Measuring Permissiveness in Parity Games: Mean-Payoff Parity Games Revisited

Patricia Bouyer, Nicolas Markey, Jörg Olschewski, Michael Ummels

June 2011

Research report LSV-2011-17

Laboratoire Spécification & Vérification

École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Measuring Permissiveness in Parity Games: Mean-Payoff Parity Games Revisited⋆

Patricia Bouyer[1], Nicolas Markey[1], Jörg Olschewski[2], Michael Ummels[1,3]

[1] LSV, CNRS & ENS Cachan, France
{bouyer,markey,ummels}@lsv.ens-cachan.fr
[2] Lehrstuhl Informatik 7, RWTH Aachen University, Germany
olschewski@automata.rwth-aachen.de
[3] LAMSADE, CNRS & Université Paris-Dauphine, France

Abstract. We study nondeterministic strategies in parity games with the aim of computing a *most permissive* winning strategy. Following earlier work, we measure permissiveness in terms of the *average* number/weight of transitions blocked by a strategy. Using a translation into mean-payoff parity games, we prove that deciding (the permissiveness of) a most permissive winning strategy is in NP ∩ coNP. Along the way, we provide a new study of mean-payoff parity games. In particular, we give a new algorithm for solving these games, which beats all previously known algorithms for this problem.

1 Introduction

Games extend the usual semantics of finite automata from one to several players, thus allowing to model interactions between agents acting on the progression of the automaton. This has proved very useful in computer science, especially for the formal verification of open systems interacting with their environment [21]. In this setting, the aim is to synthesise a controller under which the system behaves according to a given specification, whatever the environment does. Usually, this is modelled as a game between two players: Player 1 represents the controller and Player 2 represents the environment. The goal is then to find a *winning strategy* for Player 1, i.e. a recipe stating how the system should react to any possible action of the environment, in order to meet its specification.

In this paper, we consider *multi-strategies* (or *non-deterministic strategies*, cf. [1, 3]) as a generalisation of strategies: while strategies select only one possible action to be played in response to the behaviour of the environment, multi-strategies can retain several possible actions. Allowing several moves provides a way to cope with errors (e.g., actions being disabled for a short period, or timing imprecisions in timed games). Another quality of multi-strategies is their ability to be combined with other multi-strategies, yielding a refined multi-strategy, which is ideally winning for all of the original specifications. This offers a modular approach for solving games.

⋆ Sponsored by ANR-06-SETI-003 DOTS, and by ESF-Eurocores LogICCC GASICS.

Classically, a strategy is more *permissive* than another one if it allows more behaviours. Under this notion, there does not need to exist a most permissive winning strategy [1]. Hence, we follow a different approach, which is of a quantitative nature: we provide a *measure* that specifies *how* permissive a given multi-strategy is. In order to do so, we consider *weighted games*, where each edge is equipped with a weight, which we treat as a *penalty* that is incurred when disallowing this edge. The penalty of a multi-strategy is then defined to be the average sum of penalties incurred in each step (in the limit). The lower this penalty is, the more permissive is the given multi-strategy. Our aim is to find one of the most permissive multi-strategies achieving a given objective.

We deal with multi-strategies by transforming a game with penalties into a *mean-payoff game* [11, 24] with classical (deterministic) strategies. A move in the latter game corresponds to a set of moves in the former, and is assigned a (negative) *reward* depending on the penalty of the original move. The penalty of a multi-strategy in the original game equals the opposite of the payoff achieved by the corresponding strategy in the mean-payoff game. In previous work, Bouyer et al. [3] introduced the notion of penalties and showed how to compute permissive strategies wrt. reachability objectives. We extend the study of [3] to parity objectives. This is a significant extension because parity objectives can express infinitary specifications. Using the above transformation, we reduce the problem of finding a most permissive strategy in a parity game with penalties to that of computing an optimal strategy in a *mean-payoff parity game*, which combines a mean-payoff objective with a parity objective.

While mean-payoff parity games have already been studied [8, 2, 6], we propose a new proof that these games are determined and that both players have optimal strategies. Moreover, we prove that the second player does not only have an optimal strategy with finite memory, but one that uses no memory at all. Finally, we provide a new algorithm for computing the values of a mean-payoff parity game, which is faster than the best known algorithms for this problem; the running time is exponential in the number of priorities and polynomial in the size of the game graph and the largest absolute weight.

In the second part of this paper, we present our results on parity games with penalties. In particular, we prove the existence of most permissive multi-strategies, and we show that the existence of a multi-strategy whose penalty is less than a given threshold can be decided in NP $\cap$ coNP. Finally, we adapt our deterministic algorithm for mean-payoff parity games to parity games with penalties. Our algorithm computes the penalties of a most permissive multi-strategy in time exponential in the number of priorities and polynomial in the size of the game graph and the largest penalty.

Related work. Penalties as we use them were defined in [3]. Other notions of permissiveness have been defined in [1, 20], but these notions have the drawback that a most permissive strategy might not exist. Multi-strategies have also been used for different purposes in [17].

The parity condition goes back to [12, 19] and is fundamental for verification. Parity games admit optimal memoryless strategies for both players, and the problem of deciding the winner is in NP $\cap$ coNP. As of this writing, it is not known whether parity games can be solved in polynomial time; the best known algorithms run in time polynomial in the size of the game graph but exponential in the number of priorities.

Another fundamental class of games are games with quantitative objectives. Mean-payoff games, where the aim is to maximise the average weight of the transitions taken in a play, are also in NP $\cap$ coNP and admit memoryless optimal strategies [11, 24]. The same is true for *energy games*, where the aim is to always keep the sum of the weights above a given threshold [5, 4]. In fact, parity games can easily be reduced to mean-payoff or energy games [14].

Finally, several game models mixing several qualitative or quantitative objectives have recently appeared in the literature: apart from mean-payoff parity games, these include generalised parity games [9], energy parity games [6] and lexicographic mean-payoff (parity) games [2] as well as generalised energy and mean-payoff games [7].

2 Preliminaries

A *weighted game graph* is a tuple $G = (Q_1, Q_2, E, \mathrm{weight})$, where $Q \coloneqq Q_1 \dot\cup Q_2$ is a finite set of *states*, $E \subseteq Q \times Q$ is the *edge* or *transition relation*, and $\mathrm{weight}\colon E \to \mathbb{R}$ is a function assigning a *weight* to every transition. When weighted game graphs are subject to algorithmic processing, we assume that these weights are integers; in this case, we set $W \coloneqq \max\{1, |\mathrm{weight}(e)| \mid e \in E\}$.

Moreover, we define the *size* of G, denoted by $\|G\|$, as $|Q| + |E| \cdot \lceil \log_2 W \rceil$. (Up to a linear factor, $\|G\|$ is the length of a binary encoding of G). In the same spirit, the size $\|x\|$ of a rational number x equals the total length of the binary representations of its numerator and its denominator.

For $q \in Q$, we write qE for the set $\{q' \in Q \mid (q, q') \in E\}$ of all successors of q. We require that $qE \neq \emptyset$ for all states $q \in Q$. A subset $S \subseteq Q$ is a *subarena* of G if $qE \cap S \neq \emptyset$ for all states $q \in S$. If $S \subseteq Q$ is a subarena of G, then we can restrict G to states in S, in which case we obtain the weighted game graph $G \restriction S \coloneqq (Q_1 \cap S, Q_2 \cap S, E \cap (S \times S), \mathrm{weight} \restriction S \times S)$.

A *play* of G is an infinite sequence $\rho = \rho(0)\rho(1) \cdots \in Q^\omega$ of states such that $(\rho(i), \rho(i+1)) \in E$ for all $i \in \mathbb{N}$. We denote by $\mathrm{Out}^{\mathcal{G}}(q)$ the set of all plays ρ with $\rho(0) = q$ and by $\mathrm{Inf}(\rho)$ the set of states occurring infinitely often in ρ.

A *play prefix* or a *history* $\gamma = \gamma(0)\gamma(1) \cdots \gamma(n) \in Q^+$ is a finite, nonempty prefix of a play. For a play or a history ρ and $j < k \in \mathbb{N}$, we denote by $\rho[j, k) \coloneqq \rho[j, k-1] \coloneqq \rho(j) \cdots \rho(k-1)$ its infix that starts at position j and ends at position $k-1$; the play's suffix $\rho(j)\rho(j+1)\cdots$ is denoted by $\rho[j, \infty)$.

Strategies. A *(deterministic) strategy* for Player i in G is a function $\sigma\colon Q^*Q_i \to Q$ such that $\sigma(\gamma q) \in qE$ for all $\gamma \in Q^*$ and $q \in Q_i$. A strategy σ is *memoryless* if $\sigma(\gamma q) = \sigma(q)$ for all $\gamma \in Q^*$ and $q \in Q_i$. More generally, a strategy σ is *finite-memory* if the equivalence relation $\sim\ \subseteq Q^* \times Q^*$, defined by $\gamma_1 \sim \gamma_2$ if and only if $\sigma(\gamma_1 \cdot \gamma) = \sigma(\gamma_2 \cdot \gamma)$ for all $\gamma \in Q^*Q_i$, has finite index.

We say that a play ρ of G is *consistent* with a strategy σ for Player i if $\rho(k+1) = \sigma(\rho[0,k])$ for all $k \in \mathbb{N}$ with $\rho(k) \in Q_i$, and denote by $\mathrm{Out}^G(\sigma, q_0)$ the set of all plays ρ of G that are consistent with σ and start in $\rho(0) = q_0$. Given a strategy σ of Player 1, a strategy τ of Player 2, and a state $q_0 \in Q$, there exists a unique play $\rho \in \mathrm{Out}^G(\sigma, q_0) \cap \mathrm{Out}^G(\tau, q_0)$, which we denote by $\rho^G(\sigma, \tau, q_0)$.

Traps and attractors. Intuitively, a subarena $T \subseteq Q$ of states is a *trap* for one of the two players if the other player can enforce that the play stays in this set. Formally, a trap for Player 2 (or simply a 2-trap) is a subarena $T \subseteq Q$ such that $qE \subseteq T$ for all states $q \in T \cap Q_2$, and $qE \cap T \neq \emptyset$ for all $q \in T \cap Q_1$. A trap for Player 1 (or 1-trap) is defined analogously. Note that if T is n trap for Player i in $G \restriction S$ and S is a trap for Player 1 in G, then T is also a trap for Player i in G.

If $T \subseteq Q$ is not a trap for Player 1, then Player 1 has a strategy to reach a position in $Q \setminus T$. In general, given a subset $S \subseteq Q$, we denote by $\mathrm{Attr}_1^G(S)$ the set of states from where Player 1 can force a visit to S. This set can be characterised as the limit of the sequence $(A_i)_{i \in \mathbb{N}}$ defined by $A^0 = S$ and

$$A^{i+1} = A^i \cup \{q \in Q_1 \mid qE \cap A^i \neq \emptyset\} \cup \{q \in Q_2 \mid qE \subseteq A^i\}\,.$$

From every state in $\mathrm{Attr}_1^G(S)$, Player 1 has a memoryless strategy σ that guarantees a visit to S in at most $|Q|$ steps: the strategy chooses for each state $q \in (A^i \setminus A^{i-1}) \cap Q_1$ a state $p \in qE \cap A^{i-1}$ (which decreases the distance to S by 1). We call the set $\mathrm{Attr}_1^G(S) = \bigcup_{i \in \mathbb{N}} A_i$ the 1-*attractor of* S and σ an *attractor strategy for* S. The 2-*attractor* of a set S, denoted by $\mathrm{Attr}_2^G(S)$, and attractor strategies for Player 2 are defined symmetrically. Notice that for any set S, the set $Q \setminus \mathrm{Attr}_1^G(S)$ is a 1-trap, and if S is a subarena (2-trap), then $\mathrm{Attr}_1^G(S)$ is also a subarena (2-trap). Analogously, $Q \setminus \mathrm{Attr}_2^G(S)$ is a 2-trap, and if S is a subarena (1-trap), then $\mathrm{Attr}_2^G(S)$ is also a subarena (1-trap).

Convention. We often drop the superscript G from the expressions defined above, if no confusion arises, e.g. by writing $\mathrm{Out}(\sigma, q_0)$ instead of $\mathrm{Out}^G(\sigma, q_0)$.

3 Mean-payoff parity games

In this first part of the paper, we show that mean-payoff parity games are determined, that both players have optimal strategies, that for Player 2 even memoryless strategies suffice, and that the value problem for mean-payoff parity games is in NP ∩ coNP. Furthermore, we present a deterministic algorithm which computes the values in time exponential in the number of priorities, and runs in pseudo-polynomial time when the number of priorities is bounded.

3.1 Definitions

Formally, a *mean-payoff parity game* is a tuple $\mathcal{G} = (G, \chi)$, where G is a weighted game graph, and $\chi \colon Q \to \mathbb{N}$ is a priority function assigning a *priority* to every state. A play $\rho = \rho(0)\rho(1)\cdots$ is *parity-winning* if the minimal priority occurring infinitely often in ρ is even, i.e., if $\min\{\chi(q) \mid q \in \mathrm{Inf}(\rho)\} \equiv 0 \pmod 2$. All notions that we have defined for weighted game graphs carry over to mean-payoff parity games. In particular, a play of $\mathcal{G}$ is just a play of G and a strategy for Player i in $\mathcal{G}$ is nothing but a strategy for Player i in G. Hence, we write $\mathrm{Out}^{\mathcal{G}}(\sigma, q)$ for $\mathrm{Out}^{G}(\sigma, q)$, and so on. As for weighted games graphs, we often omit the superscript if $\mathcal{G}$ is clear from the context. Finally, for a mean-payoff parity game $\mathcal{G} = (G, \chi)$ and a subarena S of G, we write $\mathcal{G} \restriction S$ for the mean-payoff parity game $(G \restriction S, \chi \restriction S)$.

We say that a mean-payoff parity game $\mathcal{G} = (G, \chi)$ is a *mean-payoff game* if $\chi(q)$ is even for all $q \in Q$. In particular, given a weighted game graph G, we obtain a mean-payoff game by assigning priority 0 to all states. We denote this game by $(G, 0)$.

If $\chi(Q) \subseteq \{0, 1\}$, then we say that $\mathcal{G}$ is a *mean-payoff Büchi game*; if $\chi(Q) \subseteq \{1, 2\}$, we call it a *mean-payoff co-Büchi game*. Hence, in a Büchi game Player 1 needs to visit the set $\chi^{-1}(0)$ infinitely often, whereas in a co-Büchi game he has to visit the set $\chi^{-1}(1)$ only finitely often.

For a play ρ of $\mathcal{G}$, we define its *payoff* as

$$\mathrm{payoff}^{\mathcal{G}}(\rho) = \begin{cases} \liminf\limits_{n \to \infty} \mathrm{payoff}_n^{\mathcal{G}}(\rho) & \text{if } \rho \text{ is parity-winning,} \\ -\infty & \text{otherwise,} \end{cases}$$

where for $n \in \mathbb{N}$

$$\mathrm{payoff}_n^{\mathcal{G}}(\rho) = \begin{cases} \dfrac{1}{n} \sum\limits_{i=0}^{n-1} \mathrm{weight}(\rho(i), \rho(i+1)) & \text{if } n > 0, \\ -\infty & \text{if } n = 0. \end{cases}$$

If σ is a strategy for Player 1 in $\mathcal{G}$, we define its *value* from $q_0 \in Q$ as

$$\mathrm{val}^{\mathcal{G}}(\sigma, q_0) = \inf_\tau \mathrm{payoff}^{\mathcal{G}}(\rho(\sigma, \tau, q_0)) = \inf\{\mathrm{payoff}^{\mathcal{G}}(\rho) \mid \rho \in \mathrm{Out}^{\mathcal{G}}(\sigma, q_0)\},$$

where τ ranges over all strategies of Player 2 in $\mathcal{G}$. Analogously, the value of a strategy τ for Player 2 from q_0 is defined as

$$\mathrm{val}^{\mathcal{G}}(\tau, q_0) = \sup_\sigma \mathrm{payoff}^{\mathcal{G}}(\rho(\sigma, \tau, q_0)) = \sup\{\mathrm{payoff}^{\mathcal{G}}(\rho) \mid \rho \in \mathrm{Out}^{\mathcal{G}}(\tau, q_0)\},$$

where σ ranges over all strategies of Player 1 in $\mathcal{G}$. The *lower* and *upper value* of a state $q_0 \in Q$ are defined by

$$\underline{\mathrm{val}}^{\mathcal{G}}(q_0) = \sup_\sigma \mathrm{val}^{\mathcal{G}}(\sigma, q_0) \qquad \text{and} \qquad \overline{\mathrm{val}}^{\mathcal{G}}(q_0) = \inf_\tau \mathrm{val}^{\mathcal{G}}(\tau, q_0),$$

respectively. Intuitively, $\underline{\mathrm{val}}^{\mathcal{G}}(q_0)$ and $\overline{\mathrm{val}}^{\mathcal{G}}(q_0)$ are the maximal (respectively minimal) payoff that Player 1 (respectively Player 2) can ensure (in the limit). We say that a strategy σ of Player 1 is *optimal from* q_0 if $\mathrm{val}^{\mathcal{G}}(\sigma, q_0) = \underline{\mathrm{val}}^{\mathcal{G}}(q_0)$. Analogously, we call a strategy τ of Player 2 optimal from q_0 if $\mathrm{val}^{\mathcal{G}}(\tau, q_0) =$

$\overline{\mathrm{val}}^{\mathcal{G}}(q_0)$. A strategy is *(globally) optimal* if it is optimal from every state $q \in Q$. It is easy to see that $\underline{\mathrm{val}}^{\mathcal{G}}(q_0) \leq \overline{\mathrm{val}}^{\mathcal{G}}(q_0)$. If $\underline{\mathrm{val}}^{\mathcal{G}}(q_0) = \overline{\mathrm{val}}^{\mathcal{G}}(q_0)$, we say that q_0 has a *value*, which we denote by $\mathrm{val}^{\mathcal{G}}(q_0) := \underline{\mathrm{val}}^{\mathcal{G}}(q_0) = \overline{\mathrm{val}}^{\mathcal{G}}(q_0)$.

In the next section, we will see that mean-payoff games are *determined*, i.e., that every state has a value. The *value problem* is the following decision problem: Given a mean-payoff parity game $\mathcal{G}$ (with integral weights), a designated state $q_0 \in Q$, and a number $x \in \mathbb{Q}$, decide whether $\mathrm{val}^{\mathcal{G}}(q_0) \geq x$.

Example 1. Consider the mean-payoff parity game $\mathcal{G}$ depicted in Fig. 1: all states belong to Player 1, the number next to a state indicates the priority of the respective state, and the number next to an edge indicates the weight of the respective edge. Note that $\mathrm{val}^{\mathcal{G}}(q_1) = 1$ since Player 1 can delay visiting q_2 longer and longer while still ensuring that this vertex is seen infinitely often. However, there is no finite-memory strategy that achieves this value. Let σ be a finite-memory strategy of Player 1 in $\mathcal{G}$, and let ρ be the unique play of $\mathcal{G}$ that starts in q_1 and is consistent with σ. Assume furthermore that ρ visits q_2 infinitely often (otherwise $\mathrm{val}^{\mathcal{G}}(\sigma, q_1) = -\infty$). Then $\rho = {q_1}^{k_1} q_2 {q_1}^{k_2} q_2 \cdots$, where each $k_i \in \mathbb{N} \setminus \{0\}$. Since σ is a finite-memory strategy, there exists $m \in \mathbb{N}$ such that $k_i \leq m$ for all $i \in \mathbb{N}$. Hence, $\mathrm{val}^{\mathcal{G}}(\sigma, q_q) = \mathrm{payoff}(\rho) \leq m/(m+1) < 1$.

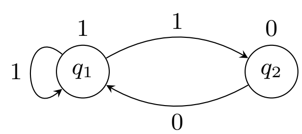

Fig. 1. A mean-payoff parity game for which infinite memory is necessary.

3.2 Strategy complexity

It follows from Martin's determinacy theorem [18] that mean-payoff parity games are determined. Moreover, Chatterjee et al. [8] gave an algorithmic proof for the existence of optimal strategies. Finally, it can be shown that for every $x \in \mathbb{R} \cup \{-\infty\}$ the set $\{\rho \in Q^\omega \mid \mathrm{payoff}(\rho) \geq x\}$ is closed under *combinations.* By Theorem 4 in [16], this property implies that Player 2 even has a memoryless optimal strategy.

We give here a self-contained proof of these facts that does not rely on Martin's theorem. We start by proving that Player 1 has an optimal strategy in games where Player 2 is absent.

Lemma 2. *Let $\mathcal{G}$ be a mean-payoff parity game with $Q_2 = \emptyset$. Then Player 1 has an optimal strategy in $\mathcal{G}$.*

Proof. It suffices to construct for each $q_0 \in Q$ a strategy σ with $\mathrm{val}^{\mathcal{G}}(\sigma, q_0) \geq \mathrm{val}^{\mathcal{G}}(q_0)$. If $\mathrm{val}^{\mathcal{G}}(q_0) = -\infty$, we can choose an arbitrary strategy σ. Otherwise, by the definition of $\mathrm{val}^{\mathcal{G}}(q_0)$, for each $\varepsilon > 0$ there exists a play $\rho_\varepsilon \in \mathrm{Out}^{\mathcal{G}}(q_0)$ with $\mathrm{payoff}(\rho_\varepsilon) \geq \mathrm{val}^{\mathcal{G}}(q_0) - \varepsilon$. Consider the sets $\mathrm{Inf}(\rho_\varepsilon)$ of states occurring infinitely often in ρ_ε. Since there are only finitely many such sets, we can find a set $P \subseteq Q$ such that for each $\varepsilon > 0$ there exists $0 < \varepsilon' < \varepsilon$ with $P = \mathrm{Inf}(\rho_{\varepsilon'})$. Let $q_{\min} \in P$ be a vertex of lowest priority. (This priority must be even since each ρ_ε fulfils the parity condition).

Let σ_1 be an optimal memoryless strategy in the mean-payoff game $\mathcal{G}_P = (G \upharpoonright P, 0)$ (the strategy σ_1 just leads the play to a simple cycle with maximum average weight), and let σ_2 be the memoryless attractor strategy in the game $\mathcal{G}_P$ that ensures a visit to $q_{\min}$ from all states $q \in P$; we extend both strategies to a strategy in $\mathcal{G}$ by combining them with a memoryless attractor strategy for P. (In particular, σ_2 enforces a visit to $q_{\min}$ from q_0.) Note that $\mathrm{val}^{\mathcal{G}_P}(q) \geq \mathrm{val}^{\mathcal{G}}(q_0)$ for all $q \in P$ since each of the plays $\rho_{\varepsilon'}$ visits each vertex in P and has payoff $\geq \mathrm{val}^{\mathcal{G}}(q_0) - \varepsilon'$.

Player 1's optimal strategy σ is played in rounds: in the ith round, Player 1 first forces a visit to $q_{\min}$ by playing according to σ_2; once $q_{\min}$ has been visited, Player 1 plays σ_1 for i steps before proceeding to the next round. Note that $\mathrm{val}^{\mathcal{G}_P}(\sigma, q_{\min}) = \mathrm{val}^{\mathcal{G}_P}(\sigma_1, q_{\min})$. Moreover, the unique play $\rho \in \mathrm{Out}^{\mathcal{G}}(\sigma, q_0)$ satisfies $q_{\min} \in \mathrm{Inf}(\rho) \subseteq P$ and therefore fulfils the parity condition. To sum up, we have $\mathrm{val}^{\mathcal{G}}(\sigma, q_0) = \mathrm{val}^{\mathcal{G}}(\sigma, q_{\min}) = \mathrm{val}^{\mathcal{G}_P}(\sigma, q_{\min}) = \mathrm{val}^{\mathcal{G}_P}(\sigma_1, q_{\min}) = \mathrm{val}^{\mathcal{G}_P}(q_{\min}) \geq \mathrm{val}^{\mathcal{G}}(q_0)$. ☐

Using Lemma 2, we can prove that mean-payoff-parity games are not only determined, but also that Player 1 has an optimal strategy and that Player 2 has a memoryless optimal strategy.

We use the *loop factorisation* technique (cf. [23]): Let γ be a play prefix and let $\hat{q} \in Q$. The *loop factorisation of γ relative to $\hat{q}$* is the unique factorisation of the form $\gamma = \gamma_0\gamma_1 \cdots \gamma_l$, where γ_0 does not contain $\hat{q}$, and each factor γ_i, $1 \leq i \leq l$, is of the form $\gamma_i = \hat{q} \cdot \gamma_i'$ where γ_i' does not contain $\hat{q}$. Analogously, for a play ρ which has infinitely many occurrences of $\hat{q}$ the *loop factorisation of ρ relative to $\hat{q}$* is the unique factorisation $\rho = \gamma_0\gamma_1 \cdots$ where each γ_i has the same properties as in the above case.

For a state $\hat{q}$ with m successors, $\hat{q}E = \{q_1, \ldots, q_m\}$, we define an operator $\pi_i \colon Q^* \to Q^*$ for each $1 \leq i \leq m$ by setting

$$\pi_i(\gamma) \coloneqq \begin{cases} \gamma & \text{if either } \gamma = \hat{q}q_i\gamma' \text{ for some } \gamma' \in Q^* \text{ or } \gamma = q_i = \hat{q}, \\ \varepsilon & \text{otherwise.} \end{cases}$$

The operator π_i induces another operator $\mathit{\Pi}_i \colon Q^* \to Q^*$ by setting

$$\mathit{\Pi}_i(\gamma) = \mathit{\Pi}_i(\gamma_0)\mathit{\Pi}_i(\gamma_1) \cdots \mathit{\Pi}_i(\gamma_l),$$

where $\gamma = \gamma_0\gamma_1 \cdots \gamma_l$ is the loop factorisation of γ relative to $\hat{q}$. The operator $\mathit{\Pi}_i$ operates on play prefixes, but it can easily be extended to operate on infinite plays with infinitely many occurrences of $\hat{q}$.

Theorem 3. *Let $\mathcal{G}$ be a mean-payoff parity game.*

1. *$\mathcal{G}$ is determined;*
2. *Player 1 has an optimal strategy in $\mathcal{G}$;*
3. *Player 2 has a memoryless optimal strategy in $\mathcal{G}$.*

Proof. We proceed by an induction over the size of $S \coloneqq \{q \in Q_2 \mid |qE| > 1\}$, the set of all Player 2 states with more than one successor. If $S = \emptyset$, all statements follow from Lemma 2. Let 1.–3. be fulfilled for all games with $|S| < n$ and let $\mathcal{G} = (G, \chi)$ be a mean-payoff parity game with $|S| = n$. We prove that the statements also hold for $\mathcal{G}$. Let $\hat{q} \in S$ with $\hat{q}E = \{q_1, \ldots, q_m\}$. For each $1 \leq j \leq m$, we define a new game $\mathcal{G}_j = (G_j, \chi)$ by setting $E_j = E \setminus (\{\hat{q}\} \times Q) \cup \{(\hat{q}, q_j)\}$, and $G_j = (Q_1, Q_2, E_j, \text{weight} \restriction E_j)$. Note that the induction hypothesis applies to each $\mathcal{G}_j$. W.l.o.g. assume that $\text{val}^{\mathcal{G}_1}(\hat{q}) \leq \text{val}^{\mathcal{G}_j}(\hat{q})$ for all $1 \leq j \leq m$. We will construct a memoryless strategy τ for Player 2 and a strategy σ for Player 1 such that $\text{val}^{\mathcal{G}}(\tau, q_0) \leq \text{val}^{\mathcal{G}_1}(q_0)$ and $\text{val}^{\mathcal{G}}(\sigma, q_0) \geq \text{val}^{\mathcal{G}_1}(q_0)$ for every $q_0 \in Q$. Hence,

$$\text{val}^{\mathcal{G}_1}(q_0) \leq \text{val}^{\mathcal{G}}(\sigma, q_0) \leq \underline{\text{val}}^{\mathcal{G}}(q_0) \leq \overline{\text{val}}^{\mathcal{G}}(q_0) \leq \text{val}^{\mathcal{G}}(\tau, q_0) \leq \text{val}^{\mathcal{G}_1}(q_0),$$

and all these numbers are equal. In particular, we have $\text{val}^{\mathcal{G}}(q_0) = \underline{\text{val}}^{\mathcal{G}}(q_0) = \overline{\text{val}}^{\mathcal{G}}(q_0)$, $\text{val}^{\mathcal{G}}(\sigma, q_0) = \text{val}^{\mathcal{G}}(q_0)$ and $\text{val}^{\mathcal{G}}(\tau, q_0) = \text{val}^{\mathcal{G}}(q_0)$, which proves 1.–3.

By the induction hypothesis, Player 2 has a memoryless optimal strategy τ in $\mathcal{G}_1$. Clearly, τ is also a memoryless strategy for Player 2 in $\mathcal{G}$, and $\text{val}^{\mathcal{G}}(\tau, q_0) = \text{val}^{\mathcal{G}_1}(\tau, q_0) = \text{val}^{\mathcal{G}_1}(q_0)$ for all $q_0 \in Q$.

It remains to construct a strategy σ for Player 1 in $\mathcal{G}$ such that $\text{val}^{\mathcal{G}}(\sigma, q_0) \geq \text{val}^{\mathcal{G}_1}(q_0)$ for all $q_0 \in Q$.

First, we devise a strategy $\hat{\sigma}$ such that $\text{val}^{\mathcal{G}}(\hat{\sigma}, \hat{q}) \geq \text{val}^{\mathcal{G}_1}(\hat{q})$. If $\text{val}^{\mathcal{G}_1}(\hat{q}) = -\infty$, we can take an arbitrary strategy. Hence, assume that $\text{val}^{\mathcal{G}_1}(\hat{q})$ is finite. By the induction hypothesis, for each $j = 1, \ldots, m$ there exists a strategy σ_j for Player 1 in $\mathcal{G}_j$ with $\text{val}^{\mathcal{G}_j}(\sigma_j, \hat{q}) = \text{val}^{\mathcal{G}_j}(\hat{q})$. We define $\hat{\sigma}$ to be the *interleaving strategy*, defined by

$$\hat{\sigma}(\gamma) = \hat{\sigma}(\gamma_0 \cdots \gamma_l) = \begin{cases} \sigma_1(\Pi_1(\gamma)) & \text{if } \gamma_l = \hat{q}q_1\gamma' \text{ for some } \gamma' \in Q^*, \\ \quad\vdots & \quad\quad\vdots \\ \sigma_m(\Pi_m(\gamma)) & \text{if } \gamma_l = \hat{q}q_m\gamma' \text{ for some } \gamma' \in Q^*, \end{cases}$$

for all play prefixes γ whose loop factorisation relative to $\hat{q}$ equals $\gamma_0 \cdots \gamma_l$. We claim that $\text{val}^{\mathcal{G}}(\hat{\sigma}, \hat{q}) \geq \text{val}^{\mathcal{G}_1}(\hat{q})$.

Let $\rho \in \text{Out}^{\mathcal{G}}(\hat{\sigma}, \hat{q})$. If ρ has only finitely many occurrences of $\hat{q}$, then ρ is equivalent to a play in $\mathcal{G}_j$ that is consistent with σ_j for some j. Since $\text{val}^{G_j}(\hat{q}) \geq \text{val}^{G_1}(\hat{q})$ and σ_j is optimal, $\text{payoff}(\rho) \geq \text{val}^{G_1}(\hat{q})$, and we are done. Otherwise, consider the loop factorisation $\rho = \gamma_0\gamma_1 \cdots$ and set

$$\Gamma = \{j \in \{1, \ldots, m\} \mid \gamma_i \cdot \hat{q} \text{ is a loop in } \mathcal{G}_j \text{ for infinitely many } i \in \mathbb{N}\}.$$

Since the mean-payoff parity condition is prefix-independent, we can assume w.l.o.g. that every loop in ρ is a loop in $\mathcal{G}_j$ for $j \in \Gamma$. For each $j \in \Gamma$, denote by $\rho_j = \Pi_j(\rho)$ the corresponding play in $\mathcal{G}_j$. By definition of $\hat{\sigma}$, we have $\rho_j \in \mathrm{Out}^{\mathcal{G}_j}(\sigma_j, \hat{q})$ for each $j \in \Gamma$. Since $\mathrm{val}^{\mathcal{G}_1}(\hat{q})$ is finite and $\mathrm{val}^{\mathcal{G}_1}(\hat{q}) \leq \mathrm{val}^{\mathcal{G}_j}(\hat{q})$, each ρ_j fulfils the parity condition. As the minimal priority occurring infinitely often in ρ also occurs infinitely often in one ρ_j, this implies that ρ fulfils the parity condition.

We claim that for each $n > 0$, $\mathrm{payoff}_n(\rho)$ is a weighted average of $\mathrm{payoff}_{n_j}(\rho_j)$ for some $n_j > 0$. To see this, consider the loop factorisation $\gamma'_0 \cdots \gamma'_k$ of $\rho[0, n]$. (Note that $\gamma'_i = \gamma_i$ for all $i < k$.) For each $j \in \Gamma$, set

$$n_j = \begin{cases} |\Pi_j(\rho[0,n])| - 1 & \text{if } \gamma'_k \text{ is a history of } \mathcal{G}_j \text{ and either } \gamma'_k \neq \hat{q} \text{ or } q_j = \hat{q}. \\ |\Pi_j(\rho[0,n])| & \text{otherwise.} \end{cases}$$

Intuitively, n_j is the number of transitions in $\rho[0, n]$ that correspond to a transition in ρ_j. Hence,

$$\{(\rho(i), \rho(i+1)) \mid 0 \leq i < n\} = \bigcup_{j \in \Gamma} \{(\rho_j(i), \rho_j(i+1)) \mid 0 \leq i < n_j\}.$$

In particular, $\sum_{j \in \Gamma} n_j = n$ and $\sum_{j \in \Gamma} n_j / n = 1$. We have

$$\begin{aligned} \mathrm{payoff}_n(\rho) &= \frac{1}{n} \sum_{i=0}^{n-1} \mathrm{weight}(\rho(i), \rho(i+1)) \\ &= \frac{1}{n} \sum_{\substack{j \in \Gamma \\ n_j > 0}} \sum_{i=0}^{n_j - 1} \mathrm{weight}(\rho_j(i), \rho_j(i+1)) \\ &= \sum_{\substack{j \in \Gamma \\ n_j > 0}} \frac{n_j}{n} \cdot \frac{1}{n_j} \sum_{i=0}^{n_j - 1} \mathrm{weight}(\rho_j(i), \rho_j(i+1)) \\ &= \sum_{\substack{j \in \Gamma \\ n_j > 0}} \frac{n_j}{n} \cdot \mathrm{payoff}_{n_j}(\rho_j). \end{aligned}$$

Since a weighted average is always bounded from below by the minimum element, we can conclude that

$$\mathrm{payoff}_n(\rho) \geq \min_{\substack{j \in \Gamma \\ n_j > 0}} \mathrm{payoff}_{n_j}(\rho_j) \geq \min_{j \in \Gamma} \mathrm{payoff}_{n_j}(\rho_j).$$

Taking the lower limit on both sides, we obtain

$$\begin{aligned} \mathrm{payoff}(\rho) &= \liminf_{n \to \infty} \mathrm{payoff}_n(\rho) \\ &\geq \liminf_{n \to \infty} \min_{j \in \Gamma} \mathrm{payoff}_{n_j}(\rho_j) \\ &= \min_{j \in \Gamma} \liminf_{n \to \infty} \mathrm{payoff}_{n_j}(\rho_j) \end{aligned}$$

$$= \min_{j \in \Gamma} \liminf_{n_j \to \infty} \text{payoff}_{n_j}(\rho_j)$$
$$= \min_{j \in \Gamma} \text{payoff}(\rho_j).$$

Since each ρ_j is consistent with σ_j and σ_j is optimal, we have $\text{payoff}(\rho_j) \geq \text{val}^{\mathcal{G}_j}(\hat{q}) \geq \text{val}^{\mathcal{G}_1}(\hat{q})$ for each $j \in \Gamma$ and therefore also $\text{payoff}(\rho) \geq \text{val}^{\mathcal{G}_1}(\hat{q})$. Since this holds for all $\rho \in \text{Out}^{\mathcal{G}}(\hat{\sigma}, \hat{q})$, we can conclude that $\text{val}^{\mathcal{G}}(\hat{\sigma}, \hat{q}) \geq \text{val}^{\mathcal{G}_1}(\hat{q})$.

Finally, we construct a strategy σ for Player 1 in $\mathcal{G}$ such that $\text{val}^{\mathcal{G}}(\sigma, q_0) \geq \text{val}^{\mathcal{G}_1}(q_0)$ for all $q_0 \in Q$. Let

$$\sigma(\gamma) = \begin{cases} \sigma_1(\gamma) & \text{if } \hat{q} \text{ does not occur in } \gamma, \\ \hat{\sigma}(\hat{q}\gamma_2) & \text{if } \gamma = \gamma_1 \hat{q} \gamma_2 \text{ with } \gamma_1 \in (Q \setminus \{\hat{q}\})^*. \end{cases}$$

Then for each play $\rho \in \text{Out}^{\mathcal{G}}(\sigma, q_0)$ where $\hat{q}$ does not occur, it holds $\text{payoff}^{\mathcal{G}}(\rho) = \text{payoff}^{\mathcal{G}_1}(\rho) \geq \text{val}^{\mathcal{G}_1}(\sigma_1, q_0) = \text{val}^{\mathcal{G}_1}(q_0)$. If $\hat{q}$ occurs in at least one play consistent with σ, then in the game $\mathcal{G}_1$ (where σ_1 is optimal), we have $\text{val}^{\mathcal{G}_1}(q_0) = \text{val}^{\mathcal{G}_1}(\sigma_1, q_0) \leq \text{val}^{\mathcal{G}_1}(\hat{q})$. Hence, for each play $\rho \in \text{Out}^{\mathcal{G}}(\sigma, q_0)$ where $\hat{q}$ occurs (say at position j), it holds $\text{payoff}^{\mathcal{G}}(\rho) = \text{payoff}^{\mathcal{G}}(\rho[j, \infty)) \geq \text{val}^{\mathcal{G}}(\hat{\sigma}, \hat{q}) \geq \text{val}^{\mathcal{G}_1}(\hat{q}) \geq \text{val}^{\mathcal{G}_1}(q_0)$. Altogether we have $\text{payoff}^{\mathcal{G}}(\rho) \geq \text{val}^{\mathcal{G}_1}(q_0)$ for every play $\rho \in \text{Out}^{\mathcal{G}}(\sigma, q_0)$ and therefore $\text{val}^{\mathcal{G}}(\sigma, q_0) \geq \text{val}^{\mathcal{G}_1}(q_0)$. ☐

A consequence of the proof of Lemma 2 and Theorem 3 is that each value of a mean-payoff parity game is either $-\infty$ or equals one of the values of a mean-payoff game played on the same weighted graph (or a subarena of it). Since optimal memoryless strategies exist in mean-payoff games [11], the values of a mean-payoff game with integral weights are rational numbers of the form r/s with $|r| \leq |Q| \cdot W$ and $|s| \leq |Q|$. Consequently, this property holds for the (finite) values of a mean-payoff parity game as well.

While Example 1 demonstrates that an optimal strategy of Player 1 requires infinite memory in general, this is not the case for mean-payoff co-Büchi games, where both players have memoryless optimal strategies. This can be seen by applying Theorem 2 of [13] or by an inductive proof, which we provide here.

Theorem 4. *Let $\mathcal{G}$ be a mean-payoff co-Büchi game. Then Player 1 has a memoryless optimal strategy from every state $q_0 \in Q$.*

Proof. The proof is by induction over the number $|Q| = n$ of states in $\mathcal{G}$. For $n = 1$, the statement is trivially fulfilled. Now let $n > 1$, $q_0 \in Q$, and assume that the statement is true for all games with less than n states. Define $Q' = Q \setminus \text{Attr}_2(\chi^{-1}(1))$. If $Q' = \emptyset$, then Player 2 can force visiting $\chi^{-1}(1)$ infinitely often by playing a memoryless attractor strategy. Hence, $\text{val}^{\mathcal{G}}(q_0) = -\infty$, and every memoryless strategy of Player 1 is optimal. In the following, assume that $Q' \neq \emptyset$. Consider the game $\mathcal{G}' \coloneqq \mathcal{G} \upharpoonright Q'$, which is a mean-payoff game, and set

$$S \coloneqq \{q \in Q' \mid \text{val}^{\mathcal{G}'}(q) \geq \text{val}^{\mathcal{G}}(q_0)\}.$$

Note that S is a trap for Player 2 both in $\mathcal{G}'$ and in $\mathcal{G}$ (since Q' is a 2-trap in $\mathcal{G}$). We claim that $S \neq \emptyset$. Towards a contradiction, assume that $S = \emptyset$, i.e.,

$\mathrm{val}^{\mathcal{G}'}(q) < \mathrm{val}^{\mathcal{G}}(q_0)$ for all $q \in Q'$, and let τ be an optimal memoryless strategy for Player 2 in $\mathcal{G}'$. We extend τ to a strategy in $\mathcal{G}$ by combining it with a memoryless attractor strategy for $\chi^{-1}(1)$ on $\mathrm{Attr}_2(\chi^{-1}(1))$. Let $\rho \in \mathrm{Out}^{\mathcal{G}}(\tau, q_0)$ and $m \coloneqq \max_{q \in Q'} \mathrm{val}^{\mathcal{G}'}(q)$. Either ρ visits $\mathrm{Attr}_2(\chi^{-1}(1))$ and therefore also $\chi^{-1}(1)$ infinitely often, in which case $\mathrm{payoff}(\rho) = -\infty < m$, or $\rho[i, \infty)$ is a play of $\mathcal{G}'$ for some $i \in \mathbb{N}$, in which case $\mathrm{payoff}(\rho) = \mathrm{payoff}(\rho[i, \infty)) \leq \mathrm{val}^{\mathcal{G}'}(\rho(i)) \leq m$. Hence, $\mathrm{val}^{\mathcal{G}}(q_0) \leq \mathrm{val}^{\mathcal{G}}(\tau, q_0) \leq m < \mathrm{val}^{\mathcal{G}}(q_0)$, a contradiction.

Now, let σ' be a memoryless optimal strategy of Player 1 in $\mathcal{G}'$. By the definition of S, we have $\mathrm{val}^{\mathcal{G}'}(\sigma', q) \geq \mathrm{val}^{\mathcal{G}}(q_0)$ for all $q \in S$. Moreover, σ' induces a memoryless strategy σ_S in $\mathcal{G} \restriction S$ such that $\mathrm{val}^{\mathcal{G} \restriction S}(\sigma_S, q) = \mathrm{val}^{\mathcal{G}'}(\sigma', q) \geq \mathrm{val}^{\mathcal{G}}(q_0)$ for all $q \in S$. Let $A = \mathrm{Attr}_1^{\mathcal{G}}(S)$. We extend σ_S to a memoryless strategy σ_A in $\mathcal{G} \restriction A$ by combining it with a memoryless attractor strategy for S on $A \setminus S$. It follows that $\mathrm{val}^{\mathcal{G} \restriction A}(\sigma_A, q) \geq \mathrm{val}^{\mathcal{G}}(q_0)$ for all $q \in \mathrm{Attr}_1(S)$. If $q_0 \in \mathrm{Attr}_1(S)$, we are done. Otherwise, $q_0 \in T \coloneqq Q \setminus A$. Since $S \neq \emptyset$, the game $\mathcal{G} \restriction T$ has less states than $\mathcal{G}$, and by the induction hypothesis, Player 1 has a memoryless optimal strategy σ_T from q_0 in $\mathcal{G} \restriction T$. Note that, since T is a trap for Player 1, we have $\mathrm{val}^{\mathcal{G} \restriction T}(\sigma_T, q_0) = \mathrm{val}^{\mathcal{G} \restriction T}(q_0) \geq \mathrm{val}^{\mathcal{G}}(q_0)$. Let σ be the union of σ_A and σ_T, which is a memoryless strategy in $\mathcal{G}$. We claim that σ is optimal from q_0 in $\mathcal{G}$. Let $\rho \in \mathrm{Out}^{\mathcal{G}}(\sigma, q_0)$. If ρ stays in T, it is consistent with σ_T and must have payoff at least $\mathrm{val}^{\mathcal{G} \restriction T}(\sigma_T, q_0) \geq \mathrm{val}^{\mathcal{G}}(q_0)$. Otherwise, there exists $i \in \mathbb{N}$ such that $\rho(i) \in A$ and $\rho[i, \infty)$ is consistent with σ_A, which implies $\mathrm{payoff}(\rho) = \mathrm{payoff}(\rho[i, \infty)) \geq \mathrm{val}^{\mathcal{G} \restriction A}(\sigma_A, \rho(i)) \geq \mathrm{val}^{\mathcal{G}}(q_0)$. ☐

3.3 Computational complexity

In this section, we prove that the value problem for mean-payoff parity games lies in NP ∩ coNP. Although this has already been proved by Chatterjee and Doyen [6], our proof has the advantage that it works immediately on mean-payoff parity games, and not on energy parity games as in [6].

In order to put the value problem for mean-payoff parity games into coNP, we first show that the value can be decided in polynomial time in games where Player 2 is absent.

Proposition 5. *The problem of deciding, given a mean-payoff parity game $\mathcal{G}$ with $Q_2 = \emptyset$, a state $q_0 \in Q$, and $x \in \mathbb{Q}$, whether $\mathrm{val}^{\mathcal{G}}(q_0) \geq x$, is in P.*

Proof. Deciding whether $\mathrm{val}^{\mathcal{G}}(q_0) \geq x$ is achieved by Algorithm 1, which employs as subroutines Tarjan's linear-time algorithm [10] for SCC decomposition and Karp's polynomial-time algorithm [15] for computing the minimum/maximum cycle weight, (i.e. the minimum/maximum average weight on a cycle) in a given strongly connected graph.

The algorithm is sound: If the algorithm accepts, then there is an even priority p and a reachable SCC C in G_p with $p \in \chi(C)$ that has maximum cycle weight $w \geq x$. We construct a strategy σ for Player 1 with $\mathrm{val}^{\mathcal{G}}(\sigma, q_0) = w$. Let $q \in C$ be a state with priority p. Since q is reachable from q_0 and C is strongly connected, both q_0 and C lie inside $\mathrm{Attr}_1(\{q\})$. Let σ_q be the memoryless

Algorithm 1. A polynomial-time algorithm for deciding the value of a state in a one-player mean-payoff parity game.

Input: mean-payoff parity game $\mathcal{G}$ with $Q_2 = \emptyset$, $q_0 \in Q$, $x \in \mathbb{Q}$.
Output: whether $\mathrm{val}^{\mathcal{G}}(q_0) \geq x$.

$G' = G \upharpoonright \{q \in Q \mid q \text{ is reachable from } q_0\}$
for each even $p \in \chi(Q)$ **do**
  $G_p = G' \upharpoonright \{q \in Q \mid \chi(q) \geq p\}$
  decompose G_p into SCCs
  for each SCC C of G_p with $p \in \chi(C)$ **do**
    compute maximum cycle weight w in C
    if $w \geq x$ **then accept**
  done
done
reject

attractor strategy for $\{q\}$. Now, since w is the maximum cycle weight in C, there exists a simple cycle $\gamma = q_1 \cdots q_n q_1$ in C with cycle weight w. We construct a (memoryless) strategy σ_γ on C by setting $\sigma_\gamma(q_n) = q_1$ and $\sigma_\gamma(q_i) = q_{i+1}$ for every $1 \leq i < n$; this strategy is extended to the whole game by combining it with an attractor strategy for $\{q_1, \dots, q_n\}$. The strategies σ_q and σ_γ are then combined to a strategy σ, which is played in rounds: in the ith round, Player 1 first forces a visit to $\chi^{-1}(p) \cap C$ by playing according to σ_q; once $\chi^{-1}(p) \cap C$ has been reached, Player 1 plays σ_γ for i steps before proceeding to the next round. Note that σ fulfils the parity condition because q is visited infinitely often and all other priorities that appear infinitely often obey $\chi(q) \geq p$. Finally, the payoff of $\rho(\sigma, q_0)$ equals the cycle weight of γ, i.e., $\mathrm{val}^{\mathcal{G}}(q_0) \geq \mathrm{val}^{\mathcal{G}}(\sigma, q_0) = w \geq x$.

The algorithm is complete: Assume that $\mathrm{val}^{\mathcal{G}}(q_0) = v \geq x$ and let $\rho \in \mathrm{Out}^{\mathcal{G}}(q_0)$ be a play with $\mathrm{payoff}^{\mathcal{G}}(\rho) = v$; such a play exists due to Lemma 2. Consider the set $\mathrm{Inf}(\rho)$ and let $p = \min \chi(\mathrm{Inf}(\rho))$ (which is even since $\mathrm{payoff}(\rho)$ is finite). Since $\mathrm{Inf}(\rho)$ is strongly connected, $\mathrm{Inf}(\rho) \subseteq C$ for an SCC C of G_p with $p \in \chi(C)$. Since optimal memoryless strategies exist in mean-payoff games, there exists a simple cycle with average weight $\geq v$ in C. Hence the algorithm accepts.

Since SCC decomposition and maximum cycle weight computation both take polynomial time, the whole algorithm runs in polynomial time. □

It follows from Theorem 3 and Proposition 5 that the value problem for mean-payoff parity games is in coNP: to decide whether $\mathrm{val}^{\mathcal{G}}(q_0) < x$, a nondeterministic algorithm can guess a memoryless strategy τ for Player 2 and check whether $\mathrm{val}^{\mathcal{G}}(\tau, q_0) < x$ in polynomial time.

Corollary 6. *The value problem for mean-payoff parity games is in coNP.*

Following ideas from [6], we prove that the value problem is not only in coNP, but also in NP. The core of Algorithm 2 is the procedure Check that on input S checks whether the value of all states in the game $\mathcal{G} \upharpoonright S$ is at least x. If the

Algorithm 2. A nondeterministic algorithm for deciding the value of a state in a mean-payoff parity game.

Input: mean-payoff parity game $\mathcal{G}$, state $q_0 \in Q$, $x \in \mathbb{Q}$

guess 2-trap T in $\mathcal{G}$ with $q_0 \in T$
Check(T)
accept

procedure Check(S)
  if $S \neq \emptyset$ **then**
    $p := \min\{\chi(q) \mid q \in S\}$
    if p is even **then**
      guess memoryless strategy σ_{M} for Player 1 in $G \upharpoonright S$
      if $\mathrm{val}^{(G\upharpoonright S,0)}(\sigma_{\mathrm{M}}, q) < x$ for some $q \in S$ **then reject**
      Check($S \setminus \mathrm{Attr}_1^{\mathcal{G}\upharpoonright S}(\chi^{-1}(p))$)
    else
      guess 2-trap $T \neq \emptyset$ in $\mathcal{G} \upharpoonright (S \setminus \mathrm{Attr}_2^{\mathcal{G}\upharpoonright S}(\chi^{-1}(p)))$
      Check(T); Check($S \setminus \mathrm{Attr}_1^{\mathcal{G}\upharpoonright S}(T)$)
    end if
  end if
end procedure

least priority p in S is even, this is witnessed by a strategy in the mean-payoff game $(G \upharpoonright S, 0)$ that ensures payoff $\geq x$ and the fact that the values of all states in the game $\mathcal{G} \upharpoonright S \setminus \mathrm{Attr}_1^{\mathcal{G}\upharpoonright S}(\chi^{-1}(p))$ are greater than x, which we can check by calling Check recursively. If, on the other hand, the least priority p in S is odd, then $\mathrm{val}^{\mathcal{G}\upharpoonright S}(q) \geq x$ for all $q \in S$ is witnessed by a 2-trap T inside $S \setminus \mathrm{Attr}_2^{\mathcal{G}\upharpoonright S}(\chi^{-1}(p))$ such that both the values in the game $\mathcal{G} \upharpoonright T$ and the values in the game $\mathcal{G} \upharpoonright S \setminus \mathrm{Attr}_1^{\mathcal{G}\upharpoonright S}(T)$ are bounded from below by x; the latter two properties can again be checked by calling Check recursively. The correctness of the algorithm relies on the following two lemmas.

Lemma 7. *Let $\mathcal{G}$ be a mean-payoff parity game with least priority p even, $T = Q \setminus \mathrm{Attr}_1(\chi^{-1}(p))$, and $x \in \mathbb{R}$. If $\mathrm{val}^{(G,0)}(q) \geq x$ for all $q \in Q$ and $\mathrm{val}^{\mathcal{G}\upharpoonright T}(q) \geq x$ for all $q \in T$, then $\mathrm{val}^{\mathcal{G}}(q) \geq x$ for all $q \in Q$.*

Proof. Assume that $\mathrm{val}^{(G,0)}(q) \geq x$ for all $q \in Q$ and $\mathrm{val}^{\mathcal{G}\upharpoonright T}(q) \geq x$ for all $q \in T$, and let $q^* \in Q$. By Theorem 3, it suffices to show that for every memoryless strategy τ of Player 2 there exists a strategy σ of Player 1 such that $\mathrm{payoff}(\rho(\sigma, \tau, q^*)) \geq x$. Hence, assume that τ is a memoryless strategy of Player 2 in $\mathcal{G}$. Moreover, let σ_{M} be a memoryless strategy for Player 1 in $(G, 0)$ with $\mathrm{val}^{(G,0)}(\sigma_{\mathrm{M}}, q) \geq x$ for all $q \in Q$, let σ_T be a strategy for Player 1 in $\mathcal{G} \upharpoonright T$ with $\mathrm{val}^{\mathcal{G}\upharpoonright T}(\sigma_T, q) \geq x$ for all $q \in T$, and let σ_{A} be a memoryless attractor strategy of Player 1 on $\mathrm{Attr}_1(\chi^{-1}(p))$ that ensures to reach $\chi^{-1}(p)$. We combine these three strategies to a new strategy σ, which is played in rounds. In the kth round, the strategy behaves as follows:

1. while the play stays inside T, play σ_T;
2. as soon as the play reaches $\mathrm{Attr}_1(\chi^{-1}(p))$, switch to strategy σ_{A} and play σ_{A} until the play reaches $\chi^{-1}(p)$;
3. when the play reaches $\chi^{-1}(p)$, play σ_M for exactly k steps and proceed to the next round.

Let $\rho \coloneqq \rho(\sigma, \tau, q^*)$. To complete the proof, we need to show that $\mathrm{payoff}(\rho) \geq x$. We distinguish whether ρ visits $\mathrm{Attr}_1(\chi^{-1}(p))$ infinitely often or not.

In the first case, we divide ρ into $\rho = \gamma_0\gamma_1\gamma_2\cdots$ where each $\gamma_i = \gamma_i^T\gamma_i^{\mathrm{A}}\gamma_i^{\mathrm{M}}$ consists of a part consistent with σ_T (thus staying inside T), a part consistent with σ_{A} (thus staying in $\mathrm{Attr}_1(\chi^{-1}(p))$), and one that starts with a state in $\chi^{-1}(p)$ and is consistent with σ_{M}. Since τ is a memoryless strategy, there can only be $|T|$ many different γ_i^T, and the length of each γ_i^T is bounded by some constant k. Since each γ_i^{A} is consistent with an attractor strategy, the length of each γ_i^{A} is bounded by $|Q|$. Hence, the length of γ_i^{M} grows continuously while the length of $\gamma_i^T\gamma_i^{\mathrm{A}}$ is bounded. Therefore, $\liminf_{n\to\infty} \mathrm{payoff}_n(\rho) = \liminf_{n\to\infty} \mathrm{payoff}_n(\gamma_1^{\mathrm{M}}\gamma_2^{\mathrm{M}}\cdots)$. Since $\mathrm{val}^{(G,0)}(\sigma_{\mathrm{M}}, q) \geq x$ for all $q \in Q$ and priority p is visited infinitely often, we have $\mathrm{payoff}(\rho) = \liminf_{n\to\infty} \mathrm{payoff}_n(\rho) \geq x$.

In the second case, $\rho = \gamma \cdot \rho'$, where ρ' is a play of $\mathcal{G} \restriction T$ that is consistent with σ_T. Hence, $\mathrm{payoff}(\rho) = \mathrm{payoff}(\rho') \geq \mathrm{val}^{\mathcal{G}\restriction T}(\sigma_T, \rho'(0)) \geq x$. ◻

Lemma 8. *Let $\mathcal{G}$ be a mean-payoff parity game with least priority p odd, $T = Q \setminus \mathrm{Attr}_2(\chi^{-1}(p))$, and $x \in \mathbb{R}$. If $\mathrm{val}^{\mathcal{G}}(q) \geq x$ for some $q \in Q$, then $T \neq \emptyset$ and $\mathrm{val}^{\mathcal{G}\restriction T}(q) \geq x$ for some $q \in T$.*

Proof. Let $q^* \in Q$ be a state with $\mathrm{val}^{\mathcal{G}}(q^*) \geq 0$. If $T = \emptyset$, then $\mathrm{Attr}_2(\chi^{-1}(p)) = Q$ and there is a memoryless attractor strategy τ for Player 2 in $\mathcal{G}$ that ensures to visit $\chi^{-1}(p)$ infinitely often. This implies $\mathrm{val}^{\mathcal{G}}(\tau, q^*) = -\infty$, a contradiction to $\mathrm{val}^{\mathcal{G}}(q^*) \geq x$. Thus $T \neq \emptyset$.

Now assume that $\mathrm{val}^{\mathcal{G}\restriction T}(q) < x$ for all $q \in T$, and let τ be a (w.l.o.g. memoryless) strategy for Player 2 in $\mathcal{G} \restriction T$ that ensures $\mathrm{val}^{\mathcal{G}\restriction T}(\tau, q) < x$ for all $q \in T$. We extend τ to a strategy τ' in $\mathcal{G}$ by combining it with a memoryless attractor strategy for $\chi^{-1}(p)$ on the states in $Q \setminus T$. Let $\rho \in \mathrm{Out}^{\mathcal{G}}(\tau', q^*)$. Either ρ reaches $\chi^{-1}(p)$ infinitely often, in which case $\mathrm{payoff}^{\mathcal{G}}(\rho) = -\infty$, or there is a position i from which onwards ρ stays in T, in which case $\mathrm{payoff}^{\mathcal{G}}(\rho) = \mathrm{payoff}^{\mathcal{G}\restriction T}(\rho[i, \infty)) \leq \mathrm{val}^{\mathcal{G}\restriction T}(\tau, \rho(i))$. In any case, $\mathrm{val}^{\mathcal{G}}(\tau', q^*) \leq \max_{q\in T} \mathrm{val}^{\mathcal{G}\restriction T}(\tau, q) < x$, a contradiction to $\mathrm{val}^{\mathcal{G}}(q^*) \geq x$. ◻

Finally, Algorithm 2 runs in polynomial time because the value of a memoryless strategy in a mean-payoff game can be computed in polynomial time [24] and because recursive calls are limited to disjoint subarenas.

Theorem 9. *The value problem for mean-payoff parity games is in NP.*

Proof. We claim that Algorithm 2 is a nondeterministic polynomial-time algorithm for the value problem. To analyse the running time, denote by $T(n)$ the worst-case running time of the procedure Check on a subarena S of size n. Since the value of a memoryless strategy for Player 1 in a mean-payoff game can be computed in polynomial time [24] and attractor computations take linear time, there exists a polynomial $f\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that the numbers $T(n)$ satisfy the following recurrence:

$$T(1) \leq f(\|G\|, \|x\|),$$
$$T(n) \leq \max_{1 \leq k < n} T(k) + T(n-k) + f(\|G\|, \|x\|)\,.$$

Solving this recurrence, we get that $T(n) \leq (2n-1) f(\|G\|, \|x\|)$ for all $n \geq 1$, again a polynomial. Consequently, the algorithm runs in polynomial time.

To prove the correctness of the algorithm, we need to prove that the algorithm is both sound and complete. We start by proving soundness: If the algorithm accepts its input, then $\mathrm{val}^{\mathcal{G}}(q_0) \geq x$. In fact, we prove the following stronger statement. We say that $\mathrm{Check}(S)$ *succeeds* if the procedure terminates without rejection (for at least one sequence of guesses).

Claim. Let $S \subseteq Q$. If S is a subarena of $\mathcal{G}$ and $\mathrm{Check}(S)$ does succeed, then $\mathrm{val}^{\mathcal{G}\restriction S}(q) \geq x$ for all $q \in S$.

Assume that the claim is true and that the algorithm accepts its input. Then there exists a 2-trap T with $q_0 \in T$ such that $\mathrm{val}^{\mathcal{G}\restriction T}(q) \geq x$ for all $q \in T$. Since T is a 2-trap, it follows that $\mathrm{val}^{\mathcal{G}}(q_0) \geq x$.

To prove the claim, we proceed by induction over the cardinality of S. If $|S| = 0$, the claim is trivially fulfilled. Hence, assume that $|S| > 0$ and that the claim is true for all sets $S' \subseteq Q$ with $|S'| < |S|$. Let $p = \min\{\chi(q) \mid q \in S\}$. We distinguish two cases:

1. The minimal priority p is even. Since $\mathrm{Check}(S)$ succeeds, there exists a memoryless strategy σ_{M} of Player 1 in $\mathcal{G} \restriction S$ such that $\mathrm{val}^{(G\restriction S,0)}(\sigma_{\mathrm{M}}, q) \geq x$ for all $q \in S$, i.e. $\mathrm{val}^{(G\restriction S,0)}(q) \geq x$ for all $q \in S$. Let $A = \mathrm{Attr}_1^{\mathcal{G}\restriction S}(\chi^{-1}(p))$. Since $\mathrm{Check}(S)$ succeeds, so does $\mathrm{Check}(S \setminus A)$. Hence, by the induction hypothesis, $\mathrm{val}^{\mathcal{G}\restriction(S\setminus A)}(q) \geq x$ for all $q \in S \setminus A$. By Lemma 7, these two facts imply that $\mathrm{val}^{\mathcal{G}\restriction S}(q) \geq x$ for all $q \in S$.
2. The minimal priority p is odd. Since $\mathrm{Check}(S)$ succeeds, there exists a 2-trap $T \neq \emptyset$ in $\mathcal{G} \restriction (S \setminus \mathrm{Attr}_2^{\mathcal{G}\restriction S}(\chi^{-1}(p)))$ such that both $\mathrm{Check}(T)$ and $\mathrm{Check}(S \setminus \mathrm{Attr}_1^{\mathcal{G}\restriction S}(T))$ succeed. Let $A = \mathrm{Attr}_1^{\mathcal{G}\restriction S}(T))$. By the induction hypothesis, Player 1 has a strategy σ_T in $\mathcal{G} \restriction T$ such that $\mathrm{val}^{\mathcal{G}\restriction T}(\sigma_T, q) \geq x$ for all $q \in T$ and a strategy σ_S in $\mathcal{G} \restriction S \setminus A$ such that $\mathrm{val}^{\mathcal{G}\restriction S\setminus A}(\sigma_S, q) \geq x$ for all $q \in S \setminus A$. We extend σ_T to a strategy σ_A in $\mathcal{G} \restriction A$ such that $\mathrm{val}^{\mathcal{G}\restriction A}(\sigma_A, q) \geq x$ for all $q \in A$ by combining σ_T with a suitable attractor strategy. By playing σ_S as long as the play stays in $S \setminus A$ and switching to σ_A as soon as the play enters A, Player 1 can ensure that $\mathrm{val}^{\mathcal{G}\restriction S}(q) \geq x$ for all $q \in S$.

Finally, we prove that the algorithm is complete: if $\mathrm{val}^{\mathcal{G}}(q_0) \geq x$, then the algorithm accepts the input $\mathcal{G}, q_0, x$. Since the set $\{q \in Q \mid \mathrm{val}^{\mathcal{G}}(q) \geq x\}$ is a trap for Player 2, it suffices to prove the following claim.

Claim. Let $S \subseteq Q$. If S is a subarena of $\mathcal{G}$ and $\mathrm{val}^{\mathcal{G}\restriction S}(q) \geq x$ for all $q \in S$, then $\mathrm{Check}(S)$ succeeds.

As the previous claim, we prove this claim by an induction over the cardinality of S. Clearly, $\mathrm{Check}(S)$ succeeds if $|S| = 0$. Hence, assume that $|S| > 0$ and that the claim is correct for all sets $S' \subseteq Q$ with $|S'| < |S|$. Moreover, assume that S is a subarena of $\mathcal{G}$ such that $\mathrm{val}^{\mathcal{G}\restriction S}(q) \geq x$ for all $q \in S$ (otherwise the claim is trivially fulfilled). Again, we distinguish whether $p \coloneqq \min\{\chi(q) \mid q \in S\}$ is even or odd.

1. The minimal priority p is even. Since $\mathrm{val}^{\mathcal{G}\restriction S}(q) \geq x$ for all $q \in S$, also $\mathrm{val}^{(G\restriction S,0)}(q) \geq x$ for all $q \in S$, which is witnessed by a memoryless strategy σ_{M}. Let $A = \mathrm{Attr}_1^{\mathcal{G}\restriction S}(\chi^{-1}(p))$. Since $S \setminus A$ is a 1-trap and $\mathrm{val}^{\mathcal{G}\restriction S}(q) \geq x$ for all $q \in S$, we must also have $\mathrm{val}^{\mathcal{G}\restriction (S\setminus A)}(q) \geq x$ for all $q \in S \setminus A$. Hence, by the induction hypothesis, $\mathrm{Check}(S \setminus A)$ succeeds. Therefore, in order to succeed, $\mathrm{Check}(S)$ only needs to guess a suitable memoryless strategy σ_{M}.
2. The minimal priority p is odd. Let $A \coloneqq \mathrm{Attr}_2^{\mathcal{G}\restriction S}(\chi^{-1}(p))$. We claim that $\mathrm{Check}(S)$ succeeds if it guesses $T \coloneqq \{q \in S \setminus A \mid \mathrm{val}^{\mathcal{G}\restriction (S\setminus A)}(q) \geq x\}$. By Lemma 8, the set T is nonempty. Note that T is a 2-trap and that $\mathrm{val}^{\mathcal{G}\restriction T}(q) \geq x$ for all $q \in T$. Hence, by the induction hypothesis, $\mathrm{Check}(T)$ succeeds. It remains to be shown that $\mathrm{Check}(S \setminus \mathrm{Attr}_1^{\mathcal{G}\restriction S}(T))$ succeeds as well. Note that $S \setminus \mathrm{Attr}_1^{\mathcal{G}\restriction S}(T)$ is a 1-trap, which together with $\mathrm{val}^{\mathcal{G}\restriction S}(q) \geq x$ for all $q \in S$ implies that $\mathrm{val}^{\mathcal{G}\restriction (S\setminus \mathrm{Attr}_1^{\mathcal{G}\restriction S}(T))}(q) \geq x$ for all $q \in S \setminus \mathrm{Attr}_1^{\mathcal{G}\restriction S}(T)$. Hence, the induction hypothesis yields that $\mathrm{Check}(S \setminus \mathrm{Attr}_1^{\mathcal{G}\restriction S}(T))$ succeeds. □

3.4 A deterministic algorithm

In this section, we present a deterministic algorithm for computing the values of a mean-payoff parity game, which runs faster than all known algorithms for solving these games. Algorithm SolveMPP is based on the classical algorithm for solving parity games, due to Zielonka [22]. The algorithm employs as a subprocedure an algorithm SolveMP for solving mean-payoff games. By [24], such an algorithm can be implemented to run in time $\mathrm{O}(n^3 \cdot m \cdot W)$ for a game with n states and m edges. We denote by $f \sqcup g$ and $f \sqcap g$ the pointwise maximum, respectively minimum, of two (partial) functions $f, g\colon Q \to \mathbb{R} \cup \{\pm\infty\}$ (where $(f \sqcup g)(q) = (f \sqcap g)(q) = f(q)$ if $g(q)$ is undefined).

The algorithm works as follows: If the least priority p in $\mathcal{G}$ is even, the algorithm first identifies the least value of $\mathcal{G}$ by computing the values of the mean-payoff game $(G, 0)$ and (recursively) the values of the game $\mathcal{G} \restriction Q \setminus \mathrm{Attr}_1(\chi^{-1}(p))$, and taking their minimum x. All states from where Player 2 can enforce a visit to a state with value x in one of these two games must have value x in $\mathcal{G}$. In the remaining subarena, the values can be computed by calling SolveMPP recursively. If the least priority is odd, we can similarly compute the greatest value of $\mathcal{G}$ and proceed by recursion.

Theorem 10. *The values of a mean-payoff parity game with d priorities can be computed in time $\mathrm{O}(|Q|^{d+2} \cdot |E| \cdot W)$.*

Algorithm SolveMPP($\mathcal{G}$)

Input: mean-payoff parity game $\mathcal{G} = (G, \chi)$
Output: $\mathrm{val}^{\mathcal{G}}$

if $Q = \emptyset$ **then return** $\emptyset$
$p \coloneqq \min\{\chi(q) \mid q \in Q\}$
if p is even **then**
  $g \coloneqq \mathrm{SolveMP}(G, 0)$
  if $\chi(q) = p$ for all $q \in Q$ **then return** g
  $T \coloneqq Q \setminus \mathrm{Attr}_1^{\mathcal{G}}(\chi^{-1}(p))$; $f \coloneqq \mathrm{SolveMPP}(\mathcal{G} \restriction T)$
  $x \coloneqq \min(f(T) \cup g(Q))$; $A \coloneqq \mathrm{Attr}_2^{\mathcal{G}}(f^{-1}(x) \cup g^{-1}(x))$
  return $(Q \to \mathbb{R} \cup \{-\infty\} \colon q \mapsto x) \sqcup \mathrm{SolveMPP}(\mathcal{G} \restriction Q \setminus A)$
else
  $T \coloneqq Q \setminus \mathrm{Attr}_2^{\mathcal{G}}(\chi^{-1}(p))$
  if $T = \emptyset$ **then return** $(Q \to \mathbb{R} \cup \{-\infty\} \colon q \mapsto -\infty)$
  $f \coloneqq \mathrm{SolveMPP}(\mathcal{G} \restriction T)$; $x \coloneqq \max f(T)$; $A \coloneqq \mathrm{Attr}_1^{\mathcal{G}}(f^{-1}(x))$
  return $(Q \to \mathbb{R} \cup \{-\infty\} \colon q \mapsto x) \sqcap \mathrm{SolveMPP}(\mathcal{G} \restriction Q \setminus A)$
end if

Proof. We claim that SolveMPP computes, given a mean-payoff parity game $\mathcal{G}$, the function $\mathrm{val}^{\mathcal{G}}$ in the given time bound. Denote by $T(n, m, d)$ the worst-case running time of the algorithm on a game with n states, m edges and d priorities. Note that, if $\mathcal{G}$ has only one priority, then there are no recursive calls to SolveMPP. Since attractors can be computed in time $\mathrm{O}(n + m)$ and the running time of SolveMP is $\mathrm{O}(n^3 \cdot m \cdot W)$, there exists a constant c such that the numbers $T(n, m, d)$ satisfy the following recurrence:

$$
\begin{aligned}
&T(1, m, d) \leq c, \\
&T(n, m, 1) \leq c \cdot n^3 \cdot m \cdot W, \\
&T(n, m, d) \leq T(n-1, m, d-1) + T(n-1, m, d) + c \cdot n^3 \cdot m \cdot W\,.
\end{aligned}
$$

We claim that $T(n, m, d) \leq c \cdot (n+1)^{d+2} \cdot m \cdot W \in \mathrm{O}(n^{d+2} \cdot m \cdot W)$. The claim is clearly true if $n = 1$. Hence, assume that $n \geq 2$ and that the claim is true for all lower values of n. If $d = 1$, the claim follows from the second inequality. Otherwise,

$$
\begin{aligned}
T(n, m, d) &\leq T(n-1, m, d-1) + T(n-1, m, d) + c \cdot n^3 \cdot m \cdot W \\
&\leq c \cdot n^{d+1} \cdot m \cdot W + c \cdot n^{d+2} \cdot m \cdot W + c \cdot n^3 \cdot m \cdot W \\
&\leq c \cdot (n^{d+1} + n \cdot n^{d+1} + n^{d+1}) \cdot m \cdot W \\
&\leq c \cdot ((n+1)^{d+1} + n \cdot (n+1)^{d+1}) \cdot m \cdot W \\
&= c \cdot (n+1)^{d+2} \cdot m \cdot W
\end{aligned}
$$

It remains to be proved that the algorithm is correct, i.e. that $\mathrm{SolveMPP}(\mathcal{G}) = \mathrm{val}^{\mathcal{G}}$. We prove the claim by induction over the number of states. If there are no states, the claim is trivial. Hence, assume that $Q \neq \emptyset$ and that the claim is true for all games with less than $|Q|$ states. Let $p \coloneqq \min\{\chi(q) \mid q \in Q\}$. We only consider the case that p is even. If p is odd, the proof is similar, but relies on Lemma 8 instead of Lemma 7.

Let T, f, g, x and A be defined as in the corresponding case of the algorithm, and let $f^* = \text{SolveMPP}(\mathcal{G})$. If $\chi(Q) = \{p\}$, then $f^* = g = \text{val}^{(G,0)} = \text{val}^{\mathcal{G}}$, and the claim is fulfilled. Otherwise, by the definition of x and applying the induction hypothesis to the game $\mathcal{G} \restriction T$, we have $\text{val}^{(G,0)}(q) \geq x$ for all $q \in Q$ and $\text{val}^{\mathcal{G} \restriction T}(q) = f(q) \geq x$ for all $q \in T$. Hence, Lemma 7 yields that $\text{val}^{\mathcal{G}}(q) \geq x$ for all $q \in Q$. On the other hand, from any state $q \in A$ Player 2 can play an attractor strategy to $f^{-1}(x) \cup g^{-1}(x)$, followed by an optimal strategy in the game $\mathcal{G} \restriction T$, respectively in the mean-payoff game $(G, 0)$, which ensures that Player 1's payoff does not exceed x. Hence, $\text{val}^{\mathcal{G}}(q) = x = f^*(q)$ for all $q \in A$.

Now, let $q \in Q \setminus A$. We already know that $\text{val}^{\mathcal{G}}(q) \geq x$. Moreover, since $Q \setminus A$ is a 2-trap and applying the induction hypothesis to the game $\mathcal{G} \restriction Q \setminus A$, we have $\text{val}^{\mathcal{G}}(q) \geq \text{val}^{\mathcal{G} \restriction Q \setminus A}(q) = \text{SolveMPP}(\mathcal{G} \restriction Q \setminus A)(q)$. Hence, $\text{val}^{\mathcal{G}}(q) \geq f^*(q)$. To see that $\text{val}^{\mathcal{G}}(q) \leq f^*(q)$, consider the strategy τ of Player 2 that mimics an optimal strategy in $\mathcal{G} \restriction Q \setminus A$ as long as the play stays in $Q \setminus A$ and switches to an optimal strategy in $\mathcal{G}$ as soon as the play reaches A. We have $\text{val}^{\mathcal{G}}(\tau, q) \leq \max\{\text{val}^{\mathcal{G} \restriction Q \setminus A}(q), x\} = f^*(q)$. ☐

Algorithm SolveMPP is faster and conceptually simpler than the original algorithm proposed for solving mean-payoff parity games [8]. Compared to the recent algorithm proposed by Chatterjee and Doyen [6], which uses a reduction to energy parity games and runs in time $\mathrm{O}(|Q|^{d+4} \cdot |E| \cdot d \cdot W)$, our algorithm has three main advantages: 1. it is faster; 2. it operates directly on mean-payoff parity games, and 3. it is more flexible since it computes the values exactly instead of just comparing them to an integer threshold.

4 Mean-penalty parity games

In this second part of the paper, we define multi-strategies and *mean-penalty parity games*. We reduce these games to mean-payoff parity games, show that their value problem is in NP $\cap$ coNP, and propose a deterministic algorithm for computing the values, which runs in pseudo-polynomial time if the number of priorities is bounded.

4.1 Definitions

Syntactically, a *mean-penalty parity game* is a mean-payoff parity game with non-negative weights, i.e. a tuple $\mathcal{G} = (G, \chi)$, where $G = (Q_1, Q_2, E, \text{weight})$ is a weighted game graph with weight: $E \to \mathbb{R}^{\geq 0}$ (or weight: $E \to \mathbb{N}$ for algorithmic purposes), and $\chi\colon Q \to \mathbb{N}$ is a priority function assigning a priority to every state. As for mean-payoff parity games, a play ρ is parity-winning if the minimal priority occurring infinitely often ($\min\{\chi(q) \mid q \in \text{Inf}(\rho)\}$) is even.

Since we are interested in controller synthesis, we define multi-strategies only for Player 1 (who represents the system). Formally, a *multi-strategy* (for Player 1) in $\mathcal{G}$ is a function $\sigma\colon Q^*Q_1 \to \mathcal{P}(Q)\setminus\{\emptyset\}$ such that $\sigma(\gamma q) \subseteq qE$ for all $\gamma \in Q^*$ and $q \in Q_1$. A play ρ of $\mathcal{G}$ is *consistent* with a multi-strategy σ if $\rho(k+1) \in \sigma(\rho[0,k])$ for all $k \in \mathbb{N}$ with $\rho(k) \in Q_1$, and we denote by $\mathrm{Out}^{\mathcal{G}}(\sigma, q_0)$ the set of all plays ρ of $\mathcal{G}$ that are consistent with σ and start in $\rho(0) = q_0$.

Note that, unlike for deterministic strategies, there is, in general, no unique play consistent with a multi-strategy σ for Player 1 and a (deterministic) strategy τ for Player 2 from a given initial state. Finally, note that every deterministic strategy can be viewed as a multi-strategy.

Let $\mathcal{G}$ be a mean-penalty parity game, and let σ be a multi-strategy. We inductively define $\mathrm{penalty}^{\mathcal{G}}_{\sigma}(\gamma)$ (the *total penalty* of γ wrt. σ) for all $\gamma \in Q^*$ by setting $\mathrm{penalty}^{\mathcal{G}}_{\sigma}(\varepsilon) = 0$ as well as $\mathrm{penalty}^{\mathcal{G}}_{\sigma}(\gamma q) = \mathrm{penalty}^{\mathcal{G}}_{\sigma}(\gamma)$ if $q \in Q_2$ and

$$\mathrm{penalty}^{\mathcal{G}}_{\sigma}(\gamma q) = \mathrm{penalty}^{\mathcal{G}}_{\sigma}(\gamma) + \sum_{q' \in qE\setminus\sigma(\gamma q)} \mathrm{weight}(q, q')$$

if $q \in Q_1$. Hence, $\mathrm{penalty}^{\mathcal{G}}_{\sigma}(\gamma)$ is the total weight of transitions blocked by σ along γ. The *mean penalty* of an infinite play ρ is then defined as the average penalty that is incurred along this play in the limit, i.e.

$$\mathrm{penalty}^{\mathcal{G}}_{\sigma}(\rho) = \begin{cases} \limsup_{n\to\infty} \frac{1}{n}\,\mathrm{penalty}^{\mathcal{G}}_{\sigma}(\rho[0,n)) & \text{if } \rho \text{ is parity-winning,} \\ \infty & \text{otherwise.} \end{cases}$$

The mean penalty of a strategy σ from a given initial state q_0 is defined as the supremum over the mean penalties of all plays that are consistent with σ, i.e.

$$\mathrm{penalty}^{\mathcal{G}}(\sigma, q_0) = \sup\{\mathrm{penalty}^{\mathcal{G}}_{\sigma}(\rho) \mid \rho \in \mathrm{Out}^{\mathcal{G}}(\sigma, q_0)\}.$$

The *value* of a state q_0 in a mean-penalty parity game $\mathcal{G}$ is the least mean penalty that a multi-strategy of Player 1 can achieve, i.e. $\mathrm{val}^{\mathcal{G}}(q_0) = \inf_{\sigma} \mathrm{penalty}^{\mathcal{G}}(\sigma, q_0)$, where σ ranges over all multi-strategies of Player 1. A multi-strategy σ is called *optimal* if $\mathrm{penalty}^{\mathcal{G}}(\sigma, q_0) = \mathrm{val}^{\mathcal{G}}(q_0)$ for all $q_0 \in Q$.

Finally, the *value problem for mean-penalty parity games* is the following decision problem: Given a mean-penalty parity game $\mathcal{G} = (G, \chi)$, an initial state $q_0 \in Q$, and a number $x \in \mathbb{Q}$, decide whether $\mathrm{val}^{\mathcal{G}}(q_0) \leq x$.

Example 11. Fig. 2 represents a mean-penalty parity game. Note that weights of transitions out of Player 2 states are not indicated as they are irrelevant for the mean penalty. In this game, Player 1 (controlling circle states) has to regularly *block* the self-loop if she wants to enforce infinitely many visits to the state with priority 0. This comes with a penalty of 2. However, the multi-strategy in which she blocks no transition can be played safely for an arbitrary number of times. Hence Player 1 can win with mean-penalty 0 (but infinite memory), by blocking the self-loop once every k moves, where k grows with the number of visits to q_2.

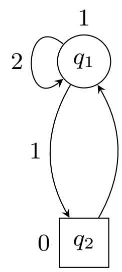

Fig. 2. A mean-penalty parity game.

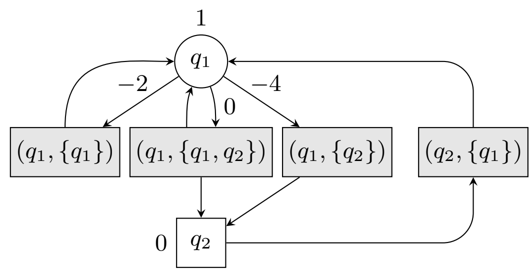

Fig. 3. The corresponding mean-payoff parity game.

4.2 Strategy complexity

In order to solve mean-penalty games, we reduce them to mean-payoff parity games. We construct from a given mean-penalty parity game $\mathcal{G}$ an exponential-size mean-payoff parity game $\mathcal{G}'$, similar to [3] but with an added priority function. Formally, for a mean-penalty parity game $\mathcal{G} = (G, \chi)$ with game graph $G = (Q_1, Q_2, E, \text{weight})$, the game graph $G' = (Q_1', Q_2', E', \text{weight}')$ of the corresponding mean-payoff parity game $\mathcal{G}'$ is defined as follows:

- $Q_1' = Q_1$ and $Q_2' = Q_2 \cup \bar{Q}$, where $\bar{Q} := \{(q, F) \mid q \in Q,\ \emptyset \neq F \subseteq qE\}$;
- E' is the (disjoint) union of three kinds of transitions:
  (1) transitions of the form $(q, (q, F))$ for each $q \in Q_1$ and $\emptyset \neq F \subseteq qE$,
  (2) transitions of the form $(q, (q, \{q'\}))$ for each $q \in Q_2$ and $q' \in qE$,
  (3) transitions of the form $((q, F), q')$ for each $q' \in F$;
- the weight function weight$'$ assigns 0 to transitions of type (2) and (3), but $\text{weight}'(q, (q, F)) = -2 \sum_{q' \in qE \setminus F} \text{weight}(q, q')$ to transitions of type (1).

Finally, the priority function χ' of $\mathcal{G}'$ coincides with χ on Q and assigns priority $M := \max\{\chi(q) \mid q \in Q\}$ to all states in $\bar{Q}$.

Example 12. Fig. 3 depicts the mean-payoff parity game obtained from the mean-penalty parity game from Example 11, depicted in Fig. 2.

The correspondence between $\mathcal{G}$ and $\mathcal{G}'$ is expressed in the following lemma.

Lemma 13. *Let $\mathcal{G}$ be a mean-penalty parity game, $\mathcal{G}'$ the corresponding mean-payoff parity game, and $q_0 \in Q$.*

1. *For every multi-strategy σ in $\mathcal{G}$ there exists a strategy σ' for Player 1 in $\mathcal{G}'$ such that* $\text{val}(\sigma', q_0) \geq -\text{penalty}(\sigma, q_0)$.
2. *For every strategy σ' for Player 1 in $\mathcal{G}'$ there exists a multi-strategy σ in $\mathcal{G}$ such that* $\text{penalty}(\sigma, q_0) \leq -\text{val}(\sigma', q_0)$.
3. $\text{val}^{\mathcal{G}'}(q_0) = -\text{val}^{\mathcal{G}}(q_0)$.

Proof. Clearly, 3. is implied by 1. and 2., and we only need to prove the first two statements. To prove 1., let σ be a multi-strategy in $\mathcal{G}$. For a play prefix $\gamma = q_0(q_0, F_0)\cdots q_n(q_n, F_n)$ in $\mathcal{G}'$, let $\widetilde{\gamma} := q_0 \cdots q_n$ be the corresponding play prefix in $\mathcal{G}$. We set $\sigma'(\gamma q) = (q, F)$ if $q \in Q_1$ and $\sigma(\widetilde{\gamma} q) = F$. Clearly, for each $\rho' \in \mathrm{Out}(\sigma', q_0)$ there exists a play $\rho \in \mathrm{Out}(\sigma, q_0)$ with $-\,\mathrm{penalty}_\sigma(\rho) = \mathrm{payoff}(\rho')$ (namely $\rho(i) = \rho'(2i)$ for all $i \in \mathbb{N}$). Hence,

$$\begin{aligned}
\mathrm{val}^{\mathcal{G}'}(\sigma', q_0) &= \inf\{\mathrm{payoff}(\rho') \mid \rho' \in \mathrm{Out}(\sigma', q_0)\} \\
&\geq \inf\{-\,\mathrm{penalty}_\sigma(\rho) \mid \rho \in \mathrm{Out}(\sigma, q_0)\} \\
&= -\sup\{\mathrm{penalty}_\sigma(\rho) \mid \rho \in \mathrm{Out}(\sigma, q_0)\} \\
&= -\,\mathrm{penalty}(\sigma, q_0)\,.
\end{aligned}$$

To prove 2., let σ' be a strategy for Player 1 in $\mathcal{G}'$. For a play prefix $\gamma = q_0 \cdots q_n$ in $\mathcal{G}$, we inductively define the corresponding play prefix $\widetilde{\gamma}$ in $\mathcal{G}'$ by setting $\widetilde{q} = q$ and $\widetilde{\gamma q} = \widetilde{\gamma} \cdot \sigma'(\widetilde{\gamma}) \cdot q$. We set $\sigma(\gamma) = F$ if $\sigma'(\widetilde{\gamma}) = (q, F)$. For each $\rho \in \mathrm{Out}(\sigma, q_0)$ there exists a play $\rho' \in \mathrm{Out}(\sigma', q_0)$ with $\mathrm{penalty}_\sigma(\rho) = -\,\mathrm{payoff}(\rho')$, namely the play ρ' defined by $\rho'(2i) = \rho(i)$ and

$$\rho'(2i+1) = \begin{cases} (\rho(i), \sigma(\rho[0,i])) & \text{if } \rho(i) \in Q_1, \\ (\rho(i), \{\rho(i+1)\}) & \text{if } \rho(i) \in Q_2, \end{cases}$$

for all $i \in \mathbb{N}$. Hence,

$$\begin{aligned}
\mathrm{penalty}(\sigma, q_0) &= \sup\{\mathrm{penalty}_\sigma(\rho) \mid \rho \in \mathrm{Out}(\sigma, q_0)\} \\
&\leq \sup\{-\,\mathrm{payoff}(\rho') \mid \rho' \in \mathrm{Out}(\sigma', q_0)\} \\
&= -\inf\{\mathrm{payoff}(\rho') \mid \rho' \in \mathrm{Out}(\sigma', q_0)\} \\
&= -\,\mathrm{val}^{\mathcal{G}'}(\sigma', q_0)\,.
\end{aligned}$$

□

It follows from Theorem 3 and Lemma 13 that every mean-penalty parity game admits an optimal multi-strategy.

Corollary 14. *In every mean-penalty parity game, Player 1 has an optimal multi-strategy.*

We now show that Player 2 has a memoryless optimal strategy of a special kind in the mean-payoff parity game derived from a mean-penalty parity game. This puts the value problem for mean-penalty parity games into coNP, and is also a crucial point in the proof of Lemma 17 below.

Lemma 15. *Let $\mathcal{G}$ be a mean-penalty parity game and $\mathcal{G}'$ the corresponding mean-payoff parity game. Then in $\mathcal{G}'$ there is a memoryless optimal strategy τ' for Player 2 such that for every $q \in Q$ there exists a total order $\leq_q$ on the set qE with $\tau'((q, F)) = \min_{\leq_q} F$ for every state $(q, F) \in \bar{Q}$.*

Proof. Let τ be a memoryless optimal strategy for Player 2 in $\mathcal{G}'$. For a state q, we consider the set qE and order it in the following way. We inductively define

$F_1 = qE$, $q_i = \tau((q, F_i))$ and $F_{i+1} = F_i \setminus \{q_i\}$ for every $1 \leq i \leq |qE|$. Note that $\{q_1, \ldots, q_{|qE|}\} = qE$. We set $q_1 \leq_q q_2 \leq_q \cdots \leq_q q_{|qE|}$ and define a new memoryless strategy τ' for Player 2 in $\mathcal{G}'$ by $\tau'((q, F)) = \min_{\leq_q} F$ for $(q, F) \in \bar{Q}$ and $\tau'(q) = \tau(q)$ for all $q \in Q_2$. To prove the lemma, we have to show that τ' is at least as good as τ and thus optimal.

Let $q_0 \in Q$ and $\rho' \in \mathrm{Out}(\tau', q_0)$. We construct a play $\rho \in \mathrm{Out}(\tau, q_0)$ with $\mathrm{payoff}(\rho) \geq \mathrm{payoff}(\rho')$ in the following way. For every position i with $\rho'(i) = (q, F')$, let $F = \{q' \in qE \mid \tau'((q, F')) \leq_q q'\}$ (then $\tau((q, F)) = \tau'((q, F'))$ by the definition of τ') and set $\rho(i) = (q, F)$. For every other position i, let $\rho(i) = \rho'(i)$. Note that $\rho \in \mathrm{Out}(\tau, q_0)$ and $\min \chi(\mathrm{Inf}(\rho)) = \min \chi(\mathrm{Inf}(\rho'))$. Moreover, we have $F' \subseteq F$ and therefore $\mathrm{weight}'(q, (q, F')) \leq \mathrm{weight}'(q, (q, F))$ whenever $\rho'(i) = (q, F')$ and $\rho(i) = (q, F)$ (because weights in $\mathcal{G}$ are nonnegative). Hence, $\mathrm{payoff}(\rho) \geq \mathrm{payoff}(\rho')$. Since ρ' was chosen arbitrarily, it follows that

$$\begin{aligned}
\mathrm{val}(\tau, q_0) &= \sup\{\mathrm{payoff}(\rho) \mid \rho \in \mathrm{Out}(\tau, q_0)\} \\
&\geq \sup\{\mathrm{payoff}(\rho') \mid \rho' \in \mathrm{Out}(\tau', q_0)\} \\
&= \mathrm{val}(\tau', q_0)\,.
\end{aligned}$$

Hence, τ' is optimal. □

4.3 Computational complexity

In order to put the value problem for mean-penalty parity games into NP ∩ coNP, we propose a more sophisticated reduction from mean-penalty parity games to mean-payoff parity games, which results in a polynomial-size mean-payoff parity game. Intuitively, in a state $q \in Q_1$ we ask Player 1 *consecutively* for each outgoing transition whether he wants to block that transition. If he allows a transition, then Player 2 has to decide whether she wishes to explore this transition. Finally, after all transitions have been processed in this way, the play proceeds along the *last* transition that Player 2 has desired to explore.

Formally, let us fix a mean-penalty parity game $\mathcal{G} = (G, \chi)$ with game graph $G = (Q_1, Q_2, E, \mathrm{weight})$, and denote by $k := \max\{|qE| \mid q \in Q\}$ the maximal out-degree of a state. Then the polynomial-size mean-payoff parity game $\mathcal{G}''$ has vertices of the form q and (q, a, i, m), where $q \in Q$, $a \in \{\mathrm{choose}, \mathrm{allow}, \mathrm{block}\}$, $i \in \{1, \ldots, k+1\}$ and $m \in \{0, \ldots, k\}$; vertices of the form q and $(q, \mathrm{choose}, i, m)$ belong to Player 1, while vertices of the form $(q, \mathrm{allow}, i, m)$ or $(q, \mathrm{block}, i, m)$ belong to Player 2. To describe the transition structure of $\mathcal{G}$, let $q \in Q$ and assume that $qE = \{q_1, \ldots, q_k\}$ (a state may occur more than once in this list). Then the following transitions originate in a state of the form q or (q, a, i, m):

1. a transition from q to $(q, \mathrm{choose}, 1, 0)$ with weight 0,
2. for all $1 \leq i \leq k$ and $0 \leq m \leq k$ a transition from $(q, \mathrm{choose}, i, m)$ to $(q, \mathrm{allow}, i, m)$ with weight 0,
3. if $q \in Q_1$ then for all $1 \leq i \leq k$ and $0 \leq m \leq k$ a transition from $(q, \mathrm{choose}, i, m)$ to $(q, \mathrm{block}, i, m)$ with weight 0, *except* if $i = k$ and $m = 0$;

4. for all $0 \leq m \leq k$ a transition from $(q, \text{choose}, k+1, m)$ to q_m with weight 0 (where q_0 can be chosen arbitrarily),
5. for all $1 \leq i \leq k$ and $0 \leq m \leq k$ a transition from (q, allow, i, m) to $(q, \text{choose}, i+1, i)$ with weight 0,
6. for all $1 \leq i \leq k$ and $1 \leq m \leq k$ a transition from (q, allow, i, m) to $(q, \text{choose}, i+1, m)$ with weight 0,
7. for all $1 \leq i \leq k$ and $0 \leq m \leq k$ a transition from (q, block, i, m) to $(q, \text{choose}, i+1, m)$ with weight $-2(k+1) \cdot \text{weight}(q, q_i)$.

Finally, the priority of a state $q \in Q$ equals the priority of the same state in $\mathcal{G}$, whereas all states of the form (q, a, i, m) have priority $M = \max\{\chi(q) \mid q \in Q\}$.

Example 16. For the game of Fig. 2, this transformation would yield the game depicted in Fig. 4. In this picture, a, b and c stand for *allow*, *block* and *choose*, respectively; zero weights are omitted.

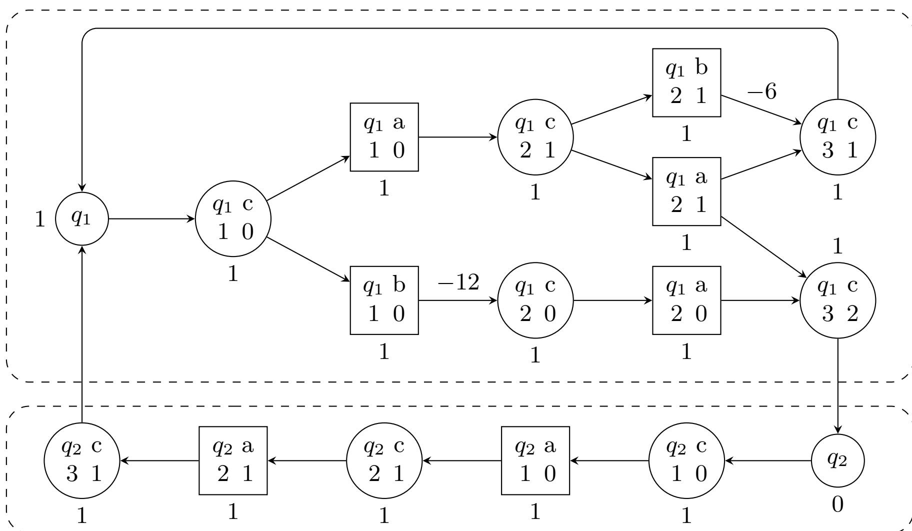

Fig. 4. The game $\mathcal{G}''$ associated with the game $\mathcal{G}$ of Fig. 2.

It is easy to see that the game $\mathcal{G}''$ has polynomial size and can, in fact, be constructed in polynomial time from the given mean-penalty parity game $\mathcal{G}$. The following lemma relates the game $\mathcal{G}''$ to the mean-payoff parity game $\mathcal{G}'$ of exponential size constructed in Sect. 4.2 and to the original game $\mathcal{G}$.

Lemma 17. *Let $\mathcal{G}$ be a mean-penalty parity game, $\mathcal{G}'$ the corresponding mean-payoff parity game of exponential size, $\mathcal{G}''$ the corresponding mean-payoff parity game of polynomial size, and $q_0 \in Q$.*

1. *For every multi strategy σ in $\mathcal{G}$ there exists a strategy σ' for Player 1 in $\mathcal{G}''$ such that* $\mathrm{val}(\sigma', q_0) \geq -\mathrm{penalty}(\sigma, q_0)$.
2. *For every strategy τ for Player 2 in $\mathcal{G}'$ there exists a strategy τ' for Player 2 in $\mathcal{G}''$ such that* $\mathrm{val}(\tau', q_0) \leq \mathrm{val}(\tau, q_0)$.
3. $\mathrm{val}^{\mathcal{G}''}(q_0) = -\mathrm{val}^{\mathcal{G}}(q_0)$.

Proof. To prove 1., let σ be a multi-strategy in $\mathcal{G}$. For any play prefix γ in $\mathcal{G}''$, let $\widetilde{\gamma}$ be the projection to states in $\mathcal{G}$ (i.e. all states of the form (q, a, i, m) are omitted). Assuming that $q_1, \ldots, q_k$ is the enumeration of qE used in the definition of $\mathcal{G}''$, we set $\sigma'(\gamma \cdot (q, \mathrm{choose}, i, m)) = (q, \mathrm{allow}, i, m)$ if (and only if) either $q \in Q_1$ and $q_i \in \sigma(\widetilde{\gamma})$ or $q \in Q_2$. It is easy to see that for each $\rho' \in \mathrm{Out}(\sigma', q_0)$ there exists a play $\rho \in \mathrm{Out}(\sigma, q_0)$ with $-\mathrm{penalty}_\sigma(\rho) = \mathrm{payoff}(\rho')$. Hence,

$$\begin{aligned}
\mathrm{val}(\sigma', q_0) &= \inf\{\mathrm{payoff}(\rho') \mid \rho' \in \mathrm{Out}(\sigma', q_0)\} \\
&\geq \inf\{-\mathrm{penalty}_\sigma(\rho) \mid \rho \in \mathrm{Out}(\sigma, q_0)\} \\
&= -\sup\{\mathrm{penalty}_\sigma(\rho) \mid \rho \in \mathrm{Out}(\sigma, q_0)\} \\
&= -\mathrm{penalty}(\sigma, q_0)\,.
\end{aligned}$$

To prove 2., let τ be a strategy for Player 2 in $\mathcal{G}'$. By Lemma 15, there exists a memoryless strategy τ^* for Player 2 in $\mathcal{G}'$ such that $\mathrm{val}(\tau^*, q_0) \leq \mathrm{val}(\tau, q_0)$ and for all $q \in Q$ there exists a total order $\leq_q$ on qE with $\tau^*((q, F)) = \min_{\leq_q} F$ for all $(q, F) \in \bar{Q}$. We define a memoryless strategy τ' for Player 2 in $\mathcal{G}''$ as follows: Assume that $q_1, \ldots, q_k$ is the enumeration of qE used in the definition of $\mathcal{G}''$. Then we set $\tau'((q, \mathrm{allow}, i, m)) = (q, \mathrm{choose}, i+1, i)$ if (and only if) one of the following three conditions is fulfilled: 1. $m = 0$, or 2. $q \in Q_1$ and $q_i \leq_q q_m$, or 3. $q \in Q_2$ and $\tau^*(q) = (q, \{q_i\})$. Now it is easy to see that for each $\rho' \in \mathrm{Out}(\tau', q_0)$ there exists a play $\rho \in \mathrm{Out}(\tau^*, q_0)$ with $\mathrm{payoff}(\rho) = \mathrm{payoff}(\rho')$. Hence,

$$\begin{aligned}
\mathrm{val}(\tau', q_0) &= \sup\{\mathrm{payoff}(\rho') \mid \rho' \in \mathrm{Out}(\tau', q_0)\} \\
&\leq \sup\{\mathrm{payoff}(\rho) \mid \rho \in \mathrm{Out}(\tau^*, q_0)\} \\
&= \mathrm{val}(\tau^*, q_0) \\
&\leq \mathrm{val}(\tau, q_0)\,.
\end{aligned}$$

Finally, we prove 3. It follows from 1. that $\mathrm{val}^{\mathcal{G}''}(q_0) \geq -\mathrm{val}^{\mathcal{G}}(q_0)$, and it follows from 2. that $\mathrm{val}^{\mathcal{G}''}(q_0) \leq \mathrm{val}^{\mathcal{G}'}(q_0)$. But $\mathrm{val}^{\mathcal{G}'}(q_0) = -\mathrm{val}^{\mathcal{G}}(q_0)$ by Lemma 13, and therefore $\mathrm{val}^{\mathcal{G}''}(q_0) = -\mathrm{val}^{\mathcal{G}}(q_0)$. □

Since the mean-payoff game $\mathcal{G}''$ can be computed from $\mathcal{G}$ in polynomial time, we obtain a polynomial-time many-one reduction from the value problem for mean-penalty parity games to the value problem for mean-payoff parity games. By Corollary 6 and Theorem 9, the latter problem belongs to NP ∩ coNP.

Theorem 18. *The value problem for mean-penalty parity games belongs to NP ∩ coNP.*

4.4 A deterministic algorithm

Naturally, we can use the polynomial translation from mean-penalty parity games to mean-payoff parity games to solve mean-penalty parity games deterministically. Note that the mean-payoff parity game $\mathcal{G}''$ derived from a mean-penalty parity game has $\mathrm{O}(|Q| \cdot k^2)$ states and $\mathrm{O}(|Q| \cdot k^2)$ edges (the number of priorities remains constant), where k is the maximum out-degree of a state in $\mathcal{G}$. Moreover, if weights are given in integers and W is the highest absolute weight in $\mathcal{G}$, then the highest absolute weight in $\mathcal{G}''$ is $\mathrm{O}(k \cdot W)$. Using Theorem 10, we thus obtain a deterministic algorithm for solving mean-penalty parity games that runs in time $\mathrm{O}(|Q|^{d+3} \cdot k^{2d+7} \cdot W)$. If k is a constant, the running time is $\mathrm{O}(|Q|^{d+3} \cdot W)$, which is acceptable. In the general case however, the best upper bound on k is the number of states, and we get an algorithm that runs in time $\mathrm{O}(|Q|^{3d+10} \cdot W)$. Even if the numbers of priorities is small, this running time would not be acceptable in practical applications.

The goal of this section is to show that we can do better; namely we will give an algorithm that runs in time $\mathrm{O}(|Q|^{d+3} \cdot |E| \cdot W)$, independently of the maximum out-degree. The idea is as follows: we use Algorithm SolveMPP on the mean-payoff parity game $\mathcal{G}'$ of exponential size, but we show that we can run it *on* $\mathcal{G}$, i.e., by handling the extra states of $\mathcal{G}'$ symbolically during the computation. As a first step, we adapt the pseudo-polynomial algorithm by Zwick & Paterson [24] to compute the values of a mean-penalty parity game with a trivial parity objective.

Lemma 19. *The values of a mean-penalty parity game with priority function* $\chi \equiv 0$ *can be computed in time* $\mathrm{O}(|Q|^4 \cdot |E| \cdot W)$.

Proof. Let $\mathcal{G} = (G, \chi)$ with $G = (Q_1, Q_2, E, \mathrm{weight})$, and $\mathcal{G}' = (G', \chi')$ with $G' = (Q'_1, Q'_2, E', \mathrm{weight}')$. For a state $q \in Q'$, we let $v_0(q) = 0$, and for $k > 0$, we define

$$v_k(q) = \begin{cases} \max\limits_{q' \in qE'} \mathrm{weight}'(q, q') + v_{k-1}(q') & \text{if } q \in Q'_1, \\ \min\limits_{q' \in qE'} \mathrm{weight}'(q, q') + v_{k-1}(q') & \text{if } q \in Q'_2. \end{cases}$$

If $q \in Q$, then the definition of $\mathcal{G}'$ yields that

$$v_k(q) = \begin{cases} \max\limits_{F \subseteq qE} \mathrm{weight}'(q, (q, F)) + \min\limits_{q' \in F} v_{k-2}(q') & \text{if } q \in Q_1, \\ \min\limits_{q' \in qE} v_{k-2}(q') & \text{if } q \in Q_2, \end{cases}$$

In the first case, a naïve computation would require the examination of an exponential number of transitions. In order to avoid this blow-up, we use the same idea as in the proof of Lemma 15: Let $qE = \{q_1, \ldots, q_r\}$ be sorted in such a way that $i \leq j$ implies $v_{k-2}(q_i) \leq v_{k-2}(q_j)$. Since $\mathrm{weight}'(q, (q, F)) \leq \mathrm{weight}'(q, (q, F'))$ if $F \subseteq F'$, we have

$$v_k(q) = \max_i \mathrm{weight}'(q, (q, \{q_i, \ldots, q_r\})) + v_{k-2}(q_i).$$

Hence the sequence v_{2k} can be computed in time $\mathrm{O}(k \cdot |E|)$ on Q. Now, despite the exponential size of $\mathcal{G}'$, the length of a simple cycle in $\mathcal{G}'$ is at most $2|Q|$. Hence, Theorem 2.2 in [24] becomes

$$2k \cdot \mathrm{val}^{\mathcal{G}'}(q) - 4|Q| \cdot W' \le v_{2k}(q) \le 2k \cdot \mathrm{val}^{\mathcal{G}'}(q) + 4|Q| \cdot W'$$

for all $q \in Q$, where W' is the maximal absolute weight in $\mathcal{G}'$. Since $W' \le |Q| \cdot 2W$, it follows from [24] that $\mathrm{val}^{\mathcal{G}} = -\mathrm{val}^{\mathcal{G}'} \restriction Q$ can be computed in time $\mathrm{O}(|Q|^4 \cdot |E| \cdot W)$. ◻

Now, given a mean-penalty parity game $\mathcal{G}$ with associated mean-payoff parity game $\mathcal{G}'$ and a set T of states of $\mathcal{G}$, we define

$$\triangle^{\mathcal{G}}(T) = T \cup \{(q,F) \in \bar{Q} \mid F \subseteq T\};$$
$$\blacktriangle^{\mathcal{G}}(T) = T \cup \{(q,F) \in \bar{Q} \mid F \cap T \neq \emptyset\}.$$

We usually omit to mention the superscript $\mathcal{G}$ when it is clear from the context.

Lemma 20. *If S is a subarena of $\mathcal{G}$, then $\triangle(S)$ and $\blacktriangle(S)$ are subarenas of $\mathcal{G}'$.*

Proof. Assume that S is a subarena of $\mathcal{G}$, and pick a state q in $\triangle(S)$. If $q \in Q$, then it also belongs to S and, as a state of $\mathcal{G}$, has a successor q' in S. Then $\triangle(S)$ contains $(q, \{q'\})$, which is a successor of q. If q belongs to $\bar{Q}$, then $qE' \subseteq S$ by definition of $\triangle(S)$; hence it has at least one successor in S. A similar argument shows that $\blacktriangle(S)$ is also a subarena of $\mathcal{G}'$. ◻

Lemma 21. *Let $\mathcal{G}$ be a mean-penalty parity game with associated mean-payoff parity game $\mathcal{G}'$, and let $A, B \subseteq Q$. Then*

$$\triangle(A \cap B) = \triangle(A) \cap \triangle(B), \qquad \triangle(A \cup B) \supseteq \triangle(A) \cup \triangle(B),$$
$$\blacktriangle(A \cup B) = \blacktriangle(A) \cup \blacktriangle(B), \qquad \blacktriangle(A \cap B) \subseteq \blacktriangle(A) \cap \blacktriangle(B),$$
$$\triangle(Q \setminus A) = Q' \setminus \blacktriangle(A), \qquad \blacktriangle(Q \setminus A) = Q' \setminus \triangle(A)\,.$$

Proof. Straightforward. ◻

Lemma 22. *Let $\mathcal{G}$ be a mean-penalty parity game with associated mean-payoff parity game $\mathcal{G}'$, and let $F \subseteq Q$. Then*

$$\triangle(\mathrm{Attr}_1^{\mathcal{G}}(F)) = \mathrm{Attr}_1^{\mathcal{G}'}(F) = \mathrm{Attr}_1^{\mathcal{G}'}(\triangle(F)),$$
$$\blacktriangle(\mathrm{Attr}_2^{\mathcal{G}}(F)) = \mathrm{Attr}_2^{\mathcal{G}'}(F) = \mathrm{Attr}_2^{\mathcal{G}'}(\blacktriangle(F))\,.$$

Proof. We only prove the first statement; the second can be proved using similar arguments. Clearly, $\mathrm{Attr}_1^{\mathcal{G}'}(F) = \mathrm{Attr}_1^{\mathcal{G}'}(\triangle(F))$, so we only need to prove that $\triangle(\mathrm{Attr}_1^{\mathcal{G}}(F)) = \mathrm{Attr}_1^{\mathcal{G}'}(F)$. First pick $q \in \triangle(\mathrm{Attr}_1^{\mathcal{G}}(F))$. If $q \in Q$, then the attractor strategy for reaching F can be mimicked in $\mathcal{G}'$, and therefore $q \in \mathrm{Attr}_1^{\mathcal{G}'}(F)$. On the other hand, if $q \in \bar{Q}$, then all successors of q lie in $\mathrm{Attr}_1^{\mathcal{G}}(F)$ and therefore also in $\mathrm{Attr}_1^{\mathcal{G}'}(F)$. Hence, $q \in \mathrm{Attr}_1^{\mathcal{G}'}(F)$. Now pick $q \in \mathrm{Attr}_1^{\mathcal{G}'}(F)$.

Algorithm SymbSolveMPP($\mathcal{G}$)

Input: mean-penalty parity game $\mathcal{G} = (G, \chi)$
Output: $\mathrm{val}^{\mathcal{G}}$

if $Q = \emptyset$ **then return** $\emptyset$
$p \coloneqq \min\{\chi(q) \mid q \in Q\}$
if p is even **then**
  $g \coloneqq \mathrm{SymbSolveMP}(G, 0)$
  if $\chi(q) = p$ for all $q \in Q$ **then return** g
  $T \coloneqq Q \setminus \mathrm{Attr}_1^{\mathcal{G}}(\chi^{-1}(p))$; $f \coloneqq \mathrm{SymbSolveMPP}(\mathcal{G} \restriction T)$
  $x \coloneqq \max(f(T) \cup g(Q))$; $A \coloneqq \mathrm{Attr}_2^{\mathcal{G}}(f^{-1}(x) \cup g^{-1}(x))$
  return $(Q \to \mathbb{R} \cup \{\infty\} \colon q \mapsto x) \sqcap \mathrm{SymbSolveMPP}(\mathcal{G} \restriction Q \setminus A)$
else
  $T \coloneqq Q \setminus \mathrm{Attr}_2^{\mathcal{G}}(\chi^{-1}(p))$
  if $T = \emptyset$ **then return** $(Q \to \mathbb{R} \cup \{\infty\} \colon q \mapsto \infty)$
  $f \coloneqq \mathrm{SymbSolveMPP}(\mathcal{G} \restriction T)$; $x \coloneqq \min f(T)$; $A \coloneqq \mathrm{Attr}_1^{\mathcal{G}}(f^{-1}(x))$
  return $(Q \to \mathbb{R} \cup \{\infty\} \colon q \mapsto x) \sqcup \mathrm{SymbSolveMPP}(\mathcal{G} \restriction Q \setminus A)$
end if

If $q \in Q$, then the attractor strategy for reaching F yields a multi-strategy σ in $\mathcal{G}$ such that all plays $\rho \in \mathrm{Out}^{\mathcal{G}}(\sigma, q)$ visit F. Hence, $q \in \mathrm{Attr}_1^{\mathcal{G}}(F) \subseteq \triangle(\mathrm{Attr}_1^{\mathcal{G}}(F))$. On the other hand, if $q \in \bar{Q}$, then all successors of q lie in $Q \cap \mathrm{Attr}_1^{\mathcal{G}'}(F)$ (since q is a Player 2 state) and therefore also in $\mathrm{Attr}_1^{\mathcal{G}}(F)$. Hence, $q \in \triangle(\mathrm{Attr}_1^{\mathcal{G}}(F))$. ☐

Algorithm SymbSolveMPP is our algorithm for computing the values of a mean-penalty parity game. The algorithm employs as a subroutine an algorithm SymbSolveMP for computing the values of a mean-penalty parity with a trivial priority function (see Lemma 19). Since SymbSolveMP can be implemented to run in time $\mathrm{O}(|Q|^4 \cdot |E| \cdot W)$, the running time of the procedure SymbSolveMPP is $\mathrm{O}(|Q|^{d+3} \cdot |E| \cdot W)$. Notably, the algorithm runs in polynomial time if the number of priorities is bounded and we are only interested in the average *number* of edges blocked by a strategy in each step (i.e. if all weights are equal to 1).

Theorem 23. *The values of a mean-penalty parity game with d priorities can be computed in time* $\mathrm{O}(|Q|^{d+3} \cdot |E| \cdot W)$*.*

Proof. From Lemma 19 and with the same runtime analysis as in the proof of Theorem 10, we get that SymbSolveMPP runs in time $\mathrm{O}(|Q|^{d+3} \cdot |E| \cdot W)$. We now prove that the algorithm is correct, by proving that there is a correspondence between the values the algorithm computes on a mean-penalty parity game $\mathcal{G}$ and the values computed by Algorithm SolveMPP on the mean-payoff parity game $\mathcal{G}'$. More precisely, we show that $\mathrm{SolveMPP}(\mathcal{G}') \restriction Q = -\mathrm{SymbSolveMPP}(\mathcal{G})$. The correctness of the algorithm thus follows from Lemma 13, which states that $\mathrm{val}^{\mathcal{G}'} \restriction Q = -\mathrm{val}^{\mathcal{G}}$.

The proof is by induction on the number of states in $\mathcal{G}$. The result holds trivially if $Q = \emptyset$. Otherwise, assume that the result is true for all games with less than $|Q|$ states and let $p = \min\{\chi(q) \mid q \in Q\}$. By construction, p is also the minimal priority in $\mathcal{G}'$. We only consider the case that p is even; the other case is proved using the same arguments.

Write g', T', f', x' and A' for the items computed by SymbSolveMPP on $\mathcal{G}'$, while q, T, f, x and A are the corresponding items computed by SolveMPP on $\mathcal{G}$. Then $g'(q) = -g(q)$ for all $q \in Q$, and $g'((q,F)) = \min_{q' \in F} g'(q')$ for all $(q,F) \in \bar{Q}$ (since such states belongs to Player 2). If $\mathcal{G}$ has only one priority, the result follows. Otherwise, by Lemmas 21 and 22, we have $T' = \blacktriangle(T)$. However, any state $(q,F) \in T'$ that is not a state of the game $(\mathcal{G} \restriction T)'$ has no predecessor in $\mathcal{G}' \restriction T'$: if $q \in T'$ then $q \in T \cap Q_1$ and $qE \setminus T \neq \emptyset$, i.e. $qE \cap \mathrm{Attr}_1(\chi^{-1}(p)) \neq \emptyset$; but then $q \in \mathrm{Attr}_1(\chi^{-1}(p))$ and thus $q \notin T$, a contradiction. It follows that $\mathrm{SolveMPP}(\mathcal{G}' \restriction T') \restriction T = \mathrm{SolveMPP}((\mathcal{G} \restriction T)') \restriction T$.

Now, since T is a strict subset of Q, the induction hypothesis applies, so that $f'(t) = -f(t)$ for all $t \in T$. It follows that $x' = -x$. Let $S \coloneqq Q \setminus A$ and $S' \coloneqq Q' \setminus A'$. By Lemma 22, $A' = \blacktriangle(A)$, and by Lemma 21, $S' = \vartriangle(S)$. Again, any state $(q,F) \in S'$ that is not a state of the game $(\mathcal{G} \restriction S)'$ has no predecessor in $\mathcal{G}' \restriction S'$. Hence, $\mathrm{SolveMPP}(\mathcal{G}' \restriction S') \restriction S = \mathrm{SolveMPP}((\mathcal{G} \restriction S)') \restriction S$ Applying the induction hypothesis to the game $G \restriction S$, we get that $\mathrm{SolveMPP}((\mathcal{G} \restriction S)') \restriction S = -\,\mathrm{SymbSolveMPP}(G \restriction S)$, and the result follows for $\mathcal{G}$. □

5 Conclusion

In this paper, we have studied mean-payoff parity games, with an application to finding permissive strategies in parity games with penalties. In particular, we have established that mean-penalty parity games are not harder to solve than mean-payoff parity games: for both kinds of games, the value problem is in NP ∩ coNP and can be solved by an exponential algorithm that becomes pseudo-polynomial when the number of priorities is bounded.

One complication with both kinds of games is that optimal strategies for Player 1 require infinite memory, which makes it hard to synthesise these strategies. A suitable alternative to optimal strategies are *ε-optimal* strategies that achieve the value of the game by at most ε. Since finite-memory ε-optimal strategies are guaranteed to exist [2], a challenge for future work is to modify our algorithms so that they compute not only the values of the game but also a finite-memory ε-optimal (multi-)strategy for Player 1.

Acknowledgement. We thank an anonymous reviewer for pointing out the polynomial reduction from mean-penalty parity games to mean-payoff parity games, which has simplified the proof that mean-penalty parity games are in NP.

References

1. Julien Bernet, David Janin, and Igor Walukiewicz. Permissive strategies: from parity games to safety games. *RAIRO – ITA*, 36(3):261–275, 2002.

2. Roderick Bloem, Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. Better quality in synthesis through quantitative objectives. In *CAV'09*, volume 5643 of *LNCS*, pages 140–156. Springer-Verlag, 2009.
3. Patricia Bouyer, Marie Duflot, Nicolas Markey, and Gabriel Renault. Measuring permissivity in finite games. In *CONCUR'09*, volume 5710 of *LNCS*, pages 196–210. Springer-Verlag, 2009.
4. Patricia Bouyer, Uli Fahrenberg, Kim G. Larsen, Nicolas Markey, and Jiří Srba. Infinite runs in weighted timed automata with energy constraints. In *FORMATS'08*, volume 5215 of *LNCS*, pages 33–47. Springer-Verlag, 2008.
5. Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Resource interfaces. In *EMSOFT'03*, volume 2855 of *LNCS*, pages 117–133. Springer-Verlag, 2003.
6. Krishnendu Chatterjee and Laurent Doyen. Energy parity games. In *ICALP'10 (2)*, volume 6199 of *LNCS*, pages 599–610. Springer-Verlag, 2010.
7. Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Generalized mean-payoff and energy games. In *FSTTCS'10*, volume 8 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.
8. Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdziński. Mean-payoff parity games. In *LICS'05*, pages 178–187. IEEE Computer Society Press, 2005.
9. Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Generalized parity games. In *FoSSaCS'07*, volume 4423 of *LNCS*, pages 153–167. Springer-Verlag, 2007.
10. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT Press, 3rd edition, 2009.
11. Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.
12. E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy. In *FOCS'91*, pages 368–377. IEEE Computer Society Press, 1991.
13. Hugo Gimbert and Wieslaw Zielonka. When can you play positionally? In *MFCS'04*, volume 3153 of *LNCS*, pages 686–697. Springer-Verlag, 2004.
14. Marcin Jurdziński. Deciding the winner in parity games is in UP $\cap$ co-UP. *Information Processing Letters*, 68(3):119–124, 1998.
15. Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23(3):309–311, 1978.
16. Eryk Kopczyński. Half-positional determinacy of infinite games. In *ICALP 2006 (2)*, volume 4052 of *LNCS*, pages 336–347. Springer-Verlag, 2006.
17. Michael Luttenberger. Strategy iteration using non-deterministic strategies for solving parity games. Research Report cs.GT/0806.2923, arXiv, 2008.
18. Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.
19. Andrzej Włodzimierz Mostowski. Games with forbidden positions. Technical Report 78, Instytut Matematyki, Uniwersytet Gdański, Poland, 1991.

20. Sophie Pinchinat and Stéphane Riedweg. You can always compute maximally permissive controllers under partial observation when they exist. In *ACC'05*, pages 2287–2292, 2005.
21. Wolfgang Thomas. Infinite games and verification (extended abstract of a tutorial). In *CAV'04*, volume 2404 of *LNCS*, pages 58–64. Springer-Verlag, 2002.
22. Wiesław Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1–2):135–183, 1998.
23. Wiesław Zielonka. Perfect-information stochastic parity games. In *FoSSaCS'04*, volume 2987 of *LNCS*, pages 499–513. Springer-Verlag, 2004.
24. Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1&2):343–359, 1996.