_{\mathcal{G}'}(v) < 0$, where $\mathcal{G}'$ is the mean-payoff game $\langle V, E, \omega, \mathbf{MP}\rangle$.*

2. Conversely, given a mean-payoff game $\mathcal{G} = \langle V, E, \omega, \mathbf{MP}\rangle$, we can build, in polynomial time, an MCR game $\mathcal{G}'$ such that for all vertices v of $\mathcal{G}$: $\mathsf{Val}_{\mathcal{G}}(v) < 0$ if and only if $\mathsf{Val}_{\mathcal{G}'}(v) = -\infty$.

Proof To prove the first item, consider an MCR game $\mathcal{G} = \langle V, E, \omega, T\text{-}\mathbf{MCR}\rangle$ such that $\mathsf{Val}_{\mathcal{G}}(v) \neq +\infty$ for all $v \in V$, and $\mathcal{G}' = \langle V, E, \omega, \mathbf{MP}\rangle$ the same weighted graph equipped with a mean-payoff objective.

If $\mathsf{Val}_{\mathcal{G}'}(v) < 0$, we know that there is a profile of optimal memoryless strategies $(\sigma^\star_{\mathsf{Max}}, \sigma^\star_{\mathsf{Min}})$ such that the outcome starting in v and following this profile necessarily starts with a finite prefix and then loops in a cycle with a total weight <0. For every $M > 0$, we construct a strategy σ^M_{Min} that ensures in $\mathcal{G}$ a cost less than or equal to $-M$: this will prove that $\mathsf{Val}_{\mathcal{G}}(v) = -\infty$. Since we have assumed that $\mathsf{Val}_{\mathcal{G}}(v) \neq +\infty$ for all v, we know that Min has a strategy to reach the target from all v (for instance, take the attractor strategy described above), by a path of length at most $|V|$. Thus, there exists a bound w and a strategy allowing Min to reach the target from every vertex of $\mathcal{G}'$ with a cost at most w. The strategy σ^M_{Min} of Min is then to follow $\sigma^\star_{\mathsf{Min}}$ until the accumulated cost is less than $-M - w$, at which point it follows his strategy to reach the target. Clearly, for all M, σ^M_{Min} guarantees that Min reaches the target with a cost at most $-M$.

Reciprocally, if $\mathsf{Val}_{\mathcal{G}}(v) = -\infty$, consider $M = |V|W$ and a strategy σ^M_{Min} of Min ensuring a cost less than $-M$, i.e. such that $\mathsf{Val}_{\mathcal{G}}(v, \sigma^M_{\mathsf{Min}}) < -M$. Consider the finitely-branching tree

built from $\mathcal{G}$ by unfolding the game from vertex v and resolving the choices of Min with strategy σ^M_{Min}. Each branch of this tree corresponds to a possible strategy of Max. 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, consider an optimal memoryless strategy $\sigma^\star_{\mathsf{Max}}$ of Max securing a non-negative mean-payoff, that is, $\mathsf{Val}_{\mathcal{G}'}(v, \sigma^\star_{\mathsf{Max}}) \geqslant 0$. Consider the branch of the previous tree where Max follows strategy $\sigma^\star_{\mathsf{Max}}$. Since this finite branch has cost less than $-M = -|V|W < 0$ (W is positive, otherwise the mean-payoff value would be 0), we know for sure that there are two occurrences of the same vertex v' with an in-between weight <0: otherwise, by removing all non-negative cycles, we obtain a play without repetition of vertices, henceforth of length bounded by $|V|$, and therefore of cost at least $-M$. Suppose that $v' \in V_{\mathsf{Max}}$. Then, Min has a strategy σ_{Min} to ensure a negative mean-payoff $\mathsf{Val}_{\mathcal{G}'}(v, \sigma_{\mathsf{Min}}) < 0$: indeed, he simply modifies[5] his strategy so that he always stays in the negative cycle starting in v' (he can do that since σ_{Max} plays a memoryless strategy, so that he can not change his decisions in the cycle), ensuring that, against the optimal strategy $\sigma^\star_{\mathsf{Max}}$ of Max, he gets a mean-payoff being the cost of the cycle. This is a contradiction since Max is supposed to have a strategy ensuring a non-negative mean-payoff from v. Hence, $v' \in V_{\mathsf{Min}}$. But the same contradiction appears in that case since Min can force that it always stays in the negative cycle by modifying his strategy. Finally, we have proved that Max can not have a memoryless strategy securing a non-negative mean-payoff from v. By memoryless determinacy of the mean-payoff games, this ensures that Min has a memoryless strategy securing a negative mean-payoff from v.

Hence, we have shown that $\mathsf{Val}_{\mathcal{G}}(v) = -\infty$ if and only if $\mathsf{Val}_{\mathcal{G}'}(v) < 0$, which concludes the first claim of Proposition 4.

To prove the second item, we reduce mean-payoff games to MCR games as follows. Let $\mathcal{G} = \langle V, E, \omega, \mathbf{MP}\rangle$ be a mean-payoff game. Without loss of generality, we may suppose that the graph of the game is bipartite, in the sense that $E \subseteq V_{\mathsf{Max}} \times V_{\mathsf{Min}} \cup V_{\mathsf{Min}} \times V_{\mathsf{Max}}$.[6] The problem we are interested in is to decide whether $\mathsf{Val}_{\mathcal{G}}(v) < 0$ for a given vertex v. We now construct an MCR game $\mathcal{G}' = \langle V', E', \omega', T'\text{-}\mathbf{MCR}\rangle$ from $\mathcal{G}$. The only difference is the presence of a fresh target vertex $\mathtt{t}$ on top of vertices of V: $V' = V \uplus \{\mathtt{t}\}$ with $T' = \{\mathtt{t}\}$. Edges of $\mathcal{G}'$ are given by $E' = E \cup \{(v, \mathtt{t}) \mid v \in V_{\mathsf{Min}}\} \cup \{(\mathtt{t}, \mathtt{t})\}$. Weights of edges are given by: $\omega'(v, v') = \omega(v, v')$ if $(v, v') \in E$, and $\omega'(v, \mathtt{t}) = \omega'(\mathtt{t}, \mathtt{t}) = 0$. We show that $\mathsf{Val}_{\mathcal{G}}(v) < 0$ if and only if $\mathsf{Val}_{\mathcal{G}'}(v) = -\infty$.

In $\mathcal{G}'$, all values are different from $+\infty$, since Min plays at least every two steps, and has the capability to go to the target vertex with weight 0. Hence, letting $\mathcal{G}'' = \langle V', E', \omega', \mathbf{MP}\rangle$ the mean-payoff game on the weighted graph of $\mathcal{G}'$, by the previous direction, we have that for every vertex $v \in V'$, $\mathsf{Val}_{\mathcal{G}'}(v) = -\infty$ if and only if $\mathsf{Val}_{\mathcal{G}''}(v) < 0$.

To conclude, we prove that for all vertices $v \in V$, $\mathsf{Val}_{\mathcal{G}''}(v) < 0$ if and only if $\mathsf{Val}_{\mathcal{G}}(v) < 0$. If $\mathsf{Val}_{\mathcal{G}}(v) < 0$, by mapping the memoryless optimal strategies of $\mathcal{G}$ into $\mathcal{G}''$, we directly obtain that $\mathsf{Val}_{\mathcal{G}''}(v) \leqslant \mathsf{Val}_{\mathcal{G}}(v) < 0$, since Max has no possibility to go by himself to the target. Reciprocally, if $\mathsf{Val}_{\mathcal{G}''}(v) < 0$, we can project a profile of memoryless optimal strategies over vertices of $\mathcal{G}$, since the target can not be visited in this case (otherwise the optimal play would have mean-payoff 0): 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, $\mathsf{Val}_{\mathcal{G}}(v) \leqslant \mathsf{Val}_{\mathcal{G}''}(v) < 0$. □

[5] This is not needed in the proof, but notice that Min necessarily modifies its strategy here, i.e. owns at least one vertex of the cycle. Otherwise, the value in the MCR game $\mathcal{G}$ would not be $-\infty$.

[6] It suffices to add a vertex of the opponent in-between two vertices of the same player related by a transition in $\mathcal{G}$: in this vertex, the opponent has no choice but to follow the transition chosen by the first player.

3.3 Computing all values

Now that we have discussed the case of vertices with value in $\{-\infty, +\infty\}$, let us present our core contribution on MCR games, which is a pseudo-polynomial time, value iteration algorithm to compute the values of those games. Note that this algorithm is correct even when some vertices have value in $\{-\infty, +\infty\}$, as we will argue later.

In all that follows, we assume that there is exactly one target vertex denoted by $\mathtt{t}$, and the only outgoing edge from $\mathtt{t}$ is a self loop with weight 0: this is reflected by denoting **MCR** the payoff mapping $\{\mathtt{t}\}$-**MCR**. This is without loss of generality since everything that happens after the first occurrence of a target vertex in a play does not matter for the payoff.

Our value iteration algorithm for MCR games is given in Algorithm 1. As it can be seen, this algorithm consists in computing a sequence of vectors X. Initially (line 2), $\mathsf{X}(v) = +\infty$ for all vertices but the target $\mathtt{t}$ where $\mathsf{X}(\mathtt{t}) = 0$. Then, a new value of X is obtained by optimising locally the value of each node, and changing to $-\infty$ the value $\mathsf{X}(v)$ of all vertices v such that the computed value $\mathsf{X}(v)$ has gone below a given threshold $-(|V|-1)W$ (line 14). The following proposition states the correctness of Algorithm 1.

Algorithm 1: Value iteration for MCR games. Gray lines correspond to the computation of optimal strategy for both players (see Sect. 3.4)

Input: MCR game $\langle V, E, \omega, \mathbf{MCR}\rangle$, W greatest weight in absolute value

1 $\mathsf{X}(\mathtt{t}) := 0$
2 **foreach** $v \in V\backslash\{\mathtt{t}\}$ **do** $\mathsf{X}(v) := +\infty$
3 **repeat**
4 $\quad \mathsf{X}_{pre} := \mathsf{X}$
5 $\quad$ **foreach** $v \in V_{\mathsf{Max}}\backslash\{\mathtt{t}\}$ **do**
6 $\quad\quad \mathsf{X}(v) := \max_{v' \in E(v)} \left(\omega\left(v, v'\right) + \mathsf{X}_{pre}\left(v'\right)\right)$
7 $\quad\quad \sigma^{\star}_{\mathsf{Max}}(v) := \operatorname{argmax}_{v' \in E(v)} \left(\omega\left(v, v'\right) + \mathsf{X}_{pre}\left(v'\right)\right)$
8 $\quad$ **foreach** $v \in V_{\mathsf{Min}}\backslash\{\mathtt{t}\}$ **do**
9 $\quad\quad \mathsf{X}(v) := \min_{v' \in E(v)} \left(\omega\left(v, v'\right) + \mathsf{X}_{pre}\left(v'\right)\right)$
10 $\quad\quad$ **if** $\mathsf{X}(v) \neq \mathsf{X}_{pre}(v)$ **then**
11 $\quad\quad\quad \sigma^{\star}_{\mathsf{Min}}(v) := \operatorname{argmin}_{v' \in E(v)} \left(\omega\left(v, v'\right) + \mathsf{X}_{pre}\left(v'\right)\right)$
12 $\quad\quad\quad$ **if** $\mathsf{X}_{pre}(v) = +\infty$ **then** $\sigma^{\dagger}_{\mathsf{Min}}(v) = \sigma^{\star}_{\mathsf{Min}}(v)$
13 $\quad$ **foreach** $v \in V \setminus \{\mathtt{t}\}$ **do**
14 $\quad\quad$ **if** $\mathsf{X}(v) < -(|V|-1)W$ **then** $\mathsf{X}(v) := -\infty$
15 **until** $\mathsf{X} = \mathsf{X}_{pre}$
16 **return** X

Proposition 5 *If an MCR game $\mathcal{G} = \langle V, E, \omega, \mathbf{MCR}\rangle$ is given as input (possibly with values $+\infty$ or $-\infty$), Algorithm 1 outputs $\mathsf{Val}_{\mathcal{G}}$, after at most $(2|V|-1)W|V| + 2|V|$ iterations.*

To establish this proposition, we consider the sequence of values $(x_i)_{i\geqslant 0}$ that vector X takes along the execution of the algorithm. More formally, we can define this sequence thanks to the operator $\mathcal{F}$, which denotes the function $\mathbb{Z}^V_\infty \to \mathbb{Z}^V_\infty$ mapping every vector $x \in \mathbb{Z}^V_\infty$ to $\mathcal{F}(x)$ defined, for all vertices v, by:

$$\mathcal{F}(x)(v) = \begin{cases} 0 & \text{if } v = \mathtt{t} \\ \max_{v' \in E(v)} \left(\omega\left(v, v'\right) + x\left(v'\right)\right) & \text{if } v \in V_{\mathsf{Max}}\backslash\{\mathtt{t}\} \\ \min_{v' \in E(v)} \left(\omega\left(v, v'\right) + x\left(v'\right)\right) & \text{if } v \in V_{\mathsf{Min}}\backslash\{\mathtt{t}\}. \end{cases}$$

Then, for all vertices v we let $x_0(\mathtt{t}) = 0$, and $x_0(v) = +\infty$ for all $v \neq \mathtt{t}$. Moreover, for all $i \geqslant 1$, we let $x_i = \mathcal{F}(x_{i-1})$.

The intuition behind this sequence is that x_i *is the value of the game if we impose that* Min *must reach the target within* i *steps* (and get a payoff of $+\infty$ if he fails to do so). Notice that operator $\mathcal{F}$ is monotonic (i.e. $\mathcal{F}(x) \preccurlyeq \mathcal{F}(y)$ for all $x \preccurlyeq y$), and that $x_1 = \mathcal{F}(x_0) \preccurlyeq x_0$, so that we know that the sequence $(x_i)_{i\geqslant 0}$ is non-increasing:

$$\forall i \geqslant 0 \quad x_{i+1} \preccurlyeq x_i. \tag{1}$$

In order to formalise the intuition that x_i is the value of the game if we impose that Min must reach the target within i steps, we define, for a play $\pi = v_0 v_1 \dots v_i \dots$:

$$\mathbf{MCR}^{\leqslant i}(\pi) = \begin{cases} \mathbf{MCR}(\pi) & \text{if } v_k = \mathtt{t} \text{ for some } k \leqslant i \\ +\infty & \text{otherwise.} \end{cases}$$

We further let $\overline{\mathsf{Val}}^{\leqslant i}(v) = \inf_{\sigma_{\mathsf{Min}}} \sup_{\sigma_{\mathsf{Max}}} \mathbf{MCR}^{\leqslant i}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}}))$ (where σ_{Max} and σ_{Min} are respectively strategies of Max and Min). Observe first that for all $v \in V$, for all $i \geqslant 0$, and for all strategies σ_{Max} and σ_{Min}:

$$\mathbf{MCR}^{\leqslant i}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})) \geqslant \mathbf{MCR}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})).$$

Indeed, if the target vertex $\mathtt{t}$ is reached within i steps, then payoffs are equal. Otherwise, $\mathbf{MCR}^{\leqslant i}(\mathsf{Play}(v, \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})) = +\infty$. Thus, for all $i \geqslant 1$ and $v \in V$:

$$\overline{\mathsf{Val}}^{\leqslant i}(v) \geqslant \overline{\mathsf{Val}}(v) = \mathsf{Val}(v)$$

which can be rewritten as

$$\overline{\mathsf{Val}}^{\leqslant i} \succcurlyeq \overline{\mathsf{Val}} = \mathsf{Val}.$$

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

Lemma 6 *For all* $i \geqslant 1$*, for all* $v \in V$*:*

$$\overline{\mathsf{Val}}^{\leqslant i}(v) = \begin{cases} 0 & \text{if } v = \mathtt{t} \\ \max_{v' \in E(v)} \left(\omega\left(v, v'\right) + \overline{\mathsf{Val}}^{\leqslant i-1}\left(v'\right)\right) & \text{if } v \in V_{\mathsf{Max}} \backslash \{\mathtt{t}\} \\ \min_{v' \in E(v)} \left(\omega\left(v, v'\right) + \overline{\mathsf{Val}}^{\leqslant i-1}\left(v'\right)\right) & \text{if } v \in V_{\mathsf{Min}} \backslash \{\mathtt{t}\}. \end{cases}$$

Proof The lemma can be established by showing that $\overline{\mathsf{Val}}^{\leqslant i}(v)$ is the value in a 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 $A_1, \dots, A_n$ is denoted by $(v, \{A_1, \dots, A_n\})$. Then, for each $v \in V$ and $i \geqslant 0$, we define $A^i(v)$ as follows:

$$A^0(v) = (v)$$
$$\text{for all } i \geqslant 1 : A^i(v) = \left(v, \left\{A^{i-1}\left(v'\right) \mid \left(v, v'\right) \in E\right\}\right).$$

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

$$\lambda(A^0(v)) = \begin{cases} 0 & \text{if } v = \mathtt{t} \\ +\infty & \text{if } v \neq \mathtt{t}. \end{cases}$$

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

$$\lambda\left(A^i(v)\right) = \begin{cases} 0 & \text{if } v = \mathtt{t} \\ \max_{1\leqslant j\leqslant m}\left(\omega(v, v_j) + \lambda\left(A^{i-1}(v_j)\right)\right) & \text{if } v \in V_{\mathsf{Max}}\backslash\{\mathtt{t}\} \\ \min_{1\leqslant j\leqslant m}\left(\omega(v, v_j) + \lambda\left(A^{i-1}(v_j)\right)\right) & \text{if } v \in V_{\mathsf{Min}}\backslash\{\mathtt{t}\}. \end{cases} \quad (2)$$

Clearly, for all $v \in V$, for all $i \geqslant 0$, the branches of $A^i(v)$ correspond to all the possible finite plays $\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})[i]$, i.e. there is a branch for each possible strategy profile $(\sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})$. Thus, $\lambda(A^i(v)) = \overline{\mathsf{Val}}^{\leqslant i}(v)$ for all $i \geqslant 0$, which permits us to conclude from (2). □

We have just shown that for all $i \geqslant 1$, $\overline{\mathsf{Val}}^{\leqslant i} = \mathcal{F}(\overline{\mathsf{Val}}^{\leqslant i-1})$, and since $x_0 = \overline{\mathsf{Val}}^{\leqslant 0}$, we obtain (as expected) $x_i = \overline{\mathsf{Val}}^{\leqslant i}$ for all $i \geqslant 0$. The main question is now to characterise the limit of the sequence $(x_i)_{i\geqslant 0}$, and more precisely, to prove that it is the value Val of the game. Indeed, at this point, it would not be too difficult to show that Val is a fixed point of operator $\mathcal{F}$, but it would be more difficult to show that it is the *greatest* fixed point of $\mathcal{F}$, that is indeed the limit of sequence $(x_i)_{i\geqslant 0}$ (by Kleene's theorem, applicable since $\mathcal{F}$ is Scott-continuous). Instead, we study refined properties of the sequence $(\overline{\mathsf{Val}}^{\leqslant i})_{i\geqslant 0}$, namely its stationarity and the speed of its convergence, and deduce that Val is the greatest fixed point as a corollary (see Corollary 11).

We start by characterising how $\overline{\mathsf{Val}}^{\leqslant i}$ evolves over the first $|V|+1$ steps. The next lemma states that, for each node v, the sequence $\overline{\mathsf{Val}}^{\leqslant 0}(v), \overline{\mathsf{Val}}^{\leqslant 1}(v), \ldots, \overline{\mathsf{Val}}^{\leqslant i}(v), \ldots, \overline{\mathsf{Val}}^{\leqslant |V|}(v)$ is of the form

$$\underbrace{+\infty, +\infty, \ldots, +\infty}_{k \text{ times}}, a_k, a_{k+1}, \ldots, a_{|V|}$$

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

Lemma 7 *Let $v \in V$ be a vertex and let $0 \leqslant k \leqslant |V|$ be such that $v \in \mathsf{Attr}_k(\{\mathtt{t}\})\backslash\mathsf{Attr}_{k-1}(\{\mathtt{t}\})$ (assuming $\mathsf{Attr}_{-1}(\{\mathtt{t}\}) = \emptyset$). Then, for all $0 \leqslant j \leqslant |V|$: (i) $j < k$ implies $\overline{\mathsf{Val}}^{\leqslant j}(v) = +\infty$ and (ii) $j \geqslant k$ implies $\overline{\mathsf{Val}}^{\leqslant j}(v) \leqslant jW$.*

Proof We prove the property for all vertices v, by induction on j.
Base case: $j = 0$. We consider two cases. Either $v = \mathtt{t}$. In this case, $k = 0$, and we must show that $\overline{\mathsf{Val}}^{\leqslant 0}(v) \leqslant 0 \times W = 0$, which is true by definition of $\overline{\mathsf{Val}}^{\leqslant 0}$. Or $v \neq \mathtt{t}$. In this case, $k > 0$, and we must show that $\overline{\mathsf{Val}}^{\leqslant 0}(v) = +\infty$, which is true again by definition of $\overline{\mathsf{Val}}^{\leqslant 0}$.
Inductive case: $j = \ell \geqslant 1$. 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 index k such that $v \in \mathsf{Attr}_k(\{\mathtt{t}\})\backslash\mathsf{Attr}_{k-1}(\{\mathtt{t}\})$. We consider two cases.

1. First, assume $k > \ell$. In this case, we must show that $\overline{\mathsf{Val}}^{\leqslant \ell}(v) = +\infty$. We consider again two cases:
   (a) If $v \in V_{\mathsf{Min}}$, then none of its successors belong to $\mathsf{Attr}_{\ell-1}(\{\mathtt{t}\})$, otherwise, v would be in $\mathsf{Attr}_\ell(\{\mathtt{t}\})$, by definition of the attractor, and we would have $k \leqslant \ell$. Hence, by induction hypothesis, $\overline{\mathsf{Val}}^{\leqslant \ell-1}(v') = +\infty$ for all v' such that $(v, v') \in E$. Thus:

$$\begin{aligned}\overline{\mathsf{Val}}^{\leqslant \ell}(v) &= \min_{(v,v')\in E}\left(\omega\left(v,v'\right)+\overline{\mathsf{Val}}^{\leqslant \ell-1}\left(v'\right)\right) && \text{(Lemma 6)}\\ &= +\infty.\end{aligned}$$

(b) If $v \in V_{\mathsf{Max}}$, then at least one successor of v does not belong to $\mathsf{Attr}_{\ell-1}(\{\mathtt{t}\})$, otherwise, v would be in $\mathsf{Attr}_{\ell}(\{\mathtt{t}\})$, by definition of the attractor, and we would have $k \leqslant \ell$. Hence, by induction hypothesis, there exists v' such that $(v,v') \in E$ and $\overline{\mathsf{Val}}^{\leqslant \ell-1}(v') = +\infty$. Thus:

$$\begin{aligned}\overline{\mathsf{Val}}^{\leqslant \ell}(v) &= \max_{(v,v')\in E}\left(\omega\left(v,v'\right)+\overline{\mathsf{Val}}^{\leqslant \ell-1}\left(v'\right)\right) && \text{(Lemma 6)}\\ &= +\infty.\end{aligned}$$

2. Second, assume $k \leqslant \ell$. In this case, we must show that $\overline{\mathsf{Val}}^{\leqslant \ell}(v) \leqslant \ell W$. As in the previous item, we consider two cases:

(a) In the case where $v \in V_{\mathsf{Min}}$, we let $\overline{v}$ be a vertex such that $\overline{v} \in \mathsf{Attr}_{k-1}(\{\mathtt{t}\})$ and $(v,\overline{v}) \in E$. Such a vertex exists by definition of the attractor. By induction hypothesis, $\overline{\mathsf{Val}}^{\leqslant \ell-1}(\overline{v}) \leqslant \ell W$. Then:

$$\begin{aligned}\overline{\mathsf{Val}}^{\leqslant \ell}(v) &= \min_{(v,v')\in E}\left(\omega\left(v,v'\right)+\overline{\mathsf{Val}}^{\leqslant \ell-1}\left(v'\right)\right) && \text{(Lemma 6)}\\ &\leqslant \omega\left(v,\overline{v}\right)+\overline{\mathsf{Val}}^{\leqslant \ell-1}\left(\overline{v}\right) && ((v,\overline{v})\in E)\\ &\leqslant \omega\left(v,\overline{v}\right)+(\ell-1)W && \text{(Ind. Hyp.)}\\ &\leqslant W+(\ell-1)W = \ell W.\end{aligned}$$

(b) In the case where $v \in V_{\mathsf{Max}}$, we know that all successors v' of v belong to $\mathsf{Attr}_{k-1}(\{\mathtt{t}\})$ by definition of the attractor. By induction hypothesis, for all successors v' of v: $\overline{\mathsf{Val}}^{\leqslant \ell-1}(v') \leqslant \ell W$. Hence:

$$\begin{aligned}\overline{\mathsf{Val}}^{\leqslant \ell}(v) &= \max_{(v,v')\in E}\left(\omega\left(v,v'\right)+\overline{\mathsf{Val}}^{\leqslant \ell-1}\left(v'\right)\right) && \text{(Lemma 6)}\\ &\leqslant \max_{(v,v')\in E}\left(W+(\ell-1)W\right) && \text{(Ind. Hyp.)}\\ &= \ell W.\end{aligned}$$

□

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

Corollary 8 *For all $v \in V$, $\overline{\mathsf{Val}}^{\leqslant |V|}(v) \leqslant |V|W$.*

The next step is to show that the sequence $(x_i)_{i\geqslant 0}$ stabilises after a bounded number of steps, when all values are finite:

Lemma 9 *In an MCR game where all values are finite, the sequence $(\overline{\mathsf{Val}}^{\leqslant i})_{i\geqslant 0}$ stabilises after at most $(2|V|-1)W|V|+|V|$ steps.*

Proof We first show that if Min can secure, from some vertex v, a payoff less than $-(|V|-1)W$, i.e. $\mathsf{Val}(v) < -(|V|-1)W$, then it can secure an arbitrarily small payoff from that vertex, i.e. $\mathsf{Val}(v) = -\infty$, which contradicts our hypothesis that the value is finite. Hence, let

us suppose that there exists a strategy σ_{Min} for Min such that $\mathsf{Val}(v, \sigma_{\mathsf{Min}}) < -(|V|-1)W$. Let $\mathcal{G}'$ be the mean-payoff game studied in Proposition 4. We will show that $\mathsf{Val}_{\mathcal{G}'}(v) < 0$, which permits to conclude that $\mathsf{Val}_{\mathcal{G}}(v) = -\infty$. Let σ_{Max} be a memoryless strategy of Max. By hypothesis, we know that $\mathbf{MCR}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})) < -(|V|-1)W$. This ensures the existence of a cycle with negative cost in the play $\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})$: otherwise, we could iteratively remove every possible non-negative cycle of the finite play before reaching $\mathtt{t}$ (hence reducing the cost of the play) and obtain a play without cycles before reaching $\mathtt{t}$ with a cost less than $-(|V|-1)W$, 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 Min choose its actions like in the cycle. By this way, we can construct another strategy σ'_{Min} for Min, verifying that for every memoryless strategy σ_{Max} of Max (this would not be true for general strategies of Max), we have $\mathbf{MP}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma'_{\mathsf{Min}}))$ being the weight of the negative cycle in which the play finishes. Since for mean-payoff games, memoryless strategies are sufficient for Max, we deduce that $\mathsf{Val}_{\mathcal{G}'}(v) < 0$.

This reasoning permits to prove that at every step i, $\overline{\mathsf{Val}}^{\leqslant i}(v) \geqslant \mathsf{Val}(v) \geqslant -(|V|-1)W+1$ for all vertices v. Recall from Corollary 8 that, after $|V|$ steps in the sequence, all vertices are assigned a value smaller that $|V|W$. Moreover, we know that the sequence is non-increasing [see (1)]. In summary, for all $k \geqslant 0$ and for all vertices v:

$$-(|V|-1)W+1 \leqslant \overline{\mathsf{Val}}^{\leqslant |V|+k}(v) \leqslant |V|W$$

Hence, in the worst case a strictly decreasing sequence will need $(2|V|-1)W|V|$ 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 stabilises in at most $(2|V|-1)W|V|+|V|$ steps. □

Let us thus denote by $\overline{\mathsf{Val}}^{\leqslant}$ the value obtained when the sequence $(\overline{\mathsf{Val}}^{\leqslant i})_{i \geqslant 0}$ stabilises. We are now ready to prove that this value is the actual value of the game:

Lemma 10 *For all MCR games where all values are finite:* $\overline{\mathsf{Val}}^{\leqslant} = \mathsf{Val}$.

Proof We already know that $\overline{\mathsf{Val}}^{\leqslant} \succcurlyeq \mathsf{Val}$. Let us show that $\overline{\mathsf{Val}}^{\leqslant} \preccurlyeq \mathsf{Val}$. Let $v \in V$ be a vertex. Since $\mathsf{Val}(v)$ is a finite integer, there exists a strategy σ_{Min} for Min that realises this value, i.e. $\mathsf{Val}(v) = \sup_{\sigma_{\mathsf{Max}}} \mathbf{MCR}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}}))$. Notice that this holds because the values are integers, inducing that the infimum in the definition of $\overline{\mathsf{Val}}(v) = \mathsf{Val}(v)$ is indeed reached.

Let us build a tree $A_{\sigma_{\mathsf{Min}}}$ unfolding all possible plays from v against σ_{Min}. $A_{\sigma_{\mathsf{Min}}}$ has a root labeled by v. If a tree node is labeled by a vertex v of Min, this tree node has a unique child labeled by $\sigma_{\mathsf{Min}}(v)$. If a tree node is labeled by a vertex v of Max, 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 $\mathtt{t}$ in which case this node is a leaf. $A_{\sigma_{\mathsf{Min}}}$ is necessarily finite. Otherwise, by König's lemma, it has one infinite branch that never reaches $\mathtt{t}$. From that infinite branch, one can extract a strategy σ_{Max} for Max such that $\mathbf{MCR}(\mathsf{Play}(v, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})) = +\infty$, hence $\mathsf{Val}(v) = +\infty$, which contradicts the hypothesis. Assume the tree has depth m. Then, $A_{\sigma_{\mathsf{Min}}}$ is a subtree of the tree A obtained by unfolding all possible plays up to length m (as in the proof of Lemma 6). In this case, it is easy to check that the value labeling the root of $A_{\sigma_{\mathsf{Min}}}$ after applying backward induction is larger than or

equal to the value labeling the root of A after applying backward induction. The latter is $\mathsf{Val}(v)$ while the former is $\overline{\mathsf{Val}}^{\leqslant m}(v)$, by Lemma 6, so that $\mathsf{Val}(v) \geqslant \overline{\mathsf{Val}}^{\leqslant m}(v)$. Since the sequence is non-increasing, we finally obtain $\mathsf{Val}(v) \geqslant \overline{\mathsf{Val}}^{\leqslant}(v)$. □

As a corollary of this lemma, we obtain:

Corollary 11 *In all MCR games where all values are finite,* Val *is the greatest fixed point of* $\mathcal{F}$.

We are finally able to establish the correctness of Algorithm 1.

Proof of Proposition 5 Let us first suppose that the values of all vertices are finite. Then, $x_j = \overline{\mathsf{Val}}^{\leqslant j}$ is the value of X at the beginning of the jth step of the loop, and the condition of line 13 can never be fulfilled. Hence, by Lemma 9, after at most $(2|V| - 1)W|V| + |V|$ iterations, all values are computed correctly (by Lemma 10) in that case.

Suppose now that there are 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 $-\infty$. By the proof of Lemma 9, we know that optimal values of vertices of values different from $-\infty$ are at least $-(|V| - 1)W + 1$ so that, if the value of a vertex reaches an integer below $-(|V| - 1)W$, we are sure that its value is indeed $-\infty$, which proves correct the line 13 of the algorithm. This update may cost at most one step per vertex, which in total adds at most $|V|$ iterations. Moreover, dropping the value to $-\infty$ does not harm the correction for the other vertices (it may only speed up the convergence of their values). This is due to the fact that, if the Kleene sequence $(\mathcal{F}^i(x_0))_{i \geqslant 0}$ is initiated with a vector of values x_0 that is greater or equal to the optimal value vector Val, then the sequence converges at least as fast as before towards the optimal value vector. □

Example 12 We close this discussion on the computation of the values by an example of execution of Algorithm 1. Consider the MCR game in Fig. 2. The successive values for vertices (v_1, v_2) (value of the target v_3 is always 0) computed by the value iteration algorithm are the following: $(+\infty, +\infty)$, $(+\infty, 0)$, $(-1, 0)$, $(-1, -1)$, $(-2, -1)$, $(-2, -2), \ldots, (-W, -W + 1)$, $(-W, -W)$. This requires $2W$ steps to converge (hence a pseudo-polynomial time).

3.4 Computing optimal strategies for both players

As we have seen earlier, Min does not always have optimal memoryless strategies. However, we will see that one can always construct so-called *negative cycle strategies* (NC-strategies), which are memoryless strategies that have a meaningful structure for Min, in the sense that they allow him either: (i) to reach the target by means of a play whose value is lower that the value of the game; or (ii) to decrease arbitrarily the partial sums along the play, when it does not reach the target (in other words, the partial sums tend to $-\infty$ as the play goes on). So, NC-strategies are, in general, not optimal, as they do not guarantee to reach the target (but in this case, they guarantee that Min will play consistently with his objective, by decreasing the value of the play prefixes).

Formally, an NC-strategy is a memoryless strategy σ_{Min} for player Min such that, for all cycles $v_0 \ldots v_{k-1} v_k$ (with $v_k = v_0$) conforming to σ_{Min} (i.e. where each $v_i \in V_{\mathsf{Min}}$ satisfies $\sigma(v_i) = v_{i+1}$),

$$\omega(v_0, v_1) + \cdots + \omega(v_{k-1}, v_k) < 0.$$

We define the *fake value*, $\mathrm{fake}(v, \sigma_{\mathsf{Min}})$, of an NC-strategy σ_{Min} as the supremum of the costs of the *finite* plays that conform with it and start in v:

$$\mathrm{fake}(v, \sigma_{\mathsf{Min}}) = \sup\{\mathbf{MCR}(v_1 \dots v_k) \mid v_1 = v, v_k = \mathtt{t}, \forall v_i \in V_{\mathsf{Min}} : \sigma_{\mathsf{Min}}(v_i) = v_{i+1}\}.$$

Notice that the fake value is not necessarily equal to the value of σ_{Min}, since, in the definition of the fake value, we only consider plays that *do reach* the target, and ignore those that don't (in the computation of the actual value of σ_{Min}, these plays would yield $+\infty$). However, a strategy's fake value is always smaller than or equal to its value. We say that an NC-strategy is *fake-optimal* if its fake value is smaller than or equal to the optimal value of the game from all vertices. In particular, if the optimal value of a vertex v is $-\infty$, the set $\{\mathbf{MCR}(v_1 \dots v_k) \mid v_1 = v, v_k = \mathtt{t}, \forall v_i \in V_{\mathsf{Min}}\ \sigma_{\mathsf{Min}}(v_i) = v_{i+1}\}$ is empty for all fake-optimal strategies σ_{Min}, hence the supremum of this set is indeed $-\infty$.

Example 13 On the game of Fig. 2, the memoryless strategy σ_{Min} mapping v_2 to v_1 is an NC-strategy as the only two possible cycles $v_1v_2v_1$ and $v_2v_1v_2$ both have weight -1. The value of σ_{Min} from v_2 is $+\infty$ as the play $v_2v_1v_2v_1v_2\dots$ agrees with it and does not reach the target, but the fake value of σ_{Min} is $-W$, since the play $v_2v_1v_3$ is the finite play that agrees with σ_{Min} with the biggest cost possible. Since we know that the actual optimal value is $-W$, the strategy σ_{Min} is fake-optimal.

The following proposition reveals the interest of NC-strategies, by explaining how one can, in some cases, construct an optimal finite-memory strategy from a fake-optimal NC-strategy.

Proposition 14 *If* Min *has a strategy* $\sigma^{\dagger}_{\mathsf{Min}}$ *to reach a target vertex (from every possible initial vertex), and has a fake-optimal NC-strategy* $\sigma^{\star}_{\mathsf{Min}}$*, then for all* $n \in \mathbb{Z}$ *one can construct a finite-memory strategy* $\sigma^{n}_{\mathsf{Min}}$ *such that for all vertices* v*, it holds that* $\mathsf{Val}(v, \sigma^{n}_{\mathsf{Min}}) \leqslant \max(n, \mathsf{Val}(v))$.

Remark 15 In particular, if the value of all vertices is finite, then one can construct an optimal finite-memory strategy. If the value of a vertex is $-\infty$, this proposition also says that there is an infinite family a strategies that allows one to secure a value which is arbitrarily low (remember that, by definition, $-\infty$ can not be the value that corresponds to a single strategy).

Proof First let us show that for all partial plays $\pi = v_1 \dots v_\ell$ of size at least $k|V|+1$ (for some k) that conforms with $\sigma^{\star}_{\mathsf{Min}}$, $\mathbf{TP}(\pi) \leqslant W(|V|-1) - k$. We establish this proof by induction on k. For the base case, we consider the case where $\ell \leqslant |V|$. Then, the play visits at most $|V|-1$ edges, and thus its total cost is at most $W(|V|-1)$. Now, for the induction, we assume that $\ell \geqslant k|V|+1$ for some $k \geqslant 1$. Then, let i and j be two indices such that $i < j$, $v_i = v_j$ and $j \leqslant i + |V|$ (those indices necessarily exist since $\ell \geqslant |V|+1$). Since $\sigma^{\star}_{\mathsf{Min}}$ is an NC-strategy, the total cost of $v_i \dots v_j$ is at most -1. As $\pi' = v_1 \dots v_i v_{j+1} \dots v_\ell$ is also a play that conforms with $\sigma^{\star}_{\mathsf{Min}}$ with size greater than $(k-1)|V|+1$, we have (by induction hypothesis) $\mathbf{TP}(\pi') \leqslant W(|V|-1) - k + 1$, thus $\mathbf{TP}(\pi) = \mathbf{TP}(v_i \dots v_j) + \mathbf{TP}(\pi') \leqslant W(|V|-1) - k$.

In the following, let $\sigma^{\dagger}_{\mathsf{Min}}$ be a memoryless strategy ensuring to reach the target (obtained by the attractor technique for instance), and let $k = \max(2W(|V|-1) - n, 0)$. The strategy $\sigma^{n}_{\mathsf{Min}}$ consists in playing $\sigma^{\star}_{\mathsf{Min}}$, until switching to $\sigma^{\dagger}_{\mathsf{Min}}$ when the length of the play is greater than $k|V|+1$: formally $\sigma^{n}_{\mathsf{Min}}(\pi) = \sigma^{\star}_{\mathsf{Min}}(\pi)$ if $|\pi| < k|V|+1$ and $\sigma^{n}_{\mathsf{Min}}(\pi) = \sigma^{\dagger}_{\mathsf{Min}}(\pi)$ otherwise. It is clear that this strategy can be implemented by a finite deterministic Moore machine, storing the size of the current play until it is greater than $k|V|+1$.

Let us now check that $\mathsf{Val}(v, \sigma^{n}_{\mathsf{Min}}) \leqslant \max(n, \mathsf{Val}(v))$ for all vertices. Let π be a play conforming to $\sigma^{n}_{\mathsf{Min}}$ starting in a vertex v and reaching $\mathtt{t}$. If $|\pi| \leqslant k|V|+1$ then π is a play that reaches the target conforming to $\sigma^{\star}_{\mathsf{Min}}$ and therefore

$$\begin{aligned}\mathbf{MCR}(\pi) &\leqslant \mathrm{fake}\left(v, \sigma^{\star}_{\mathsf{Min}}\right) && \pi \text{ is a finite play reaching } \mathtt{t}\\ &\leqslant \mathsf{Val}(v) && \sigma^{\star}_{\mathsf{Min}} \text{ is fake-optimal}\\ &\leqslant \max(n, \mathsf{Val}(v)).\end{aligned}$$

If $|\pi| > k|V| + 1$, then let π_1 be its prefix of size $k|V| + 1$, and π_2 be the rest of the play ($\pi = \pi_1 \cdot \pi_2$). As π_2 is a play that conforms with $\sigma^{\dagger}_{\mathsf{Min}}$, a memoryless strategy ensuring to reach the target, we know that it reaches the target in at most $|V|$ steps. Therefore, $\mathbf{TP}(\pi_2) \leqslant W(|V| - 1)$. As π_1 is a play that conforms with $\sigma^{\star}_{\mathsf{Min}}$ of size $k|V| + 1$, from the reasoning above, $\mathbf{TP}(\pi_1) \leqslant W(|V|-1)-k \leqslant n-W(|V|-1)$, therefore $\mathbf{TP}(\pi) = \mathbf{TP}(\pi_1)+\mathbf{TP}(\pi_2) \leqslant n \leqslant \max(n, \mathsf{Val}(v))$. □

In practice, rather than using a Moore machine, we can simulate the strategy $\sigma^{n}_{\mathsf{Min}}$ (of the proof above) quite easily: one just has to handle two memoryless strategies and a counter keeping track of the length of the current play. Since $\sigma^{\dagger}_{\mathsf{Min}}$ can easily be obtained by the classical attractor algorithm, we turn our attention to the construction of a fake-optimal NC-strategy $\sigma^{\star}_{\mathsf{Min}}$. Without loss of generality, we suppose that no vertex has optimal value $+\infty$, since for these vertices, all strategies are equivalent.

For vertices of value $-\infty$, we can obtain $\sigma^{\star}_{\mathsf{Min}}$ as an optimal strategy for Min in the mean-payoff game of the first item in Proposition 4. Since the mean-payoff value is negative, this strategy guarantees that it does not reach target, thus generating a fake value $-\infty$, equal to the optimal value of the vertex. Moreover, since it is a memoryless strategy, we know that, as soon as Max plays a memoryless strategy that necessarily reaches a cycle, this cycle must have a negative weight (at most the optimal value of the initial vertex): this strategy is thus a fake-optimal NC-strategy.

From now on, we thus concentrate our study on the vertices of finite value, thus considering that no vertices have value $+\infty$ or $-\infty$ in the MCR game. Let X^i denote the value of variable X after i iterations of the loop of Algorithm 1, and let $X^0(v) = +\infty$ for all $v \in V$. We have seen that the sequence $X^0 \succcurlyeq X^1 \succcurlyeq X^2 \succcurlyeq \dots$ is stationary at some point, equal to Val. Let us now define $\sigma^{\star}_{\mathsf{Min}}(v)$ for all vertices $v \in V_{\mathsf{Min}}\backslash\{\mathtt{t}\}$. We let $i_v > 0$ be the smallest index such that $X^{i_v}(v) = \mathsf{Val}(v)$. Fix a vertex v' such that $X^{i_v}(v) = \omega(v, v') + X^{i_v-1}(v')$ (such a v' exists by definition) and define $\sigma^{\star}_{\mathsf{Min}}(v) = v'$. Notice that it is exactly what is achieved in line 11 of Algorithm 1. Note also that strategy $\sigma^{\dagger}_{\mathsf{Min}}$ is correctly computed in line 12 of Algorithm 1.

Let us prove that this construction indeed yields a fake-optimal NC strategy $\sigma^{\star}_{\mathsf{Min}}$. We first prove the following lemma, that states that the vertex $\sigma^{\star}_{\mathsf{Min}}(v)$ has already reached its final value at step $i_v - 1$ of the algorithm, for all vertices v.

Lemma 16 *For all vertices $v \in V_{\mathsf{Min}}\backslash\{\mathtt{t}\}$, $X^{i_v-1}(\sigma^{\star}_{\mathsf{Min}}(v)) = \mathsf{Val}(\sigma^{\star}_{\mathsf{Min}}(v))$.*

Proof Let $v' = \sigma^{\star}_{\mathsf{Min}}(v)$. Since $(X^i(v'))_{i\geqslant 0}$ is non-increasing and converges towards the value of v', we know that $X^{i_v-1}(v') \geqslant \mathsf{Val}(v')$. By contradiction assume that $X^{i_v-1}(v') > \mathsf{Val}(v')$. Note that there exists $j > i_v$ such that $X^{j-1}(v') = \mathsf{Val}(v')$. Therefore:

$$\begin{aligned}\mathsf{Val}(v) &\leqslant X^{j}(v) && \left(\text{since } \left(X^k(v)\right)_{k\in\mathbb{N}} \text{ is non-increasing}\right)\\ &\leqslant \omega\left(v, v'\right) + X^{j-1}\left(v'\right) && \left(\text{by definition of } X^j\right)\\ &= \omega\left(v, v'\right) + \mathsf{Val}\left(v'\right) && (\text{by choice of } j)\end{aligned}$$

$$< \omega\left(v, v'\right) + X^{i_v - 1}\left(v'\right) \qquad \text{(by the contradiction hypothesis)}$$
$$= X^{i_v}(v) = \mathsf{Val}(v) \qquad \left(\text{by definition of } X^{i_v}\right)$$

which raises a contradiction. □

We can now prove that our definition of $\sigma^\star_{\mathsf{Min}}$ has the announced properties:

Proposition 17 *$\sigma^\star_{\mathsf{Min}}$ is an NC-strategy, and* $\mathrm{fake}(v, \sigma^\star_{\mathsf{Min}}) \leqslant \mathsf{Val}(v)$ *for all vertices v.*

Proof Equivalently, we want to prove that for all vertices v, and for all plays $\pi = v_1 v_2 \dots$ starting in v and conforming to $\sigma^\star_{\mathsf{Min}}$,

(1) if there exists $i < j$ such that $v_i = v_j$, then $\mathbf{TP}(v_i \dots v_j) < 0$,
(2) if π reaches $\mathtt{t}$ then $\mathbf{MCR}(\pi) \leqslant \mathsf{Val}(v)$.

For (1), consider a cycle $v_i \dots v_j$ with $v_j = v_i$. Notice that at least one vertex of this cycle belongs to Min, since, otherwise, Max would have a strategy to obtain a value $+\infty$ for vertex v_i, which contradicts the hypothesis that every vertex has value different from $+\infty$. Hence, for the sake of the explanation, we suppose that $v_i \in V_{\mathsf{Min}}$, and that the index i_{v_i} where vertex v_i stabilises in the sequence $(X^k(v_i))_{k \geqslant 0}$ is maximal among all possible vertices of $\{v_i, \dots, v_j\} \cap V_{\mathsf{Min}}$. The following extends easily to the general case.

We prove by induction over $i < \ell \leqslant j$ that

$$X^{i_{v_i}}(v_i) \geqslant \mathbf{TP}(v_i \dots v_\ell) + X^{i_{v_i} - 1}(v_\ell). \tag{3}$$

The base case $\ell = i + 1$ comes from the fact that, since $v_i \in V_{\mathsf{Min}}$, we have $\sigma^\star_{\mathsf{Min}}(v_i) = v_{i+1}$. Therefore, $X^{i_{v_i}}(v_i) = \omega(v_i, v_{i+1}) + X^{i_{v_i} - 1}(v_{i+1})$, and by definition $\mathbf{TP}(v_i v_{i+1}) = \omega(v_i, v_{i+1})$.

For the inductive case, let us consider $i < \ell < j$ such that (3) holds and let us prove it for $\ell + 1$. If $v_\ell \in V_{\mathsf{Max}}$, by definition of $X^{i_{v_i}}$, we have

$$X^{i_{v_i}}(v_\ell) = \max_{(v_\ell, v') \in E} \omega\left(v_\ell, v'\right) + X^{i_{v_i} - 1}\left(v'\right) \geqslant \omega(v_\ell, v_{\ell+1}) + X^{i_{v_i} - 1}(v_{\ell+1}).$$

If $v_\ell \in V_{\mathsf{Min}}$, by maximality of i_{v_i}, we have

$$X^{i_{v_i}}(v_\ell) = X^{i_{v_\ell}}(v_\ell) = \omega(v_\ell, v_{\ell+1}) + X^{i_{v_\ell} - 1}(v_{\ell+1})$$
$$\geqslant \omega(v_\ell, v_{\ell+1}) + X^{i_{v_i} - 1}(v_{\ell+1})$$

using that the sequence $X^0, X^1, X^2, \dots$ is non-increasing. In all cases, we have

$$X^{i_{v_i}}(v_\ell) \geqslant \omega(v_\ell, v_{\ell+1}) + X^{i_{v_i} - 1}(v_{\ell+1}).$$

Using again that $X^0, X^1, X^2, \dots$ is non-increasing, we obtain

$$X^{i_{v_i} - 1}(v_\ell) \geqslant X^{i_{v_i}}(v_\ell) \geqslant \omega(v_\ell, v_{\ell+1}) + X^{i_{v_i} - 1}(v_{\ell+1}).$$

Injecting this into the induction hypothesis, we have

$$X^{i_{v_i}}(v_i) \geqslant \mathbf{TP}(v_i \dots v_\ell) + \omega(v_\ell, v_{\ell+1}) + X^{i_{v_i} - 1}(v_{\ell+1})$$
$$= \mathbf{TP}(v_i \dots v_{\ell+1}) + X^{i_{v_i} - 1}(v_{\ell+1})$$

which concludes the proof by induction.

In particular, for $\ell = j$, as $v_i = v_j$ we obtain that

$$X^{i_{v_i}}(v_i) \geqslant X^{i_{v_i} - 1}(v_i) + \mathbf{TP}(v_i \dots v_j).$$

and as, by definition of i_{v_i}, we have $X^{i_{v_i}}(v_i) < X^{i_{v_i}-1}(v_i)$, we necessarily have $\mathbf{TP}(v_i \dots v_j) < 0$.

We now prove (2). We decompose π as $\pi = v_1 \dots v_k \mathtt{t}^\omega$ with for all i, $v_i \neq \mathtt{t}$. We prove by decreasing induction on i that $\mathbf{MCR}(v_i \dots v_k \mathtt{t}^\omega) \leqslant \mathsf{Val}(v_i)$. If $i = k+1$, $\mathbf{MCR}(\mathtt{t}^\omega) = 0 = \mathsf{Val}(\mathtt{t})$. If $i \leqslant k$, by induction we have $\mathbf{MCR}(v_{i+1} \dots v_k \mathtt{t}^\omega) \leqslant \mathsf{Val}(v_{i+1})$. Thus $\mathbf{MCR}(v_i \dots v_k \mathtt{t}^\omega) \leqslant \omega(v_i, v_{i+1}) + \mathsf{Val}(v_{i+1})$. If $v_i \in V_{\mathsf{Min}}$, then $v_{i+1} = \sigma^\star_{\mathsf{Min}}(v_i)$, and by Lemma 16,

$$\mathsf{Val}(v_{i+1}) = X^{i_{v_i}-1}(v_{i+1}) = X^{i_{v_i}}(v_i) - \omega(v_i, v_{i+1})$$

by definition of $\sigma^\star_{\mathsf{Min}}$. Since $X^{i_{v_i}}(v_i) = \mathsf{Val}(v_i)$, this can be rewritten as $\mathsf{Val}(v_i) = \omega(v_i, v_{i+1}) + \mathsf{Val}(v_{i+1}) \geqslant \mathbf{MCR}(v_i \dots v_k \mathtt{t}^\omega)$, which proves the inequality for i. If $v_i \in V_{\mathsf{Max}}$, then

$$\mathbf{MCR}\left(v_i \dots v_k \mathtt{t}^\omega\right) \leqslant \omega(v_i, v_{i+1}) + \mathsf{Val}(v_{i+1}) \leqslant \max_{(v_i, v') \in E} \omega\left(v_i, v'\right) + \mathsf{Val}\left(v'\right) = \mathsf{Val}(v_i)$$

the last equality coming from Corollary 11. □

As a corollary, we obtain the existence of *finite memory strategies* (obtained from $\sigma^\star_{\mathsf{Min}}$ and $\sigma^\dagger_{\mathsf{Min}}$ as described above) when all values are finite:

Corollary 18 *When the values of all vertices of an MCR game are finite, one can construct an optimal finite-memory strategy for player* Min.

Strategies of Max Let us now show that Max always has a *memoryless optimal strategy*. This asymmetry stems directly from the asymmetric definition of the game—while Min has the double objective of reaching $\mathtt{t}$ and minimising its cost, Max aims at avoiding $\mathtt{t}$, and if not possible, maximising the cost.

Proposition 19 *In all MCR games,* Max *has a memoryless optimal strategy.*

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

We now explain how to define a memoryless optimal strategy $\sigma^\star_{\mathsf{Max}}$ for Max in case of a graph containing only finite values. For every finite play π ending in a vertex $v \in V_{\mathsf{Max}}$ of Max, we let

$$\sigma^\star_{\mathsf{Max}}(\pi) = \operatorname*{argmax}_{v' \in E(v)} \left(\omega\left(v, v'\right) + \mathsf{Val}\left(v'\right)\right).$$

This is clearly a memoryless strategy. Let us prove that it is optimal for Max, that is, for every vertex $v \in V$, and every strategy σ_{Min} of Min

$$\mathbf{MCR}\left(\mathsf{Play}\left(v, \sigma^\star_{\mathsf{Max}}, \sigma_{\mathsf{Min}}\right)\right) \geqslant \mathsf{Val}(v).$$

In case $\mathsf{Play}(v, \sigma^\star_{\mathsf{Max}}, \sigma_{\mathsf{Min}})$ does not reach the target set of vertices, the inequality holds trivially. Otherwise, we let $\mathsf{Play}(v, \sigma^\star_{\mathsf{Max}}, \sigma_{\mathsf{Min}}) = v_0 v_1 \dots v_\ell \dots$ with ℓ the least position such that $v_\ell = \mathtt{t}$. If $\ell = 0$, i.e. $v = v_0 = \mathtt{t}$, we have

$$\mathbf{MCR}\left(\mathsf{Play}\left(v, \sigma^\star_{\mathsf{Max}}, \sigma_{\mathsf{Min}}\right)\right) = 0 = \mathsf{Val}(v).$$

Otherwise, let us prove by induction on $0 \leqslant i \leqslant \ell$ that

$$\mathbf{MCR}(v_{\ell-i} \dots v_\ell) \geqslant \mathsf{Val}(v_{\ell-i}).$$

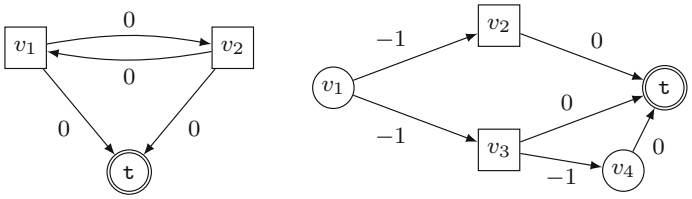

Fig. 3 Importance of the **if** condition of Algorithm 1

This will permit to conclude since

$$\mathbf{MCR}\left(\mathsf{Play}\left(v, \sigma^{\star}_{\mathsf{Max}}, \sigma_{\mathsf{Min}}\right)\right) = \mathbf{MCR}(v_0 v_1 \dots v_\ell) \geqslant \mathsf{Val}(v_0) = \mathsf{Val}(v).$$

The base case $i = 0$ corresponds to the previous case where the starting vertex is $\mathtt{t}$. Supposing that the property holds for index i, let us prove it for $i + 1$. We have

$$\mathbf{MCR}(v_{\ell-i-1} \dots v_\ell) = \omega(v_{\ell-i-1}, v_{\ell-i}) + \mathbf{MCR}(v_{\ell-i} \dots v_\ell).$$

By induction hypothesis, we have

$$\mathbf{MCR}\left(v_{\ell-i-1} \dots v_\ell\right) \geqslant \omega\left(v_{\ell-i-1}, v_{\ell-i}\right) + \mathsf{Val}(v_{\ell-i}). \tag{4}$$

We now consider two cases:

– If $v_{\ell-i-1} \in V_{\mathsf{Max}} \backslash \{\mathtt{t}\}$, then $v_{\ell-i} = \sigma^{\star}_{\mathsf{Max}}(v_0 v_1 \dots v_{\ell-i-1})$, so that by definition of $\sigma^{\star}_{\mathsf{Max}}$:

$$\omega(v_{\ell-i-1}, v_{\ell-i}) + \mathsf{Val}(v_{\ell-i}) = \max_{v' \in V \mid (v_{\ell-i-1}, v') \in E} \left(\omega\left(v_{\ell-i-1}, v'\right) + \mathsf{Val}(v')\right).$$

Using Corollary 11 and (4), we obtain

$$\mathbf{MCR}(v_{\ell-i-1} \dots v_\ell) \geqslant \mathsf{Val}(v_{\ell-i-1}).$$

– If $v_{\ell-i-1} \in V_{\mathsf{Min}} \backslash \{\mathtt{t}\}$, then

$$\omega(v_{\ell-i-1}, v_{\ell-i}) + \mathsf{Val}(v_{\ell-i}) \geqslant \min_{v' \in V \mid (v_{\ell-i-1}, v') \in E} \left(\omega\left(v_{\ell-i-1}, v'\right) + \mathsf{Val}\left(v'\right)\right).$$

Once again using Corollary 11 and (4), we obtain

$$\mathbf{MCR}(v_{\ell-i-1} \dots v_\ell) \geqslant \mathsf{Val}(v_{\ell-i-1}).$$

This concludes the proof. □

Notice that strategy $\sigma^{\star}_{\mathsf{Max}}$ can be computed directly, along the execution of the value iteration algorithm. This corresponds to line 7 of Algorithm 1.

Notice further that the presence of the **if** condition in line 11, and the absence of a similar condition at line 7, are crucial. Indeed, removing the **if** from line 11 would amount to computing $\sigma^{\star}_{\mathsf{Min}}$ from the vector of values obtained at the end of the value iteration, when the vector X has stabilised. Let us show that, in this case, the algorithm might fail to compute a fake-optimal NC-strategy, by considering the MCR game in the left part of Fig. 3. Clearly, the values of both v_1 and v_2 are 0. However, if we extract $\sigma^{\star}_{\mathsf{Min}}$ from X_{pre} at that point of the execution of the algorithm [i.e. we let $\sigma^{\star}_{\mathsf{Min}}(v) = \mathrm{argmin}_{v' \in E(v)}(\omega(v, v') + \mathsf{X}_{pre}(v'))$], then we could end up with $\sigma^{\star}_{\mathsf{Min}}(v_1) = v_2$ and $\sigma^{\star}_{\mathsf{Min}}(v_2) = v_1$. In this case, $\sigma^{\star}_{\mathsf{Min}}$ is no longer an NC-strategy because the cycle $v_1 v_2 v_1$ does not have negative weight. Similarly, let us consider the MCR game in the right part of Fig. 3 to explain why line 7 is not under the

range of an **if** $(\mathsf{X}(v) \neq \mathsf{X}_{pre}(v))$ condition. After two iterations, X reaches the optimal values $(-1, 0, -1, 0)$ but a Max strategy $\sigma^\star_{\mathsf{Max}}$ such that $\sigma^\star_{\mathsf{Max}}(v_1) = v_3$ can still be chosen since $\mathsf{X}_{pre}(v_3) = 0$ at that point. However, on the next iteration, $\mathsf{X}(v_3) = \mathsf{X}_{pre}(v_3) = -1$ (indeed, X has now stabilised on all nodes), and it is crucial that $\sigma^\star_{\mathsf{Max}}(v_1) = v_2$ gets computed, otherwise the strategy would not be optimal for Max.

4 An efficient algorithm to solve total-payoff games

We now turn our attention back to total-payoff games (without reachability objective), and discuss our main contribution. Building on the results of the previous section, we introduce the *first* (as far as we know) *pseudo-polynomial time algorithm* for solving those games in the presence of arbitrary weights, thanks to a reduction from total-payoff games to MCR games. The MCR game produced by the reduction has size pseudo-polynomial in the size of the original total-payoff game. Then, we show how to compute the values of the total-payoff game without building the entire MCR game, and explain how to deduce memoryless optimal strategies from the computation of our algorithm.

4.1 Reduction to MCR games

We provide a transformation from a total-payoff game $\mathcal{G} = \langle V, E, \omega, \mathbf{TP}\rangle$ to an MCR game $\mathcal{G}^K$ (where K is a parameter in $\mathbb{N}$) such that the values of $\mathcal{G}$ can be extracted from the values in $\mathcal{G}^K$ (as formalised below). Intuitively, $\mathcal{G}^K$ simulates the game where players play in $\mathcal{G}$; Min may propose to stop playing and reach a fresh vertex $\mathtt{t}$ acting as the target; Max can then accept, in which case we reach the target, or refuse at most K times, in which case the game continues. Structurally, $\mathcal{G}^K$ consists of a sequence of copies of $\mathcal{G}$ along with some new states that we now describe formally. We let $\mathtt{t}$ be a fresh vertex, and, for all $n \geqslant 1$, we define the MCR game $\mathcal{G}^n = \langle V^n, E^n, \omega^n, \{\mathtt{t}\}\text{-}\mathbf{MCR}\rangle$ where V^n_{Max} (respectively, V^n_{Min}) consists of n copies (v, j), with $1 \leqslant j \leqslant n$, of each vertex $v \in V_{\mathsf{Max}}$ (respectively, $v \in V_{\mathsf{Min}}$) and some *exterior vertices* $(\mathtt{ex}, v, j)$ for all $v \in V$ and $1 \leqslant j \leqslant n$ [respectively, *interior vertices* $(\mathtt{in}, v, j)$ for all $v \in V$ and $1 \leqslant j \leqslant n$]. Moreover, V^n_{Max} contains the fresh target vertex $\mathtt{t}$. Edges are given by

$$
\begin{aligned}
E^n = {} & \{(\mathtt{t}, \mathtt{t})\} \uplus \left\{\left((v, j), \left(\mathtt{in}, v', j\right)\right) \mid \left(v, v'\right) \in E, 1 \leqslant j \leqslant n\right\} \\
& \uplus \{((\mathtt{in}, v, j), (v, j)) \mid v \in V, 1 \leqslant j \leqslant n\} \uplus \{((\mathtt{ex}, v, j), \mathtt{t}) \mid v \in V, 1 \leqslant j \leqslant n\} \\
& \uplus \{((\mathtt{in}, v, j), (\mathtt{ex}, v, j)) \mid v \in V, 1 \leqslant j \leqslant n\} \\
& \uplus \{((\mathtt{ex}, v, j), (v, j-1)) \mid v \in V, 1 < j \leqslant n\}\,.
\end{aligned}
$$

All edge weights are zero, except edges $((v, j), (\mathtt{in}, v', j))$ that have weight $\omega(v, v')$.

Example 20 For example, considering the weighted graph of Fig. 2, the corresponding reachability total-payoff game $\mathcal{G}^3$ is depicted in Fig. 4 (where weights 0 have been removed).

The next proposition formalises the relationship between the two games, and is proved in the rest of this subsection.

Proposition 21 *Let* $K = |V|(2(|V| - 1)W + 1)$. *For all* $v \in V$ *and* $k \geqslant K$,

- $\mathsf{Val}_{\mathcal{G}}(v) \neq +\infty$ *if and only if* $\mathsf{Val}_{\mathcal{G}}(v) = \mathsf{Val}_{\mathcal{G}^k}((v, k))$;
- $\mathsf{Val}_{\mathcal{G}}(v) = +\infty$ *if and only if* $\mathsf{Val}_{\mathcal{G}^k}((v, k)) \geqslant (|V| - 1)W + 1$.

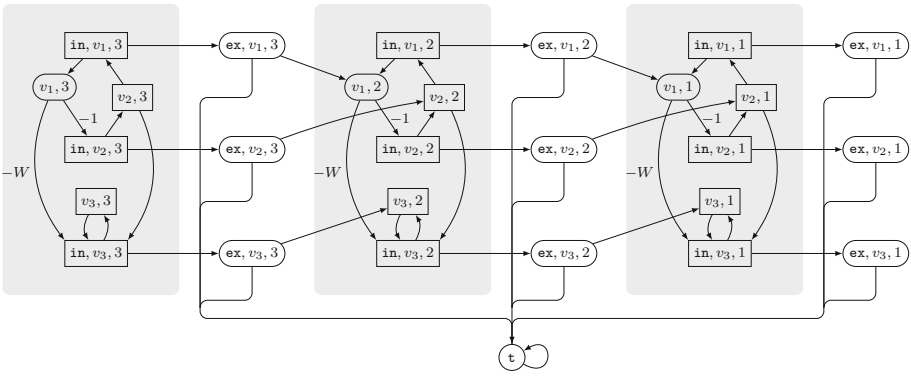

Fig. 4 MCR game $\mathcal{G}^3$ associated with the total-payoff game of Fig. 2

The bound K will be found by using the fact (informally described in the previous section) that if not infinite, the value of a MCR game belongs in $[-(|V|-1)\times W+1, |V|\times W]$, and that after enough visits of the same vertex, an adequate loop ensures that $\mathcal{G}^k$ verifies the above properties.

To prove Proposition 21, we must relate paths in games $\mathcal{G}$ and $\mathcal{G}^n$: with each finite path in $\mathcal{G}^n$, we associate a finite path in $\mathcal{G}$, obtained by looking at the sequence of vertices of V appearing inside the vertices of the finite play. Formally, the *projection* of a finite path π is the sequence $\mathsf{proj}(\pi)$ of vertices of $\mathcal{G}$ inductively defined by $\mathsf{proj}(\varepsilon)=\varepsilon$ and for all finite path π, $v\in V$ and $1\leqslant j\leqslant n$:

$$\mathsf{proj}((\mathtt{in},v,j)\pi)=\mathsf{proj}(\pi),\qquad \mathsf{proj}((\mathtt{ex},v,j)\mathtt{t}\pi)=v,\qquad \mathsf{proj}((\mathtt{ex},v,j))=\varepsilon,$$
$$\mathsf{proj}((\mathtt{ex},v,j+1)(v,j)\pi)=\mathsf{proj}((v,j)\pi)=v\,\mathsf{proj}(\pi).$$

In particular, notice that in the case of a play with prefix $(\mathtt{ex},v,j)\mathtt{t}$, the rest of the play is entirely composed of target vertices $\mathtt{t}$, since $\mathtt{t}$ is a sink state. For instance, the projection of the finite play

$$(v_1,3)(\mathtt{in},v_2,3)(\mathtt{ex},v_2,3)(v_2,2)(\mathtt{in},v_3,2)(v_3,2)(\mathtt{in},v_3,2)(\mathtt{ex},v_3,2)\mathtt{t}$$

of the game $\mathcal{G}^3$ of Fig. 4 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 22 *The projection mapping satisfies the following properties.*

1. *If π is a finite play in $\mathcal{G}^n$ then $\mathsf{proj}(\pi)$ is a finite play in $\mathcal{G}$.*
2. *If π is a play in $\mathcal{G}^n$ that does not reach the target, then $\mathsf{proj}(\pi)$ is a play in $\mathcal{G}$.*
3. *For all finite play π, $\mathbf{TP}(\pi)=\mathbf{TP}(\mathsf{proj}(\pi))$.*

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

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

Then notice that if $(w,w')\in E^n$ with $w,w'\neq\mathtt{t}$, then either $f(w)=f(w')$ or $(f(w),f(w'))\in E$. We now prove 1 and 3 inductively on the size of the finite play $\pi=w_1\dots w_k$ of $\mathcal{G}^n$, along with the fact that

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

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

- If $\pi = (\mathtt{in}, v, j)\pi'$, then $\mathsf{proj}(\pi) = \mathsf{proj}(\pi')$. Hence, 1 holds by induction hypothesis. If $\mathsf{proj}(\pi)$ is non-empty, so is $\mathsf{proj}(\pi')$. Moreover, the first vertex of π' is either (v, j) or $(\mathtt{ex}, v, j)$, so that we have 4 by induction hypothesis. Finally, the previous remark shows that the first edge of π has necessarily weight 0, so that, $\mathbf{TP}(\pi) = \mathbf{TP}(\pi')$, and 3 also holds by induction hypothesis.
- If $\pi = (v, j)\pi'$, then $\mathsf{proj}(\pi) = v\,\mathsf{proj}(\pi')$ so that 4 holds directly. Moreover, π' is a non-empty finite play so that $\pi' = (\mathtt{in}, v', j)\pi''$ with $(v, v') \in E$, and $\mathsf{proj}(\pi') = \mathsf{proj}(\pi'')$. By induction, $\mathsf{proj}(\pi')$ is a finite play in $\mathcal{G}$, and it starts with v' (by 4). Since $(v, v') \in E$, this shows that $\mathsf{proj}(\pi)$ is a finite play. Moreover, $\mathbf{TP}(\pi) = \omega^n((v, j), (\mathtt{in}, v', j)) + \mathbf{TP}(\pi') = \omega(v, v') + \mathbf{TP}(\pi')$. By induction hypothesis, we have $\mathbf{TP}(\pi') = \mathbf{MCR}(\mathsf{proj}(\pi'))$. Moreover, $\mathbf{MCR}(\mathsf{proj}(\pi)) = \omega(v, v') + \mathbf{MCR}(\mathsf{proj}(\pi'))$ which concludes the proof of 3.
- If $\pi = (\mathtt{ex}, v, j)(v, j-1)\pi'$ then $\mathsf{proj}(\pi) = v\,\mathsf{proj}(\pi') = \mathsf{proj}((v, j-1)\pi')$: this allows us to conclude directly by using the previous case.
- Otherwise, $\pi = (\mathtt{ex}, v, j)\mathtt{t}\pi'$, and then $\mathsf{proj}(\pi) = v$ is a finite play with total-payoff 0, like π, and 4 holds trivially. □

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

Lemma 23 *Let $v \in V$, and σ_{Min} (respectively, σ_{Max}) be a memoryless strategy for* Min (*respectively,* Max) *in the total-payoff game $\mathcal{G}$, such that* $\mathsf{Val}(v, \sigma_{\mathsf{Min}}) \neq +\infty$ (*respectively,* $\mathsf{Val}(v, \sigma_{\mathsf{Max}}) \neq -\infty$). *Then for all finite play π conforming to σ_{Min} (respectively, to σ_{Max}),* $\mathbf{TP}(\pi) \leqslant (|V| - 1)W$ (*respectively,* $\mathbf{TP}(\pi) \geqslant -(|V| - 1)W$).

Proof We prove the part for Min, the other case is similar. The proof proceeds by induction on the size of a partial play $\pi = v_1 \dots v_k$ with $v_1 = v$. If $k \leqslant |V|$ then $\mathbf{TP}(\pi) = \sum_{i=1}^{k-1} \omega(v_i, v_{i+1}) \leqslant (k-1)W \leqslant (|V| - 1)W$. If $k \geqslant |V| + 1$ then there exists $i < j$ such that $v_i = v_j$. Assume by contradiction that $\mathbf{TP}(v_i \dots v_j) > 0$. Then the play $\pi' = v_1 \dots v_i \dots v_j (v_{i+1} \dots v_j)^\omega$ conforms to σ_{Min} and $\mathbf{TP}(\pi') = +\infty$ which contradicts $\mathsf{Val}(v, \sigma_{\mathsf{Min}}) \neq +\infty$. Therefore $\mathbf{TP}(v_i \dots v_j) \leqslant 0$. We have $\mathbf{TP}(\pi) = \mathbf{TP}(v_1 \dots v_i) + \mathbf{TP}(v_i \dots v_j) + \mathbf{TP}(v_{j+1} \dots v_k)$, and since $v_i = v_j$, $v_1 \dots v_i v_{j+1} \dots v_k$ is a finite play starting from v that conforms to σ_{Min}, and by induction hypothesis $\mathbf{TP}(v_1 \dots v_i v_{j+1} \dots v_k) \leqslant (|V| - 1)W$. Then $\mathbf{TP}(\pi) = \mathbf{TP}(v_1 \dots v_i v_{j+1} \dots v_k) + \mathbf{TP}(v_i \dots v_j) \leqslant \mathbf{TP}(v_1 \dots v_i v_{j+1} \dots v_k) \leqslant (|V| - 1)W$. □

This permits to bound the finite values $\mathsf{Val}(v)$ of vertices v of the game:

Corollary 24 *For all $v \in V$,* $\mathsf{Val}(v) \in [-(|V| - 1)W, (|V| - 1)W] \uplus \{-\infty, +\infty\}$.

Proof From the result of [12], we know that total-payoff games are memorylessly determined, i.e. there exists two memoryless strategies $\sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}}$ such that for all v, $\mathsf{Val}(v) = \mathsf{Val}(v, \sigma_{\mathsf{Max}}) = \mathsf{Val}(v, \sigma_{\mathsf{Min}})$. Assume that $\mathsf{Val}(v) \notin \{-\infty, +\infty\}$. Then since $\mathsf{Val}(v, \sigma_{\mathsf{Min}}) = \mathsf{Val}(v) \neq -\infty$, Lemma 23 shows that all finite play π that conforms to σ_{Min} verifies $\mathbf{TP}(\pi) \geqslant -(|V| - 1)W$, therefore $\mathsf{Val}(v) \geqslant -(|V| - 1)W$. One can similarly prove that $\mathbf{TP}(v) \leqslant (|V| - 1)W$. □

We now compare values in both games. A first lemma shows, in particular, that $\mathsf{Val}_{\mathcal{G}^n}(v, n) \leqslant \mathsf{Val}_{\mathcal{G}}(v)$, in case $\mathsf{Val}_{\mathcal{G}}(v) \neq +\infty$.

Lemma 25 *For all $m \in \mathbb{Z}$, $v \in V$, and $n \geqslant 1$, if $\mathsf{Val}_{\mathcal{G}}(v) \leqslant m$ then $\mathsf{Val}_{\mathcal{G}^n}(v, n) \leqslant m$.*

Proof By hypothesis and using the memoryless determinacy of total-payoff games [12], there exists a memoryless strategy σ_{Min} for Min in $\mathcal{G}$ such that $\mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Min}}) \leqslant m$. Let σ^m_{Min} be the strategy in $\mathcal{G}^n$ defined, for all finite play π, vertex v' and $1 \leqslant j \leqslant n$, by

$$\sigma^m_{\mathsf{Min}}\left(\pi\left(v', j\right)\right) = \left(\mathtt{in}, \sigma_{\mathsf{Min}}\left(\mathsf{proj}(\pi)v'\right), j\right),$$

$$\sigma^m_{\mathsf{Min}}\left(\pi\left(\mathtt{in}, v', j\right)\right) = \begin{cases} \left(v', j\right) & \text{if } \mathbf{TP}\left(\pi\left(\mathtt{in}, v', j\right)\right) \geqslant m+1, \\ \left(\mathtt{ex}, v', j\right) & \text{if } \mathbf{TP}\left(\pi\left(\mathtt{in}, v', j\right)\right) \leqslant m. \end{cases}$$

Intuitively σ^m_{Min} simulates σ_{Min}, and asks to leave the copy when the current total-payoff is less than or equal to m. Notice that, by construction of σ^m_{Min}, $\mathsf{proj}(\pi)$ conforms to σ_{Min}, if π conforms to σ^m_{Min}.

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

$$\mathbf{MCR}(\pi) = \mathbf{TP}\left(\pi'\left(\mathtt{in}, v', j\right)\left(\mathtt{ex}, v', j\right)\mathtt{t}\right) = \mathbf{TP}\left(\pi'\left(\mathtt{in}, v', j\right)\right) \leqslant m.$$

Then, assume, by contradiction, that there exists a play π starting in (v, n) and conforming to σ^m_{Min}, that does not encounter the target. Then, this means that Min does not ask $n+1$ times the ability to exit in π [since on the $(n+1)$th time that we jump in an exterior vertex, Max is forced to go to the target]. In particular, there exists $0 \leqslant j \leqslant n$ such that π is of the form $\pi'(v_1, j)(\mathtt{in}, v_2, j)(v_2, j)(\mathtt{in}, v_3, j)\ldots(v_k, j)(\mathtt{in}, v_k, j)\ldots$. Since for all i, $\sigma^m_{\mathsf{Min}}(\pi'(v_1, j)(\mathtt{in}, v_2, j)\ldots(\mathtt{in}, v_i, j)) = (v_i, j)$, we have that $\mathbf{TP}(\pi'(v_1, j)(\mathtt{in}, v_2, j)\ldots(\mathtt{in}, v_i, j)) \geqslant m+1$. Therefore, since every prefix of $\mathsf{proj}(\pi)$ is the projection of a prefix of π, Lemma 22 shows that $\mathbf{TP}(\mathsf{proj}(\pi)) \geqslant m+1 > m$, which raises a contradiction since $\mathsf{proj}(\pi)$ conforms to σ_{Min} and $\mathsf{Val}(v, \sigma_{\mathsf{Min}}) \leqslant m$. Hence every play that conforms to σ^m_{Min} encounters the target, and, hence, has value at most m. This implies that $\mathsf{Val}_{\mathcal{G}^n}(v, n) \leqslant m$. □

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

Lemma 26 *For all $m \leqslant (|V|-1)W+1$, $k \geqslant K$, and vertex v, if $\mathsf{Val}_{\mathcal{G}}(v) \geqslant m$ then $\mathsf{Val}_{\mathcal{G}^k}(v, k) \geqslant m$.*

Proof By hypothesis and using the memoryless determinacy proved by Gimbert and Zielonka [12], there exists a memoryless strategy σ_{Max} for Max in $\mathcal{G}$ such that $\mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Max}}) \geqslant m$. Let σ^m_{Max} be the strategy in $\mathcal{G}^K$ defined, for all finite play π, vertex v' and $1 \leqslant j \leqslant n$, by:

$$\sigma^m_{\mathsf{Max}}(\pi(v, j)) = (\mathtt{in}, \sigma_{\mathsf{Max}}(\mathsf{proj}(\pi)v), j),$$

$$\sigma^m_{\mathsf{Max}}(\pi(\mathtt{ex}, v, j)) = \begin{cases} (v, j-1) & \text{if } \mathbf{TP}(\pi) \leqslant m-1 \text{ and } j > 1, \\ \mathtt{t} & \text{otherwise.} \end{cases}$$

Intuitively σ^m_{Max} simulates σ_{Max}, and accepts to go to the target when the current total-payoff is greater than or equal to m.

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

We now show that, in $\mathcal{G}^k$, $\mathbf{MCR}(\pi) \geqslant m$ for all play π starting in (v, k) and conforming to σ^m_{Max}. There are three cases to consider.

1. If π does not reach the target, then $\mathbf{MCR}(\pi) = +\infty \geqslant m$.
2. If $\pi = \pi_k(\mathtt{ex}, v_k, k)\dots\pi_j(\mathtt{ex}, v_j, j)\mathtt{t}^\omega$ and $j > 1$ then,

$$\sigma^m_{\mathsf{Max}}\left(\pi_k(\mathtt{ex}, v_0, k)\dots\pi_j(\mathtt{ex}, v_j, j)\right) = \mathtt{t}.$$

Thus, using Lemma 22,

$$\begin{aligned}\mathbf{MCR}(\pi) &= \mathbf{TP}(\pi_k(\mathtt{ex}, v_k, k)\dots\pi_j(\mathtt{ex}, v_j, j)\mathtt{t}) \\ &= \mathbf{TP}(\mathsf{proj}(\pi_k(\mathtt{ex}, v_k, k)\dots\pi_j(\mathtt{ex}, v_j, j)\mathtt{t})) \\ &\geqslant \mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Max}}) \geqslant m.\end{aligned}$$

3. If $\pi = \pi_k(\mathtt{ex}, v_k, k)\dots\pi_1(\mathtt{ex}, v_1, 1)\mathtt{t}^\omega$, assume by contradiction that

$$\mathbf{TP}(\pi_k(\mathtt{ex}, v_k, k)\dots\pi_1) \leqslant m - 1.$$

Otherwise, we directly obtain $\mathbf{MCR}(\pi) \geqslant m$. Let $v^\star$ be a vertex that occurs at least $N = \lceil K/|V|\rceil = 2(|V|-1)W+1$ times in the sequence $v_1, \dots, v_k$: such a vertex exists, since otherwise $K \leqslant k \leqslant (N-1)|V|$ which contradicts the fact that $(N-1)|V| < K$. Let $j_1 > \dots > j_N$ be a sequence of indices such that $v_{j_i} = v^\star$ for all i. We give a new decomposition of π:

$$\pi = \pi'_1(\mathtt{ex}, v_{j_1}, j_1)\dots\pi'_N(\mathtt{ex}, v_{j_N}, j_N)\pi'_{N+1}.$$

Since π conforms to σ^m_{Max} and according to the assumption, we have that for all i,

$$\mathbf{TP}\left(\pi'_1(\mathtt{ex}, v_{j_1}, j_1)\dots\pi'_i\right) \leqslant m - 1.$$

We consider two cases.

(a) If there exists π'_i such that $\mathbf{TP}(\pi'_i) \leqslant 0$ then, let $\mathsf{proj}(\pi'_i) = u_1\dots u_\ell$ with $u_1 = u_\ell = v^\star$, Since π'_i conforms to σ^m_{Max}, $\mathsf{proj}(\pi'_i)$ conforms to σ_{Max}. Therefore the play

$$\widetilde{\pi} = \mathsf{proj}\left(\pi'_1(\mathtt{ex}, v_{j_1}, j_1)\dots\pi'_i(\mathtt{ex}, v_{j_i}, j_i)\right)(u_1\dots u_{\ell-1})^\omega$$

conforms to σ_{Max}. Furthermore, using again Lemma 22,

$$\mathbf{TP}(\widetilde{\pi}) = \liminf_{n\to+\infty}\left(\mathbf{TP}\left(\pi'_1(\mathtt{ex}, v_{j_1}, i_1)\dots\pi'_i(\mathtt{ex}, v_{j_i}, j_i)\right) + n\mathbf{TP}(u_1\dots u_\ell)\right)$$

and since $\mathbf{TP}(u_1\dots u_\ell) = \mathbf{TP}(\pi'_i) \leqslant 0$, we have

$$\mathbf{TP}(\widetilde{\pi}) \leqslant \mathbf{TP}\left(\pi'_1(\mathtt{ex}, v_{j_1}, i_1)\dots\pi'_i(\mathtt{ex}, v_{j_i}, j_i)\right) \leqslant m - 1.$$

Thus $\widetilde{\pi}$ is a play starting from v that conforms to σ_{Max} but whose total-payoff is strictly less than m, which raises a contradiction.

(b) If for all π'_i, $\mathbf{TP}(\pi'_i) \geqslant 1$ (notice that it is implied by $\mathbf{TP}(\pi'_i) > 0$). From Lemma 23, since $\mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Max}}) \geqslant m \neq -\infty$, we know that $\mathbf{TP}(\mathsf{proj}(\pi'_0)) \geqslant -(|V|-1)W$. From Lemma 22, $\mathbf{TP}(\pi'_0) \geqslant -(|V|-1)W$. Therefore

$$\begin{aligned}\mathbf{TP}\left(\pi'_1(\mathtt{ex}, v_{j_1}, i_1)\dots\pi'_N\right) &\geqslant -(|V|-1)W + N \\ &= (|V|-1)W + 1 \geqslant m\end{aligned}$$

which contradicts the assumption that

$$\mathbf{TP}\left(\pi'_1(\mathtt{ex}, v_{j_1}, i_1)\dots\pi'_N\right) < m.$$

We have shown that $\mathbf{MCR}(\pi) \geqslant m$ for all play π starting in (v, k) and conforming to σ^m_{Max}, which implies $\mathsf{Val}_{\mathcal{G}^K}((v, k), \sigma^m_{\mathsf{Max}}) \geqslant m$. □

Using the two last lemmas, we can now prove Proposition 21 by relating precisely values in $\mathcal{G}$ and $\mathcal{G}^k$.

Proof of Proposition 21 Let $v \in V$ be a vertex. We consider three cases:

- If $\mathsf{Val}_{\mathcal{G}}(v) = -\infty$, then for all m, $\mathsf{Val}_{\mathcal{G}}(v) \leqslant m$. By Lemma 25, $\mathsf{Val}_{\mathcal{G}^K}(v, K) \leqslant m$. Therefore $\mathsf{Val}_{\mathcal{G}^K}(v, K) = -\infty$.
- If $\mathsf{Val}_{\mathcal{G}}(v) = m \in [-(|V|-1)W, (|V|-1)W]$. Then, $m \leqslant \mathsf{Val}_{\mathcal{G}}(v) \leqslant m$. Thus, by Lemmas 25 and 26, $m \leqslant \mathsf{Val}_{\mathcal{G}^K}(v, K) \leqslant m$. Therefore $\mathsf{Val}_{\mathcal{G}^K}(v, K) = m$.
- If $\mathsf{Val}_{\mathcal{G}}(v) = +\infty$, then $\mathsf{Val}_{\mathcal{G}}(v) \geqslant (|V|-1)W + 1$. By Lemma 26, $\mathsf{Val}_{\mathcal{G}^K}(v, K) \geqslant (|V|-1)W + 1$. □

4.2 Value iteration algorithm for total-payoff games

By Proposition 21, an immediate way to obtain a value iteration algorithm for total-payoff games is to build game $\mathcal{G}^K$, run Algorithm 1 on it, and map the computed values back to $\mathcal{G}$. We take advantage of the structure of $\mathcal{G}^K$ to provide a better algorithm that avoids building $\mathcal{G}^K$. Intuitively, we first compute the values of the vertices in the *last copy of the game* (vertices of the form $(v, 1)$, $(\mathtt{in}, v, 1)$ and $(\mathtt{ex}, v, 1)$), then of those in the penultimate (vertices of the form $(v, 2)$, $(\mathtt{in}, v, 2)$ and $(\mathtt{ex}, v, 2)$), and so on.

We first sketch the intuitions that lead to the formalisation of these ideas. Let Z^j be a vector mapping each vertex v of $\mathcal{G}$ to the value $Z^j(v)$ of vertex (v, j) in $\mathcal{G}^K$. Then, we define an operator $\mathcal{H}$ such that $Z^{j+1} = \mathcal{H}(Z^j)$. Thus, assuming that we have computed the values of all vertices in the jth copy, $\mathcal{H}$ returns the value of the vertices in the $j+1$st copy. In other words, the definition of $\mathcal{H}(Y)$ for some vector Y is to extract from $\mathcal{G}^K$ one copy of the game (that we call $\mathcal{G}_Y$), and make Y appear in the weights of some edges as illustrated in Fig. 5. This game, $\mathcal{G}_Y$, simulates a play in $\mathcal{G}$ in which Min can opt for 'leaving the game' at each round (by moving to the target), obtaining $\max(0, Y(v))$, if v is the current vertex. Then, we can define $\mathcal{H}(Y)(v)$ as the value of v in $\mathcal{G}_Y$. By construction, it is easy to see that $Z^{j+1} = \mathcal{H}(Z^j)$ holds for all $j \geqslant 1$, i.e. that $\mathcal{H}$ indeed corresponds to computing the values of the vertices in the $j+1$st copy, given the values in the jth copy. Furthermore, letting $Z^0(v) = -\infty$ for all v, and $Z^1 = \mathcal{H}(Z^0)$, we will prove that: (i) $\mathcal{H}$ is monotonic, but may not be Scott-continuous; (ii) the sequence $(Z^j)_{j\geqslant 0}$ converges towards $\mathsf{Val}_{\mathcal{G}}$. These arguments are the main ideas justifying Algorithm 2 to solve total-payoff games. Intuitively, the outer loop computes, in variable Y, a non-decreasing sequence of vectors whose limit is $\mathsf{Val}_{\mathcal{G}}$, and that is stationary (this is not necessarily the case for the sequence $(Z^j)_{j\geqslant 0}$).

Algorithm 2: A value iteration algorithm for total-payoff games

Input: Total-payoff game $\mathcal{G} = \langle V, E, \omega, \mathbf{TP}\rangle$, W largest weight in absolute value

1 **foreach** $v \in V$ **do** $\mathsf{Y}(v) := -\infty$
2 **repeat**
3 **foreach** $v \in V$ **do** $\mathsf{Y}_{pre}(v) := \mathsf{Y}(v)$; $\mathsf{Y}(v) := \max(0, \mathsf{Y}(v))$; $\mathsf{X}(v) := +\infty$
4 **repeat**
5 $\mathsf{X}_{pre} := \mathsf{X}$
6 **foreach** $v \in V_{\mathsf{Max}}$ **do** $\mathsf{X}(v) := \max_{v' \in E(v)} \left[\omega\left(v, v'\right) + \min\left(\mathsf{X}_{pre}\left(v'\right), \mathsf{Y}\left(v'\right)\right)\right]$
7 **foreach** $v \in V_{\mathsf{Min}}$ **do** $\mathsf{X}(v) := \min_{v' \in E(v)} \left[\omega\left(v, v'\right) + \min\left(\mathsf{X}_{pre}\left(v'\right), \mathsf{Y}\left(v'\right)\right)\right]$
8 **foreach** $v \in V$ such that $\mathsf{X}(v) < -(|V|-1)W$ **do** $\mathsf{X}(v) := -\infty$
9 **until** $\mathsf{X} = \mathsf{X}_{pre}$
10 $\mathsf{Y} := \mathsf{X}$
11 **foreach** $v \in V$ such that $\mathsf{Y}(v) > (|V|-1)W$ **do** $\mathsf{Y}(v) := +\infty$
12 **until** $\mathsf{Y} = \mathsf{Y}_{pre}$
13 **return** Y

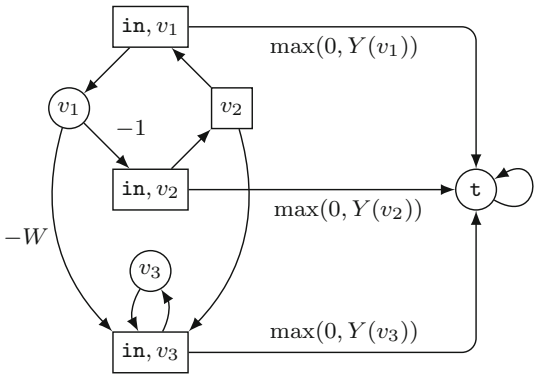

Fig. 5 MCR game $\mathcal{G}_Y$ associated with the total-payoff game of Fig. 2

Line 1 initialises Y to Z^0. Each iteration of the outer loop amounts to running Algorithm 1 to compute $\mathcal{H}(\mathsf{Y}_{pre})$ (lines 3–10), then detecting if some vertices have value $+\infty$, updating Y accordingly (line 11, following the second item of Proposition 21). We will show that, for all $j > 0$, if we let Y^j be the value of Y after the jth iteration of the main loop, then $Z^j \preccurlyeq Y^j \preccurlyeq \mathsf{Val}_{\mathcal{G}}$, which ensures the correctness of the algorithm.

Formal proof of the correctness of Algorithm 2 Let us now formalise the arguments sketched above to establish the correctness of Algorithm 2.

Theorem 27 *If a total-payoff game $\mathcal{G} = \langle V, E, \omega, \mathbf{TP}\rangle$ is given as input,* Algorithm 2 *outputs the vector $\mathsf{Val}_{\mathcal{G}}$ of optimal values, after at most $K = |V|(2(|V|-1)W+1)$ iterations of the external loop. The complexity of the algorithm is $O(|V|^4|E|W^2)$.*

We first define formally the game $\mathcal{G}_Y$ described informally above. With the original total-payoff game $\mathcal{G} = \langle V, E, \omega, \mathbf{TP}\rangle$ and with every vector $Y \in \mathbb{Z}_\infty^V$, we associate the MCR game $\mathcal{G}_Y = \langle V', E_Y, \omega_Y, \{\mathtt{t}\}\text{-}\mathbf{MCR}\rangle$ as follows. The sets of vertices are given by

$$V'_{\mathsf{Max}} = V_{\mathsf{Max}} \uplus \{\mathtt{t}\} \quad \text{and} \quad V'_{\mathsf{Min}} = V_{\mathsf{Min}} \uplus \{(\mathtt{in}, v) \mid v \in V\}.$$

As in game $\mathcal{G}^j$, vertices of the form $(\mathtt{in}, v)$ are called *interior vertices*. Edges are defined by

$$E_Y = \left\{\left(v, \left(\mathtt{in}, v'\right)\right) \mid \left(v, v'\right) \in E\right\} \uplus \{((\mathtt{in}, v), v) \mid v \in V\}$$
$$\uplus \{((\mathtt{in}, v), \mathtt{t}) \mid v \in V \wedge Y(v) \neq +\infty\} \uplus \{(\mathtt{t}, \mathtt{t})\}$$

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

$$\omega_Y\left(v, \left(\mathtt{in}, v'\right)\right) = \omega\left(v, v'\right), \qquad \omega_Y\left((\mathtt{in}, v), \mathtt{t}\right) = \max(0, Y(v)),$$
$$\omega_Y((\mathtt{in}, v), v) = 0.$$

It is easy to see that lines 3–10 are a rewriting of Algorithm 1 in the special case of game $\mathcal{G}_Y$: in particular, neither the target vertex nor interior vertices are explicit, but their behaviour is taken into account by the transformation performed in line 3 and the operators min used in the inner computation of lines 6 and 7. Hence, if we define $\mathcal{H}(Y)(v) = \mathsf{Val}_{\mathcal{G}_Y}(v)$ for all $v \in V$, we can say that if inside the main loop, at line 3 the variable Y has value Y, then after line 10, it has value $\mathcal{H}(Y)$.

Notice that the game $\mathcal{G}_Y$ resembles a copy of $\mathcal{G}$ in the game $\mathcal{G}^j$ of the previous section. More, precisely, from the values $(\mathsf{Val}_{\mathcal{G}^j}(v, j))_{v\in V}$ in the jth copy, we can deduce the values in the $(j+1)$th copy by an application of operator $\mathcal{H}$:

$$\left(\mathsf{Val}_{\mathcal{G}^{j+1}}(v, j+1)\right)_{v\in V} = \mathcal{H}\left((\mathsf{Val}_{\mathcal{G}^j}(v, j))_{v\in V}\right). \tag{5}$$

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

Notice the absence of exterior vertices $(\mathtt{ex}, v', j)$ in game $\mathcal{G}_Y$, replaced by the computation of the maximum between 0 and $X(v')$ on the edge towards the target. Before proving the correctness of Algorithm 2, we prove several interesting properties of operator $\mathcal{H}$.

Proposition 28 *$\mathcal{H}$ is a monotonic operator.*

Proof For every vector $Y \in \mathbb{Z}_\infty^V$, let $\mathcal{F}_Y$ be the operator associated with the MCR game as defined in Sect. 3, i.e. for all $X \in \mathbb{Z}_\infty^{V'}$, and for all $v_1 \in V'$

$$\mathcal{F}_Y(X)(v_1) = \begin{cases} \max_{v_2\in E_Y(v_1)}\left(\omega_Y(v_1, v_2) + X(v_2)\right) & \text{if } v \in V'_{\mathsf{Max}}\backslash\{\mathtt{t}\} \\ \min_{v_2\in E_Y(v_1)}\left(\omega_Y(v_1, v_2) + X(v_2)\right) & \text{if } v \in V_{\mathsf{Min}} \\ 0 & \text{if } v_1 = \mathtt{t}. \end{cases}$$

We know from Corollary 11 that $\mathsf{Val}_{\mathcal{G}_Y}$ is the greatest fixed point of $\mathcal{F}_Y$. Consider now two vectors $Y, Y' \in \mathbb{Z}_\infty^V$ such that $Y \preccurlyeq Y'$.

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

$$\mathcal{F}_Y(X) \preccurlyeq \mathcal{F}_{Y'}(X). \tag{6}$$

Indeed, to get the result it suffices to notice that for all $v_1, v_2 \in V'$, $\omega_Y(v_1, v_2) \leqslant \omega_{Y'}(v_1, v_2)$ [the only differences are on the edges $((\mathtt{in}, v), \mathtt{t})$].

Consider then the vector X_0 defined by $X_0(v_1) = +\infty$ for all $v_1 \in V'$. Given $Y \preccurlyeq Y'$, from (6), we have that $\mathcal{F}_Y(X_0) \preccurlyeq \mathcal{F}_{Y'}(X_0)$, then a simple induction shows that for all i, $\mathcal{F}_Y^i(X_0) \preccurlyeq \mathcal{F}_{Y'}^i(X_0)$. Thus, since $\mathsf{Val}_{\mathcal{G}_Y}$ (respectively, $\mathsf{Val}_{\mathcal{G}_{Y'}}$) is the greatest fixed point of $\mathcal{F}_Y$ [respectively, $\mathcal{F}_{Y'}$], we have $\mathsf{Val}_{\mathcal{G}_Y} \preccurlyeq \mathsf{Val}_{\mathcal{G}_{Y'}}$. As a consequence $\mathcal{H}(Y) \preccurlyeq \mathcal{H}(Y')$. □

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

Example 29 Recall that, in our setting, a Scott-continuous operator is a mapping $F : \mathbb{Z}_\infty^V \to \mathbb{Z}_\infty^V$ such that for every sequence of vectors $(x_i)_{i\geqslant 0}$ having a limit x_ω, the sequence $(F(x_i))_{i\geqslant 0}$ has a limit equal to $F(x_\omega)$.

We present a total-payoff game whose associated operator $\mathcal{H}$ is not continuous. Let $\mathcal{G}$ be the total-payoff game containing one vertex v of Min and a self loop of weight -1 (as

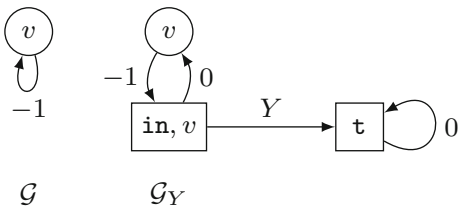

Fig. 6 An example where the operator $\mathcal{H}$ is not Scott-continuous

depicted in Fig. 6). For all $Y \in \mathbb{Z}$, in the MCR game $\mathcal{G}_Y$, 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 $(Y_i)_{i\geqslant 0}$ of integers, $\mathcal{H}(Y_i)(v) = -\infty$ for all i, thus the limit of the sequence $(\mathcal{H}(Y_i))_{i\geqslant 0}$ is $-\infty$. However, the limit of the sequence $(Y_i)_{i\geqslant 0}$ is $+\infty$ and $\mathcal{H}(+\infty)(v) = +\infty$, since the target is not reachable anymore (in case the weight of an edge would be $+\infty$, it is removed in the definition of E_Y). Thus, $\mathcal{H}$ is not Scott-continuous. □

In particular, we may not use the Kleene sequence, as we have done for MCR games, to conclude to the correctness of our algorithm. Anyhow, we will show that the sequence $(Y^j)_{j\geqslant 0}$ indeed converges towards the vector of values of the total-payoff game. We first show that this vector is a pre-fixed point of $\mathcal{H}$ starting with a technical lemma that is useful in the subsequent proof.

Lemma 30 *Let σ_{Min} be a strategy for* Min *in $\mathcal{G}$, and $\pi = v_1 \dots v_i$ a finite play that conforms to σ_{Min}. Then:*

$$\mathbf{TP}(v_1 \dots v_i) + \mathsf{Val}_{\mathcal{G}}(v_i) \leqslant \mathsf{Val}_{\mathcal{G}}(v_1, \sigma_{\mathsf{Min}}).$$

Proof Let σ_{Max} be an optimal strategy for Max and $v_i v_{i+1} v_{i+2} \dots$ be the play $\mathsf{Play}(v_i, \sigma_{\mathsf{Max}}, \sigma_{\mathsf{Min}})$. Since σ_{Max} is optimal, $\mathbf{TP}(v_i v_{i+1} v_{i+2} \dots) \geqslant \mathsf{Val}_{\mathcal{G}}(v_i)$. Furthermore notice that $v_1 \dots v_i v_{i+1} \dots$ conforms to the strategy σ_{Min}, therefore we have $\mathbf{TP}(v_1 v_1 \dots v_i v_{i+1} \dots) \leqslant \mathsf{Val}_{\mathcal{G}}(v_1, \sigma_{\mathsf{Min}})$. Thus:

$$\begin{aligned}\mathbf{TP}(v_1 \dots v_i) + \mathsf{Val}_{\mathcal{G}}(v_i) &\leqslant \mathbf{TP}(v_1 \dots v_i) + \mathbf{TP}(v_i v_{i+1} \dots) \\ &\leqslant \mathbf{TP}(v_1 v_1 \dots v_i v_{i+1} \dots) \leqslant \mathsf{Val}_{\mathcal{G}}(v_1, \sigma_{\mathsf{Min}}).\end{aligned}$$

□

Lemma 31 $\mathsf{Val}_{\mathcal{G}}$ *is a pre-fixed point of $\mathcal{H}$, i.e. $\mathcal{H}(\mathsf{Val}_{\mathcal{G}}) \preccurlyeq \mathsf{Val}_{\mathcal{G}}$.*

Proof To ease the notations, we denote $\mathsf{Val}_{\mathcal{G}}$ by $Y^\star$ in this proof. To prove this lemma, we just have to show that for all $v_1 \in V$, the value of v_1 in the MCR game $\mathcal{G}_{Y^\star}$ is at most its value in the original total-payoff game $\mathcal{G}$, i.e.

$$\mathcal{H}\left(Y^\star\right)(v_1) = \mathsf{Val}_{\mathcal{G}_{Y^\star}}(v_1) \leqslant Y^\star(v_1).$$

Let σ_{Min} be a memoryless strategy in $\mathcal{G}$ such that $\mathsf{Val}_{\mathcal{G}}(v_1, \sigma_{\mathsf{Min}}) \leqslant m$ for some $m \in \mathbb{Z} \uplus \{+\infty\}$. And let σ^m_{Min} be a strategy in $\mathcal{G}_X$ defined for all finite play $\pi v'$ with $v' \in V_{\mathsf{Min}}$ by $\sigma^m_{\mathsf{Min}}(\pi v') = (\mathtt{in}, \sigma_{\mathsf{Min}}(v'))$ and for all finite play $\pi(\mathtt{in}, v')$,

$$\sigma^m_{\mathsf{Min}}\left(\pi\ (\mathtt{in}, v')\right) = \begin{cases} \mathtt{t} & \text{if } \mathbf{TP}\left(\pi\ (\mathtt{in}, v')\ \mathtt{t}\right) \leqslant m \\ v' & \text{otherwise.} \end{cases}$$

Notice that, by construction, all plays starting in v_1, conforming to σ^m_{Min} and that reach the target have value at most m. Assume by contradiction that there exists a play $v_1(\mathtt{in}, v_2)v_2(\mathtt{in}, v_3)\ldots \in \mathsf{Play}(v_1, \sigma^m_{\mathsf{Min}})$ that never reaches $\mathtt{t}$. In particular, for all i, $\mathbf{TP}(v_1(\mathtt{in}, v_2)\ldots(\mathtt{in}, v_i)\mathtt{t}) > m$.

Again by construction, $v_1v_2\ldots$ is a play in $\mathcal{G}$ that conforms to σ_{Min} and for all i, $\mathbf{TP}(v_1(\mathtt{in}, v_2)\ldots(\mathtt{in}, v_i)) = \mathbf{TP}(v_1\ldots v_i)$.

– If there exists $i \geqslant 2$ such that $Y^\star(v_i) = \mathsf{Val}_{\mathcal{G}}(v_i) \geqslant 0$, then

$$\begin{aligned}\mathbf{TP}(v_1(\mathtt{in}, v_2)\ldots(\mathtt{in}, v_i)\mathtt{t}) &= \mathbf{TP}(v_1(\mathtt{in}, v_2)\ldots(\mathtt{in}, v_i)) + \omega_X((\mathtt{in}, v_i), \mathtt{t})\\ &= \mathbf{TP}(v_1\ldots v_i) + \max\left(0, Y^\star(v_i)\right)\\ &= \mathbf{TP}(v_1\ldots v_i) + Y^\star(v_i)\\ &\leqslant \mathsf{Val}(v_1, \sigma_{\mathsf{Min}}) \qquad \text{(from Lemma 30)}\\ &\leqslant m\end{aligned}$$

which raises a contradiction.

– If for all $i \geqslant 2$, $Y^\star(v_i) = \mathsf{Val}_{\mathcal{G}}(v_i) < 0$ then for all $i \geqslant 2$,

$$\mathbf{TP}(v_1(\mathtt{in}, v_2)\ldots(\mathtt{in}, v_i)\mathtt{t}) = \mathbf{TP}(v_1(\mathtt{in}, v_2)\ldots(\mathtt{in}, v_i)) = \mathbf{TP}(v_1\ldots v_i) > m.$$

Thus $\mathbf{TP}(v_1v_2\ldots) > m$, which contradicts the fact that $v_1v_2\ldots$ conforms to σ_{Min} and $\mathsf{Val}_{\mathcal{G}}(v_1, \sigma_{\mathsf{Min}}) \leqslant m$.

Thus $\mathsf{Val}_{\mathcal{G}_{Y^\star}}(v_1, \sigma_{\mathsf{Min}}) \leqslant m$. As a consequence, $\mathsf{Val}_{\mathcal{G}_{Y^\star}}(v_1) \leqslant \mathsf{Val}_{\mathcal{G}}(v_1)$. □

Remark 32 Even if it is not necessary for the proof of Theorem 27, we can show that $\mathsf{Val}_{\mathcal{G}}$ is the least pre-fixed point of $\mathcal{H}$. Notice that, by monotonicity of $\mathcal{H}$, this directly implies that $\mathsf{Val}_{\mathcal{G}}$ is the least fixed point of $\mathcal{H}$. The proof that $\mathsf{Val}_{\mathcal{G}}$ is the least pre-fixed point of $\mathcal{H}$ amounts to better understand the convergence of the sequence $(\mathsf{Val}_{\mathcal{G}^j}(v, j))_{j\geqslant 0}$ [remember that we set $\mathsf{Val}_{\mathcal{G}^0}(v, 0) = -\infty$ for all v]. Indeed, Proposition 21 already shows that for vertices v such that $\mathsf{Val}_{\mathcal{G}}(v) < +\infty$, $(\mathsf{Val}_{\mathcal{G}^j}(v, j))_{j\geqslant 1}$ converges towards $\mathsf{Val}_{\mathcal{G}}(v)$. It is also the case for vertices v such that $\mathsf{Val}_{\mathcal{G}}(v) = +\infty$ as we show in Lemma 33 below. Then, consider a pre-fixed point Y of $\mathcal{H}$, i.e. $\mathcal{H}(Y) \preccurlyeq Y$. Since $\mathsf{Val}_{\mathcal{G}^0}(v, 0) = -\infty \leqslant Y(v)$ and $\mathcal{H}$ is monotonous, we prove by immediate induction that $\mathsf{Val}_{\mathcal{G}^j}(v, j) \leqslant Y(v)$ for all v and $j \geqslant 0$: indeed, if $\mathsf{Val}_{\mathcal{G}^j}(v, j) \leqslant Y(v)$ for all v, we have

$$(\mathsf{Val}_{\mathcal{G}^{j+1}}(v, j+1))_{v\in V} = \mathcal{H}\left((\mathsf{Val}_{\mathcal{G}^j}(v, j))_{v\in V}\right) \preccurlyeq \mathcal{H}(Y) \preccurlyeq Y.$$

This implies that $\mathsf{Val}_{\mathcal{G}} \preccurlyeq Y$, showing that $\mathsf{Val}_{\mathcal{G}}$ is indeed the least pre-fixed point of $\mathcal{H}$, and hence the least fixed point of $\mathcal{H}$, by the above reasoning.

Before continuing the proof of Theorem 27, we show the result used in the previous remark.

Lemma 33 *Let $v \in V$ such that $\mathsf{Val}_{\mathcal{G}}(v) = +\infty$, and σ_{Max} a memoryless strategy for* Max *in $\mathcal{G}$ such that $\mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Max}}) = +\infty$. Then the following holds:*

(*i*) *For every finite play $v_1\ldots v_k$ conforming to σ_{Max} starting in $v_1 = v$, if there exists $i < j$ such that $v_i = v_j$ then $\mathbf{TP}(v_i\ldots v_j) \geqslant 1$.*
(*ii*) *For every $m \in \mathbb{N}$, $k \geqslant m|V| + 1$ and $v_1\ldots v_k$ a finite play conforming to σ_{Max} and starting in $v_1 = v$, $\mathbf{TP}(v_1\ldots v_k) \geqslant m - (|V| - 1)W$.*
(*iii*) *For all $m \in \mathbb{N}$ and $k \geqslant (m + (|V| - 1)W)|V| + 1$, $\mathsf{Val}_{\mathcal{G}^k}(v, k) \geqslant m$.*
(*iv*) $\lim_{j\to\infty} \mathsf{Val}_{\mathcal{G}^j}(v, j) = +\infty$.

Proof We prove (i) by contradiction. Therefore, assume that $\mathbf{TP}(v_i \dots v_j) \leqslant 0$, then $\pi = v_1 \dots v_{i-1}(v_i \dots v_{j-1})^\omega$ conforms to σ_{Max} and $\mathbf{TP}(\pi) \leqslant \mathbf{TP}(v_1 \dots v_{i-1}) < +\infty$, which contradicts the fact that $\mathsf{Val}_{\mathcal{G}}(v, \sigma_{\mathsf{Max}}) = +\infty$.

We prove (ii) by induction on m. The base case is straightforward. For the inductive case, let $m > 0$, $k \geqslant m|V| + 1$ and $v_1 \dots v_k$ a finite play conforming to σ_{Max} and starting in $v_1 = v$. Since $k \geqslant m|V|+1 \geqslant |V|+1$ there exists $i < j \leqslant |V|+1$ such that $v_i = v_j$. Thus:

$$\begin{aligned}\mathbf{TP}(v_1 \dots v_k) &= \mathbf{TP}(v_1 \dots v_i) + \mathbf{TP}(v_i \dots v_j) + \mathbf{TP}(v_j \dots v_k)\\ &= \mathbf{TP}(v_1 \dots v_i v_{j+1} \dots v_k) + \mathbf{TP}(v_i \dots v_j)\\ &\geqslant \mathbf{TP}(v_1 \dots v_i v_{j+1} \dots v_k) + 1. \qquad \text{from } (i)\end{aligned}$$

By induction hypothesis, as $v_1 \dots v_i v_{j+1} \dots v_k$ conforms to σ_{Max} and has length at least $(m-1)|V| + 1$ (because $i < j \leqslant |V| + 1$, implying that $j - i \leqslant |V|$), we have $\mathbf{TP}(v_1 \dots v_i v_{j+1} \dots v_k) \geqslant m - 1 - (|V| - 1)W$, thus $\mathbf{TP}(v_1 \dots v_k) \geqslant m - (|V| - 1)W$.

To prove (iii), let σ'_{Max} be a strategy of Max in $\mathcal{G}^k$ defined by $\sigma'_{\mathsf{Max}}(v, j) = (\mathtt{in}, \sigma_{\mathsf{Max}}(v), j)$ and $\sigma'_{\mathsf{Max}}(\mathtt{ex}, v, j) = (v, j-1)$ for all $v \in V$ and $j \geqslant k$. Let π be a play starting in (v, k) and conforming to σ'_{Max}. If π does not reach $\mathtt{t}$, then $\mathbf{MCR}(\pi) = +\infty \geqslant m$. If π reaches the target then $\mathsf{proj}(\pi)$ is of the form $v_1 \dots v_\ell \mathtt{t}^\omega$, with $\mathbf{MCR}(\pi) = \mathbf{TP}(v_1 \dots v_\ell)$. It is clear by construction of σ'_{Max} that $v_1 \dots v_\ell$ is a finite play of $\mathcal{G}$ that conforms to σ_{Max}. Furthermore, $\ell \geqslant k \geqslant (m+(|V|-1)W)|V|+1$ thus, from (ii), we have that $\mathbf{TP}(v_1 \dots v_\ell) \geqslant m$. This implies $\mathbf{MCR}(\pi) \geqslant m$. Hence, every play in $\mathcal{G}^k$ conforming to σ'_{Max} and starting in (v, k) has a value at least m, which means that $\mathsf{Val}_{\mathcal{G}^k}(v, k) \geqslant m$.

Item (iv) is then a direct consequence of (iii). □

We are now ready to state and prove the inductive invariant allowing us to show the correctness of Algorithm 2.

Lemma 34 *Before the jth iteration of the external loop of Algorithm 2, we have $\mathsf{Val}_{\mathcal{G}^j}(v, j) \leqslant Y^j(v) \leqslant \mathsf{Val}_{\mathcal{G}}(v)$ for all vertices $v \in V$.*

Proof For $j = 0$, we have $Y^0(v) = -\infty = \mathsf{Val}_{\mathcal{G}^0}(v, 0)$ for all vertex $v \in V$. Suppose then that the invariant holds for $j \geqslant 0$. We know [see (5)] that $\mathsf{Val}_{\mathcal{G}^{j+1}}(v, j+1) = \mathcal{H}((\mathsf{Val}_{\mathcal{G}^j}(v', j))_{v' \in V})$. Moreover, after the assignment of line 10, by definition of $\mathcal{H}$, variable Y contains $\mathcal{H}(Y^j)$. The operation performed on line 11 only increases the values of vector Y, so that at the end of the jth iteration, we have $\mathcal{H}(Y^j) \preccurlyeq Y^{j+1}$. Since $\mathcal{H}$ is monotonous, and by the invariant at step j, we obtain

$$\mathsf{Val}_{\mathcal{G}^{j+1}}(v, j+1) = \mathcal{H}\left(\left(\mathsf{Val}_{\mathcal{G}^j}\left(v', j\right)\right)_{v' \in V}\right) \leqslant \mathcal{H}\left(Y^j\right) \leqslant Y^{j+1}.$$

Moreover, using again the monotony of $\mathcal{H}$ and Lemma 31, we have

$$\mathcal{H}\left(Y^j\right) \preccurlyeq \mathcal{H}(\mathsf{Val}_{\mathcal{G}}) \preccurlyeq \mathsf{Val}_{\mathcal{G}}.$$

A closer look at line 11 shows that $\mathcal{H}(Y^j)$ and Y^{j+1} coincide over vertices v such that $\mathcal{H}(Y^j)(v) \leqslant (|V|-1)W$, and otherwise $Y^{j+1}(v) = +\infty$. Hence, if $\mathcal{H}(Y^j)(v) \leqslant (|V|-1)W$, we directly obtain $Y^{j+1}(v) = \mathcal{H}(Y^j)(v) \leqslant \mathsf{Val}_{\mathcal{G}}(v)$. Otherwise, we know that $\mathsf{Val}_{\mathcal{G}}(v) > (|V| - 1)W$. By Corollary 24, we know that $\mathsf{Val}_{\mathcal{G}}(v) = +\infty$, so that $Y^{j+1}(v) = +\infty = \mathsf{Val}_{\mathcal{G}}(v)$. In the overall, we have proved

$$\mathsf{Val}_{\mathcal{G}^{j+1}}(v, j+1) \leqslant Y^{j+1}(v) \leqslant \mathsf{Val}_{\mathcal{G}}(v)$$

□

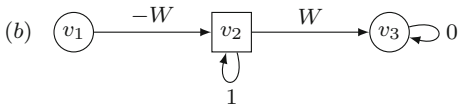

Fig. 7 An example total-payoff game where each execution of the inner loop requires two iterations while the outer loop takes W iterations to stabilise

We are now able to prove the correctness and termination of the algorithm.

Proof of Theorem 27 For $j = K$ (remember that K was defined in the previous section), the invariant of Lemma 34 becomes

$$\mathsf{Val}_{\mathcal{G}^K}(v, K) \leqslant Y^K(v) \leqslant \mathsf{Val}_{\mathcal{G}}(v)$$

for all vertices $v \in V$. Notice that the iteration may have stopped before iteration K, in which case the sequence $(Y^j)_{j \geqslant 0}$ may be considered as stationary. In case $\mathsf{Val}_{\mathcal{G}}(v) \neq +\infty$, Proposition 21 proves that $\mathsf{Val}_{\mathcal{G}^K}(v, K) = \mathsf{Val}_{\mathcal{G}}(v)$, so that we have $Y^K(v) = \mathsf{Val}_{\mathcal{G}}(v)$. In case $\mathsf{Val}_{\mathcal{G}}(v) = +\infty$, Proposition 21 shows that $\mathsf{Val}_{\mathcal{G}^K}(v, K) > (|V| - 1)W$: by the operation performed at line 11, we obtain that $Y^K(v) = +\infty = \mathsf{Val}_{\mathcal{G}}(v)$.

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

Convergence speed of the algorithm Observe that the number of iterations in each internal loop is controlled by Theorem 3. On the example of Fig. 2, only two external iterations are necessary, but the number of iterations of each internal loop would be $2W$. By contrast, for the total-payoff game depicted in Fig. 7, each internal loop requires two iterations to converge, but the external loop takes W iterations to stabilise. A combination of both examples would experience a pseudo-polynomial number of iterations to converge in both the internal and external loops, matching the W^2 term of the above complexity: this gives rise to the parametric example of Fig. 8.

4.3 Optimal strategies

In Sect. 3, we have shown, for all MCR games, the existence of a fake-optimal NC-strategy permitting to reconstruct an optimal finite-memory strategy for Min (if every vertex has value different from $-\infty$, or a strategy ensuring every possible threshold for vertices with value $-\infty$). Given a total-payoff game $\mathcal{G}$, if we apply this construction to the game $\mathcal{G}_{\mathsf{Val}_{\mathcal{G}}}$, we obtain an NC-strategy $\sigma^\star_{\mathsf{Min}}$. Consider the strategy $\overline{\sigma_{\mathsf{Min}}}$, obtained by projecting $\sigma^\star_{\mathsf{Min}}$ on V as follows: for all finite plays π and vertices $v \in V_{\mathsf{Min}}$, we let $\overline{\sigma_{\mathsf{Min}}}(\pi v) = v'$ if $\sigma^\star_{\mathsf{Min}}(v) = (\mathtt{in}, v')$. We show thereafter that $\overline{\sigma_{\mathsf{Min}}}$ is optimal for Min in $\mathcal{G}$. Notice that $\sigma^\star_{\mathsf{Min}}$, and hence $\overline{\sigma_{\mathsf{Min}}}$, can be computed during the last iteration of the value iteration algorithm, as explained in the case of MCR games in Sect. 3.4. A similar construction can be done to compute an optimal strategy for Max.

Theorem 35 *The memoryless strategy $\overline{\sigma_{\mathsf{Min}}}$ is optimal in $\mathcal{G}$.*

Proof We start by showing that for all v,

$$(1)\ \text{if } \sigma^\star_{\mathsf{Min}}(\mathtt{in}, v) = \mathtt{t}, \quad \text{then} \quad \mathsf{Val}_{\mathcal{G}}(v) = \omega((\mathtt{in}, v), \mathtt{t}) \geqslant 0.$$

By definition of $\sigma^\star_{\mathsf{Min}}$, $\mathtt{t} = \mathrm{argmin}_{v' \in E(\mathtt{in}, v)}(\omega((\mathtt{in}, v), v') + \mathsf{Val}_{\mathcal{G}_{\mathsf{Val}_{\mathcal{G}}}}(v'))$. In particular, for $v' = v$, as $\omega((\mathtt{in}, v), v) = 0$, we have $\mathsf{Val}_{\mathcal{G}_{\mathsf{Val}_{\mathcal{G}}}}(v) \geqslant \omega((\mathtt{in}, v), \mathtt{t}) + \mathsf{Val}_{\mathcal{G}_{\mathsf{Val}_{\mathcal{G}}}}(\mathtt{t})$, i.e.

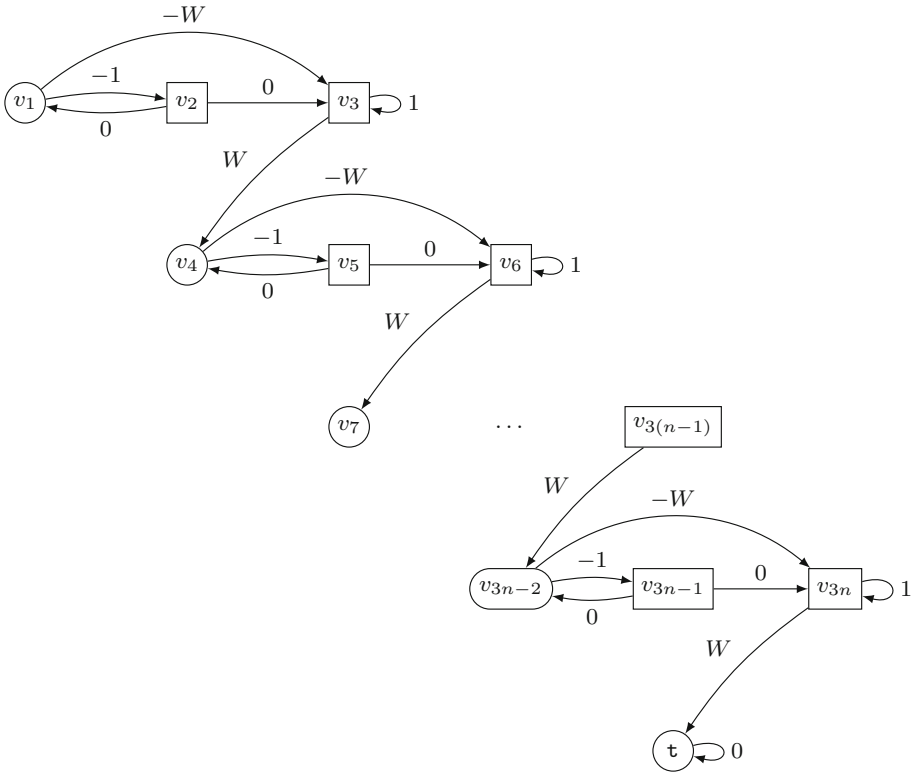

Fig. 8 Parametric weighted graph

$\mathsf{Val}_{\mathcal{G}}(v) \geqslant \omega((\mathtt{in}, v), \mathtt{t})$. Reciprocally, we know that $\omega((\mathtt{in}, v), \mathtt{t}) = \max(\mathsf{Val}_{\mathcal{G}}(v), 0) \geqslant \mathsf{Val}_{\mathcal{G}}(v)$, so that, we finally have $\mathsf{Val}_{\mathcal{G}}(v) = \omega((\mathtt{in}, v), \mathtt{t})$.

We then show that

(2) for all plays $\pi = v_0 v_1 \dots$ in $\mathcal{G}$ that conforms with $\overline{\sigma_{\mathsf{Min}}}$ and such that $\mathsf{Val}_{\mathcal{G}}(v_0) < +\infty$, either $\mathbf{TP}(\pi) = -\infty$ or there exists $i_\pi > 0$ such that $\sigma^\star_{\mathsf{Min}}(\mathtt{in}, v_{i_\pi}) = \mathtt{t}$ and $\mathbf{TP}(v_0 \dots v_{i_\pi}) \leqslant \mathsf{Val}_{\mathcal{G}}(v_0) - \mathsf{Val}_{\mathcal{G}}(v_{i_\pi})$.

Assume first that during π, Min never asks to go to the target, i.e. for all $i > 0$, $\sigma^\star_{\mathsf{Min}}(\mathtt{in}, v_i) = v_i$. Then the play $v_0(\mathtt{in}, v_1)v_1(\mathtt{in}, v_2)\dots$ is a play of $\mathcal{G}_{\mathsf{Val}_{\mathcal{G}}}$ that conforms with $\sigma^\star_{\mathsf{Min}}$. As there are only finitely many vertices in $\mathcal{G}_{\mathsf{Val}_{\mathcal{G}}}$, there must exist a vertex v that appears infinitely often in this play. As $\sigma^\star_{\mathsf{Min}}$ is an NC-strategy, the accumulated cost of the chunk of the play between two occurrences of v has weight at most -1, thus the total payoff of $v_0(\mathtt{in}, v_1)v_1(\mathtt{in}, v_2)\dots$ is $-\infty$. As the total-payoff of this play is equal to the total payoff of π, we have that $\mathbf{TP}(\pi) = -\infty$.

Otherwise, Min asks at some point to go to the target: let i_π be the first index such that $\sigma^\star_{\mathsf{Min}}(\mathtt{in}, v_{i_\pi}) = \mathtt{t}$. As the strategy $\sigma^\star_{\mathsf{Min}}$ is a fake-optimal NC-strategy, we know that the accumulated cost of π until the target verifies $\mathbf{TP}(\pi) = \mathbf{MCR}(\pi) \leqslant \mathsf{Val}_{\mathcal{G}_{\mathsf{Val}_{\mathcal{G}}}}(v_0)$. As $\mathsf{Val}_{\mathcal{G}}$ is a fixed point of $\mathcal{H}$ (see Remark 32), we have that for all v, $\mathsf{Val}_{\mathcal{G}_{\mathsf{Val}_{\mathcal{G}}}}(v) = \mathcal{H}(\mathsf{Val}_{\mathcal{G}}(v)) = \mathsf{Val}_{\mathcal{G}}(v)$, thus $\mathbf{TP}(\pi) \leqslant \mathsf{Val}_{\mathcal{G}}(v_0)$. We have $\mathbf{TP}(v_0 \dots v_{i_\pi}) = \mathbf{TP}(v_0(\mathtt{in}, v_1)$

Table 1 Results of value iteration on a parametric example

W	n	Without heuristics			With heuristics		
		t (s)	k_e	k_i	t (s)	k_e	k_i
50	100	0.52	151	12,603	0.01	402	1404
50	500	9.83	551	53,003	0.42	2002	7004
200	100	2.96	301	80,103	0.02	402	1404
200	500	45.64	701	240,503	0.47	2002	7004
500	1000	536	1501	1,251,003	2.37	4002	14,004

$v_1 \dots v_{i_\pi - 1}(\mathtt{in}, v_{i_\pi})) = \mathbf{TP}(\pi) - \omega((\mathtt{in}, v_{i_\pi}), \mathtt{t})$. From (1), we have $\omega((\mathtt{in}, v_{i_\pi}), \mathtt{t}) = \mathsf{Val}_{\mathcal{G}}(v_{i_\pi})$, thus $\mathbf{TP}(v_0 \dots v_{i_\pi}) \leqslant \mathsf{Val}_{\mathcal{G}}(v_0) - \mathsf{Val}_{\mathcal{G}}(v_{i_\pi})$, which proves (2).

Now let us prove the theorem. Let v be a vertex in $\mathcal{G}$. If $\mathsf{Val}_{\mathcal{G}}(v) = +\infty$ then trivially $\mathsf{Val}_{\mathcal{G}}(v, \overline{\sigma_{\mathsf{Min}}}) \leqslant +\infty = \mathsf{Val}_{\mathcal{G}}(v)$. Otherwise let $\pi = v_0 v_1 \dots$ be a play in $\mathcal{G}$ that conforms with $\overline{\sigma_{\mathsf{Min}}}$ such that $v = v_0$. If $\mathbf{TP}(\pi) = -\infty$ then $\mathbf{TP}(\pi) \leqslant \mathsf{Val}_{\mathcal{G}}(v)$. Otherwise we construct inductively an increasing sequence of indices $i_0, i_1, \dots$ such that $\mathbf{TP}(v_{i_j} \dots v_{i_{j+1}}) \leqslant \mathsf{Val}_{\mathcal{G}}(v_{i_j}) - \mathsf{Val}_{\mathcal{G}}(v_{i_{j+1}})$ (for all j) as follows. First, we let $i_0 = 0$. Then, for all j, let $\pi' = v_{i_j} v_{i_j+1} \dots$ be the current suffix of π: since $\mathbf{TP}(\pi) \neq -\infty$, we have $\mathbf{TP}(\pi') \neq -\infty$, thus (2) shows that by letting $i_{j+1} = i_j + i_{\pi'}$, we obtain $\mathbf{TP}(v_{i_j} \dots v_{i_{j+1}}) \leqslant \mathsf{Val}_{\mathcal{G}}(v_{i_j}) - \mathsf{Val}_{\mathcal{G}}(v_{i_{j+1}})$, and we know that $\sigma^\star_{\mathsf{Min}}(\mathtt{in}, v_{i_{j+1}}) = \mathtt{t}$.

We can then show, by a direct induction, that for all $j \geqslant 1$, $\mathbf{TP}(v_0 \dots v_{i_j}) \leqslant \mathsf{Val}_{\mathcal{G}}(v) - \mathsf{Val}_{\mathcal{G}}(v_{i_j})$. From (1), we know that $\mathsf{Val}_{\mathcal{G}}(v_{i_j}) \geqslant 0$, thus $\mathbf{TP}(v_0 \dots v_{i_j}) \leqslant \mathsf{Val}_{\mathcal{G}}(v)$. Hence

$$\mathbf{TP}(\pi) = \liminf_{n \to \infty} \mathbf{TP}(v_0 \dots v_n) \leqslant \liminf_{j \to \infty} \mathbf{TP}(v_0 \dots v_{i_j}) \leqslant \mathsf{Val}_{\mathcal{G}}(v).$$

□

5 Implementation and heuristics

In this section, we report on a prototype implementation of our algorithms.[7] For convenience reasons, we have implemented them as an add-on to PRISM-games [7], although we could have chosen to extend another model-checker as we do not rely on the probabilistic features of PRISM models (i.e. we use the PRISM syntax of *stochastic multi-player games*, allowing arbitrary rewards, and forbidding probability distributions different of Dirac ones). We then use rPATL specifications of the form $\langle\!\langle C \rangle\!\rangle \mathsf{R}^{\min / \max =?}[\mathsf{F}^\infty \varphi]$ and $\langle\!\langle C \rangle\!\rangle \mathsf{R}^{\min / \max =?}[\mathsf{F}^c \bot]$ to model respectively MCR games and total-payoff games, where C represents a coalition of players that want to minimise/maximise the payoff, and φ is another rPATL formula describing the target set of vertices (for total-payoff games, such a formula is not necessary). We have tested our implementation on toy examples. On the parametric one Fig. 8, results obtained by applying our algorithm for total-payoff games are summarised in Table 1, where for each pair (W, n), we give the time t in seconds, the number k_e of iterations in the external loop, and the total number k_i of iterations in the internal loop.

Notice that due to the very little memory consumption of the algorithm, there is no risk of running out of memory. However, the execution time can become very large. For instance, in case $W = 500$ and $n = 1000$, the execution time becomes 536 s whereas the total number of iterations in the internal loop is greater than a million.

[7] Source and binary files, as well as some examples, can be downloaded from http://www.ulb.ac.be/di/verif/monmege/tool/TP-MCR/.

5.1 Acceleration techniques

We close this section by sketching two techniques that can be used to speed up the computation of the fixed point in Algorithms 1 and 2. We fix a weighted graph $\langle V, E, \omega \rangle$. Both accelerations rely on a topological order of the strongly connected components (SCC for short) of the graph, given as a function $\mathsf{c}\colon V \to \mathbb{N}$, mapping each vertex to its *component*, verifying that (i) $\mathsf{c}(V) = \{0, \ldots, p\}$ for some $p \geqslant 0$, (ii) $\mathsf{c}^{-1}(q)$ is a maximal SCC for all q, (iii) and $\mathsf{c}(v) \geqslant \mathsf{c}(v')$ for all $(v, v') \in E$.[8]

In case of an MRC game with $\mathtt{t}$ the unique target, $\mathsf{c}^{-1}(0) = \{\mathtt{t}\}$. Intuitively, c induces a directed acyclic graph whose vertices are the sets $\mathsf{c}^{-1}(q)$ for all $q \in \mathsf{c}(V)$, and with an edge (S_1, S_2) if and only if there are $v_1 \in S_1$, $v_2 \in S_2$ such that $(v_1, v_2) \in E$.

The *first acceleration heuristic* is a divide-and-conquer technique that consists in applying Algorithm 1 (or the inner loop of Algorithm 2) iteratively on each $\mathsf{c}^{-1}(q)$ for $q = 0, 1, 2, \ldots, p$, using at each step the information computed during steps $j < q$ (since the value of a vertex v depends only on the values of the vertices v' such that $\mathsf{c}(v') \leqslant \mathsf{c}(v)$).

The *second acceleration heuristic* consists in studying more precisely each component $\mathsf{c}^{-1}(q)$. Having already computed the optimal values $\mathsf{Val}(v)$ of vertices $v \in \mathsf{c}^{-1}(\{0, \ldots, q-1\})$, we ask an oracle to precompute a finite set $S_v \subseteq \mathbb{Z}_\infty$ of possible optimal values for each vertex $v \in \mathsf{c}^{-1}(q)$. For MCR games and the inner iteration of the algorithm for total-payoff games, one way to construct such a set S_v is to consider that possible optimal values are the one of non-looping paths inside the component exiting it, since, in MCR games, there exist optimal strategies for both players whose outcome is a non-looping path (see Sect. 3).

Algorithms 3 and 4 are enhanced versions of Algorithms 1 and 2 respectively, that apply these acceleration heuristics.

Algorithm 3: Accelerated value iteration algorithm for min-cost reachability games

Input: MCR game $\langle V, E, \omega, \{\mathtt{t}\}\text{-}MCR \rangle$, SCC-decomposition $\mathsf{c}\colon V \to \{0, 1, \ldots, p\}$ and an oracle $\mathcal{O}(q, v)$ outputting sets $(S_v)_{v \in \mathsf{c}^{-1}(q)}$

$\mathsf{X}(\mathtt{t}) := 0$
for $q = 1$ **to** p **do**
  $(S_v)_{v \in \mathsf{c}^{-1}(q)} := \mathcal{O}(q, \mathsf{X})$ /* Use of the oracle */
  foreach $v \in \mathsf{c}^{-1}(q)$ **do** $\mathsf{X}(v) := \max S_v$
  repeat
    $\mathsf{X}_{pre} := \mathsf{X}$
    foreach $v \in \mathsf{c}^{-1}(q) \cap V_{\mathsf{Max}}$ **do**
      $\mathsf{X}(v) := \max_{v' \in E(v)} \left(\omega\left(v, v'\right) + \mathsf{X}_{pre}\left(v'\right)\right)$
    foreach $v \in \mathsf{c}^{-1}(q) \cap V_{\mathsf{Min}}$ **do**
      $\mathsf{X}(v) := \min_{v' \in E(v)} \left(\omega\left(v, v'\right) + \mathsf{X}_{pre}\left(v'\right)\right)$
    foreach $v \in \mathsf{c}^{-1}(q)$ **do** $\mathsf{X}(v) := \max(S_v \cap [-\infty, \mathsf{X}(v)])$
  until $\mathsf{X} = \mathsf{X}_{pre}$
return X

Finally, we note that we can identify classes of weighted graphs for which there exists an oracle that runs in polynomial time and returns, for all vertices v, a set S_v of polynomial size. On such classes, Algorithms 1 and 2, enhanced with our two acceleration techniques, *run in polynomial time*. For instance, for all fixed positive integers L, the class of weighted

[8] Such a mapping is computable in linear time, e.g., by Tarjan's algorithm [18].

Algorithm 4: Accelerated value iteration algorithm for total-payoff games

Input: Total-payoff game $\mathcal{G} = \langle V, E, \omega, \mathbf{TP}\rangle$, SCC-decomposition $\mathsf{c}\colon V \to \{0, 1, \ldots, p\}$ and an oracle $\mathcal{O}(q, v)$ outputting sets $(S_v)_{v\in\mathsf{c}^{-1}(q)}$

foreach $v \in V$ **do** $\mathsf{Y}(v) := -\infty$
for $q = 0$ **to** p **do**
  $(S_v)_{v\in\mathsf{c}^{-1}(q)} := \mathcal{O}(q, \mathsf{Y})$ `/* Use of the oracle */`
  repeat
    foreach $v \in \mathsf{c}^{-1}(q)$ **do**
      $\mathsf{Y}_{pre}(v) := \mathsf{Y}(v)$; $\mathsf{Y}(v) := \max(0, \mathsf{Y}(v))$; $\mathsf{X}(v) := \max S_v$
    repeat
      $\mathsf{X}_{pre} := \mathsf{X}$
      foreach $v \in V_{\mathsf{Max}} \cap \mathsf{c}^{-1}(q)$ **do**
        $\mathsf{X}(v) := \max_{v'\in E(v)} \left[\omega\left(v, v'\right) + \min\left(\mathsf{X}_{pre}\left(v'\right), \mathsf{Y}\left(v'\right)\right)\right]$
      foreach $v \in V_{\mathsf{Min}} \cap \mathsf{c}^{-1}(q)$ **do**
        $\mathsf{X}(v) := \min_{v'\in E(v)} \left[\omega\left(v, v'\right) + \min\left(\mathsf{X}_{pre}\left(v'\right), \mathsf{Y}\left(v'\right)\right)\right]$
      foreach $v \in \mathsf{c}^{-1}(q)$ **do** $\mathsf{X}(v) := \max(S_v \cap [-\infty, \mathsf{X}(v)])$
    until $\mathsf{X} = \mathsf{X}_{pre}$
    $\mathsf{Y} := \mathsf{X}$
    foreach $v \in V$ such that $\mathsf{Y}(v) > (|V| - 1)W$ **do** $\mathsf{Y}(v) := +\infty$
  until $\mathsf{Y} = \mathsf{Y}_{pre}$
return Y

graphs where every component $\mathsf{c}^{-1}(q)$ uses at most L distinct weights (that can be arbitrarily large in absolute value) satisfies this criterion. Table 1 contains the results obtained with the heuristics on the parametric example presented before. Observe that the acceleration technique permits here to decrease drastically the execution time, the number of iterations in both loops depending not even anymore on W. Even though the number of iterations in the external loop increases with heuristics, due to the decomposition, less computation is required in each internal loop since we only apply the computation for the active component.

6 Conclusion

In this work, we have provided the first (to the best of our knowledge) pseudo-polynomial time algorithm to solve total-payoff games with arbitrary (positive and negative) weights. This algorithm is a variation on the classical value iteration technique. To obtain this algorithm, we have reduced the problem of solving total-payoff games to that of solving MCR games, a variant of the former where the game stops as soon as a target vertex is reached (in which case the payoff of the plays is the total accumulated weight of the play up to the target). We believe that those MCR games are interesting by themselves, as they can be used to model problems where, for instance, a target configuration must be reached while ensuring a minimal energy spending. Notice also that they have been used as a building block for the resolution of priced timed games in [3,4]. We have characterised the optimal strategies that one can extract in those total-payoff and MCR games. Finally, we have implemented our algorithms and proposed some heuristics that take into account the structure of the games to speed up the computation. As future works, we would like to push further the MCR games in a context of non-zero sum games where each player wants to optimise its accumulated cost until reaching its own target. As a possible direction, the search for Nash equilibria in

this context will most likely benefit from our better understanding of optimal strategies for both players in the underlying zero-sum games. This bridge from zero-sum to non-zero-sum games has already been investigated for concurrent priced games by Klimoš et al. [14], and Brihaye et al. [2] to find simple Nash equilibria for large classes of multiplayer cost games.

Acknowledgements The authors are indebted to Jean-François Raskin for enlightening discussions, and, in particular, for suggesting the example in Fig. 2. They also want to thank reviewers of this version, as well as the short version [5], that helped greatly improving its content.

Appendix: Comparison with the *longest shortest path problem* of Björklund and Vorobyov

Björklund and Vorobyov have studied games related to our min-cost reachability games, as the LSP in [1]. To clarify the following discussion, let us recall the definition of LSP, adapted to our syntax.

The LSP problem considers a weighted graph $\mathcal{G}$, whose vertices are partitioned into two players, and equipped with a single target vertex $\mathtt{t}$. The problem asks to find a *memoryless* strategy σ_{Max} of Max such that in the graph $\mathcal{G}_{\sigma_{\mathsf{Max}}}$, obtained from $\mathcal{G}$ by deleting all outgoing edges from vertices of V_{Max} except those selected in σ_{Max}, the length of the shortest path from every vertex to the target $\mathtt{t}$ is as large as possible (over all memoryless strategies). The definition of paths from a vertex v to the target is, however, very different from ours: such a path may indeed be a finite play, without cycle, from v to the target $\mathtt{t}$ (in which case its length is the sum of the weights of edges). However, every cycle containing v (even if it can not reach the target) is also considered as a path to the target. The *length* of such cycle is defined as follows:

1. if the cycle has a negative weight, its length is $-\infty$;
2. else, if it has a positive weight, its length is $+\infty$;
3. else, if it has zero weight, its length is 0.

Consider, as an example, the weighted graph of Fig. 9, where, once again, Max vertices are depicted with circles. Clearly, vertices $\{v_1, v_2, v_3\}$ and $\{v_4\}$ form independant subgames. We first consider the status of v_1, v_2 and v_3. In this subgame, Max has two possible strategies: σ_1 that selects the (v_1, v_2) edge, or σ_2 that selects the (v_1, v_3) edge instead.

1. If we select σ_1, we obtain the following values for v_1, v_2 and v_3 respectively: 0, 0 and 1. Indeed, from vertex v_2, we can loop in the zero cycle in-between v_1 and v_2, which is better than the simple path to the target (having weight 3), so that the distance from v_1 and v_2 is 0. From vertex v_3, only a single path leads to the target with weight 1.
2. If we select σ_2, we obtain the following values for v_1, v_2 and v_3 respectively: 1, 3 and 1. Indeed, from vertex v_3, the distance is the length of the simple path to the target, which is better than looping in-between v_1 and v_3 with a positive weight cycle. Hence, from vertex v_1, the distance is 1 too, whereas from v_2, the distance is 3.

Now, let us turn our attention to v_4. Again, we must consider two strategies: strategy σ_3 selects the (v_4, v_4) edge (the self-loop on v_4) and strategy σ_4 selects the $(v_4, \mathtt{t})$ edge.

1. If we select σ_3, we obtain value $-\infty$ for v_4, because all plays starting in v_4 loop in this negatively priced cycle. Recall that with our definition of min-cost reachability, such a cycle that never reaches the target would yield a total-payoff of $+\infty$, regardless of the weights. This is coherent with our intuition that the first goal of Min is to reach the target.
2. If we select σ_4, the value of v_4 is now 0, which is thus better for Max.

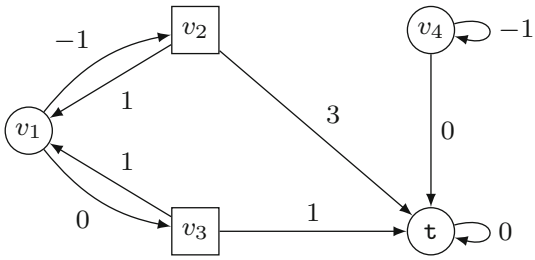

Fig. 9 An instance of the longest shortest path problem

Finally, the optimal strategy for Max is to choose edges (v_1, v_3) and $(v_4, \mathtt{t})$, leading to the LSP distances $(1, 3, 1, 0)$ for vertices (v_1, v_2, v_3, v_4). With our definition of min-cost reachability game, the situation is *completely different*: the value vector is then $(2, 3, 1, +\infty)$, associated with the optimal strategy σ_{Max} selecting edges (v_1, v_2) and (v_4, v_4).

Indeed, two main differences separate our two definitions. The first one is the treatment of negative weight cycles. Actually, in LSP, the fact that after selecting strategy σ_{Max} a vertex can not reach the target anymore in $\mathcal{G}_{\sigma_{\mathsf{Max}}}$ does not prevent from mapping the distance $-\infty$ to a vertex contained in a negative cycle. This is in contrast with our definition, that would benefit to Max since in such a situation, the target is not reachable leading to the value $+\infty$.[9] The second difference consists in the treatment of zero weight cycle. In LSP, a distance zero is then computed, which is highly related with the fact that the authors want to apply the resolution of LSP to mean-payoff games.

Nevertheless, in Section 9 of [1], the authors *briefly* study another more natural definition, mapping zero weight cycle to the distance $+\infty$ (closer to our definition). In that case, the LSP distances of vertices v_1, v_2 and v_3 then match our value vector in min-cost reachability games. Unfortunately, it is not clear how the algorithm that is presented by Björklund and Vorobyov can be adapted to accomodate this new definition of the cost and compute the values of the nodes. Indeed, Proposition 9.1 of [1] proves that the decision version of the LSP (i.e. deciding if the LSP of a vertex is positive or not in a given graph) is in NP ∩ co-NP , and they claim (without formal proof) that their pseudo-polynomial time algorithm permits to decide this problem. This requires a first transformation of the problem, so that no zero weight cycles remain in the weighted graph. This transformation, explained above Proposition 9.1, is correct but *does not preserve the LSP of the vertices*, so that their algorithm do not compute the LSP distance of vertices, but only study their positivity.

Hence, the comparison between our definition of min-cost reachability and the definition of LSP can be summarised as follows:

1. Either one relies on the first definition of LSP given in the article. In this case, Björklund and Vorobyov propose a pseudo-polynomial time algorithm to compute the LSP, but these values *do not match our definition of the min-cost reachability* payoff, as demonstrated by the above example.
2. Or one relies on the second definition. In this case, we believe that the LSP values may match our definition of min-cost reachability but the article *does not explain formally how to compute those new values*.

It is plausible that the pseudo-polynomial time algorithm of Björklund and Vorobyov could be adapted to compute the LSP value according to the second definition. Yet, even in this case, there are several points of comparison worth mentioning:

[9] We believe that this difference would certainly be eliminated by our preprocessing of vertices of value $+\infty$ presented in the first item of Theorem 3.

1. The worst-case complexity of the algorithm proposed by Björklund and Vorobyov is $O(|V|^2|E|W)$, which matches ours (see Proposition 5). Nevertheless, our solution is much easier to describe and to implement: while they must make repeated calls to a modified version of the Bellman–Ford algorithm (the modification is crucial to obtain their complexity), we have a simple fixed point algorithm.
2. Our algorithm exploits the value iteration paradigm (that can be seen as a *backward induction*), while theirs is a *strategy iteration algorithm*. Because of that, there are examples on which our algorithm is more efficient that theirs (although the worst-case complexity is the same). As an example, the LSP instance presented in Fig. 2 of [1], with $2n + 1$ vertices but a biggest weight W exponential in n, requires an exponential number 2^n of iterations for the strategy iteration algorithm, but our value iteration algorithm (based on backward induction) would solve it in linear time with respect to n. More precisely, with our tool, we are able to compute the values of this game in less than a millisecond for constant $n = 15$ (i.e. a game with 32 vertices and largest weight $W \approx 2.68 \times 10^8$): the value of the initial state obtained is 4.74×10^9, and as aforementioned, the number of iterations that our value iteration requires is 16, linear in n. Our tool being an add-on of the PRISM model-checker, relying on the use of integer rewards, we have not been able to build the game for a value $n > 15$.
3. We propose several acceleration heuristics that perform well on several examples we have tried. These accelerations cannot easily be incorporated in their algorithm (because it is a strategy iteration algorithm).
4. Finally, from the theoretical point of view, Björklund and Vorobyov do not study the strategies of the opponent player Min. In contrast, our study allows us to produce optimal strategies for both players, and in particular, show that memoryless strategies may not be sufficient for Min, whereas they are enough for Max.

References

1. Björklund, H., Vorobyov, S.: A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. Discret. Appl. Math. **155**, 210–229 (2007)
2. Brihaye, T., De Pril, J., Schewe., S.: Multiplayer cost games with simple nash equilibria. In: Proceedings of the International Symposium on Logical Foundations of Computer Science (LFCS'13). Lecture Notes in Computer Science, vol. 7734, pp. 59–73. Springer, Berlin (2013)
3. Brihaye, T., Geeraerts, G., Krishna, S.N., Manasa, L., Monmege, B., Trivedi, A.:Adding negative prices to priced timed games. In: Proceedings of the 25th International Conference on Concurrency Theory (CONCUR'14). Lecture Notes in Computer Science, vol. 8704, pp. 560–575. Springer, Berlin (2014). doi:10.1007/978-3-662-44584-63_8
4. Brihaye, T., Geeraerts, G., Haddad, A., Lefaucheux, E., Monmege, B.: Simple priced timed games are not that simple. In: Proceedings of the 35th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'15), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, LIPIcs (2015)
5. Brihaye, T., Geeraerts, G., Haddad, A., Monmege, B.: To reach or not to reach? Efficient algorithms for total-payoff games. In: Proceedings of the 26th International Conference on Concurrency Theory (CONCUR '15), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, LIPIcs, vol. 42, pp. 297–310 (2015)
6. Brim, L., Chaloupka, J., Doyen, L., Gentilini, R., Raskin, J.F.: Faster algorithms for mean-payoff games. Form. Methods Syst. Des. **38**(2), 97–118 (2011)
7. Chen, T., Forejt, V., Kwiatkowska, M., Parker, D., Simaitis, A.: Automatic verification of competitive stochastic systems. Form. Methods Syst. Des. **43**(1), 61–92 (2013)
8. Comin, C., Rizzi, R.: Improved Pseudo-polynomial bound for the value problem and optimal strategy synthesis in mean payoff games. Algorithmica (2016). doi:10.1007/s00453-016-0123-1
9. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. Int. J. Game Theory **8**(2), 109–113 (1979)

10. Filiot, E., Gentilini, R., Raskin, J.F.: Quantitative languages defined by functional automata. In: Proceedings of the 23rd International Conference on Concurrency theory (CONCUR '12). Lecture Notes in Computer Science, vol. 7454, pp. 132–146. Springer, Berlin (2012)
11. Gawlitza, T.M., Seidl, H.: Games through nested fixpoints. In: Proceedings of the 21st International Conference on Computer Aided Verification (CAV '09). Lecture Notes in Computer Science, vol. 5643, pp. 291–305. Springer, Berlin (2009)
12. Gimbert, H., Zielonka, W.: When can you play positionally? In: Proceedings of the 29th International Conference on Mathematical Foundations of Computer Science (MFCS '04). Lecture Notes in Computer Science, vol. 3153, pp. 686–698. Springer, Berlin (2004)
13. Khachiyan, L., Boros, E., Borys, K., Elbassioni, K., Gurvich, V., Rudolf, G., Zhao, J.: On short paths interdiction problems: total and node-wise limited interdiction. Theory Comput. Syst. **43**, 204–233 (2008)
14. Klimoš, M., Larsen, K.G., Štefaňák, F., Thaarup, J.: Nash equilibria in concurrent priced games. In: Proceedings of the 6th International Conference on Language and Automata Theory and Applications (LATA'12). Lecture Notes in Computer Science, vol. 7183, pp. 363–376. Springer, Berlin (2012)
15. Martin, D.A.: Borel determinacy. Ann. Math. **102**(2), 363–371 (1975)
16. Puterman, M.L.: Markov Decision Processes. Wiley, New York (1994)
17. Strauch, R.E.: Negative dynamic programming. Ann. Math. Stat. **37**, 871–890 (1966)
18. Tarjan, R.E.: Depth first search and linear graph algorithms. SIAM J. Comput. **1**(2), 146–160 (1972)
19. Thomas, W.: On the synthesis of strategies in infinite games. In: Symposium on Theoretical Aspects of Computer Science (STACS '95). Lecture Notes in Computer Science, vol. 900, pp. 1–13. Springer, Berlin (1995)
20. Thuijsman, F., Vrieze, O.J.: The bad match: a total reward stochastic game. Oper. Res. Spektrum **9**(2), 93–99 (1987)
21. Zwick, U., Paterson, M.S.: The complexity of mean payoff games. Theor. Comput. Sci. **158**, 343–359 (1996)