

Mean-Payoff Parity Games

Krishnendu Chatterjee[†] Thomas A. Henzinger^{†‡} Marcin Jurdziński[§]

[†] Electrical Engineering and Computer Sciences, University of California, Berkeley

[‡] Computer and Communication Sciences, EPFL, Switzerland

[§] Department of Computer Science, University of Warwick, UK

{c.krish,tah,mju}@eecs.berkeley.edu

Abstract

Games played on graphs may have qualitative objectives, such as the satisfaction of an ω -regular property, or quantitative objectives, such as the optimization of a real-valued reward. When games are used to model reactive systems with both fairness assumptions and quantitative (e.g., resource) constraints, then the corresponding objective combines both a qualitative and a quantitative component. In a general case of interest, the qualitative component is a parity condition and the quantitative component is a mean-payoff reward. We study and solve such mean-payoff parity games. We also prove some interesting facts about mean-payoff parity games which distinguish them both from mean-payoff and from parity games. In particular, we show that optimal strategies exist in mean-payoff parity games, but they may require infinite memory.

1. Introduction

Games played on graphs have come to play a central role in many areas of computer science. In particular, when the vertices and edges of a graph represent the states and transitions of a reactive system, then the synthesis problem (Church’s problem) asks for the construction of a winning strategy in a graph game [4, 18, 17]. Game-theoretic formulations have also proved useful for the modeling [9], verification [1], refinement [12], and compatibility checking [8] of reactive systems. In all these cases, the winning objective is typically an ω -regular condition, because the ω -regular sets are expressive enough to model the usual fairness assumptions of reactive systems [14]. Of particular interest is the *parity* (or Rabin chain) condition, which can express all ω -regular sets and enjoys pleasant duality properties [19]. Algorithms for solving parity games have enjoyed much recent interest [13, 20, 2], not least because of the close and intriguing connection between parity games and μ -calculus

model checking [11]. We refer to ω -regular graph games in general, and parity games in particular, as *qualitative* games, because the result of a game is binary for each player (win or lose).

In classical game theory, usually more traditional *quantitative* objectives are pursued, that is, the result of a game is a real-valued payoff for each player. Because of their rich applicability in economics, these games have been studied from an algorithmic perspective for some time [6, 21]. However, recently the interest in quantitative games has been rising for two additional reasons. First, their algorithmics, such as strategy improvement schemes, is hoped to give insight and lead to improved algorithms for qualitative games such as parity games [20, 2]. Second, many natural models of reactive systems, such as probabilistic and resource models, include quantitative information, and the corresponding synthesis question requires the solution of quantitative games. Quantities may represent, for example, the power usage of an embedded component, or the buffer size of a networking element [5]. In such cases, for reactive systems with both quantitative information and fairness assumptions, the appropriate objective combines both a quantitative and qualitative component; for instance, in [5] the authors present an algorithm for solving so-called Büchi-threshold games, where the qualitative component of the objective is a Büchi fairness constraint and the quantitative component mandates that an infinite behavior of a reactive system stay below some threshold power (say) value.

In this paper, we solve a general combination of qualitative and quantitative objectives for graph games, namely, the combination of qualitative parity objectives and quantitative mean-payoff objectives. In a graph game, the vertices are partitioned into player-1 and player-2 vertices: at player-1 vertices, the first player chooses an outgoing edge, and at player-2 vertices, the second player chooses an outgoing edge. The game is played forever and the result, or *play*, of the game is an infinite path through the graph. In a (pure) mean-payoff game, each vertex is labeled by a real-valued payoff for player 1, whose objective is to maximize

the limit-average of all (infinitely many) payoffs in a play. In a *mean-payoff parity* game, player 1 obtains the mean-payoff value only if the play satisfies the parity condition; otherwise her reward is $-\infty$. The reward for player 2 is dual, as the game is zero-sum. To our knowledge, such a combination of general qualitative (parity) and general quantitative (mean-payoff) objectives has not been studied before.

Interestingly, the combination of mean-payoff and parity objectives results in games whose properties differ from both mean-payoff games and parity games. In particular, player-1 optimal strategies in general require infinite memory. To see this, consider the following example.

Example 1 Consider the mean-payoff parity game shown in Fig 1. The game consist of three vertices v_0 , v_1 , and v_2 . The priority function is as follows: $p(v_0) = 1$, $p(v_1) = 1$, and $p(v_2) = 0$, and the parity objective of player 1 is to ensure that the minimum priority that is visited infinitely often is even. The reward function is as follows: $r(v_0) = 10$, $r(v_1) = 10$, and $r(v_2) = 2$. All three vertices are player-1 vertices. A memoryless strategy for player 1 chooses a successor at each vertex independent of the sequence of vertices visited previously in the history of the play. Consider the two possible memoryless strategies of player 1: (a) if at v_0 the strategy chooses successor v_1 , then the minimum priority visited infinitely often is odd, and therefore player 1's reward is $-\infty$; (b) if at v_0 the strategy chooses successor v_2 instead, then player 1's reward is the mean payoff 6.

Now consider the following (not memoryless) strategy for player 1: the strategy chooses $v_0 \rightarrow v_1$ for k times followed by choosing $v_0 \rightarrow v_2$ once, and this sequence of $k+1$ choices is repeated forever. The resulting play ω visits v_2 infinitely often; hence the minimum priority visited infinitely often is even, and the reward of player 1 is the mean-payoff value of ω . For all $\varepsilon > 0$, it is possible to choose large k , depending on ε , such that player 1 ensures that she gets a reward of at least $10 - \varepsilon$. Finally, consider the following strategy σ for player 1: the strategy is played in rounds; in round $i \geq 0$, the strategy σ plays $v_0 \rightarrow v_1$ for i times, then plays $v_0 \rightarrow v_2$ once and progresses to round $i + 1$. The strategy σ ensures player 1 a limit-average reward of value 10 and is an optimal strategy.

However, there is no finite-memory optimal strategy for player 1 in this game. Consider any finite-memory strategy such that the size of the memory M is k . The synchronous product of the game graph with the memory M has $3k$ vertices. In the product graph consider any memoryless strategy σ' that satisfies the parity objective. The mean-payoff value of a play given the strategy σ' is at most $\frac{(3k-1) \cdot 10 + 2}{3k} = 10 + \frac{8}{3k}$. Hence, for any finite k , there is an ε with $0 < \varepsilon < \frac{8}{3k}$ such that no finite-memory strategy with memory of size k can ensure a reward that lies within ε of the optimal value. Hence, achieving the value of

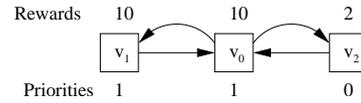


Figure 1. A mean-payoff parity game.

a mean-payoff parity game with arbitrary precision in general requires infinite-memory strategies. ■

We prove that, as in the above example, infinite-memory optimal player-1 strategies exist for all mean-payoff parity games. However, as the above example shows, an optimal player-1 strategy may require infinite memory. However, for player 2 finite-memory optimal strategies exist for all mean-payoff parity games. The results are in sharp contrast to parity games as well as to mean-payoff games, where memoryless optimal strategies exist for both players [11, 10]. While in the case of more general class of *concurrent* games (where the players choose their moves simultaneously) infinite-memory strategies are required even for simple qualitative objectives such as Büchi [7], in our setting of perfect-information games the requirement of infinite memory arises due to the combination of quantitative and qualitative objectives.

Our main result is an algorithm for solving mean-payoff parity games. The running time of the algorithm is $O(n^d \cdot (m + \text{MP} + \text{Parity}))$, where n and m denote the number of vertices and edges of the game graph, respectively, d is the number of priorities of the parity objective, and MP and Parity denote the complexities required for solving mean-payoff games and parity games, respectively. Our algorithm for combined mean-payoff and parity objectives recursively solves smaller subgames based on the following insight. While in general infinite-memory strategies are required, the infinite-memory strategies are well-structured. Informally, an optimal strategy follows a mean-payoff strategy for a sufficiently long time, followed by a finite strategy to ensure that the parity objective is not violated; and this alternation of mean-payoff and parity strategies is repeated infinitely often. The longer the mean-payoff strategies are played, the closer the resulting reward will be to the optimal value. This observation allows us to identify the least and greatest value classes of a mean-payoff parity game by a recursive decomposition. The corresponding algorithms are described in Section 3. Using these algorithms recursively, we compute the values of a mean-payoff parity game in Section 4. As a corollary, it follows that the values are always rational (provided all rewards are rational).

2. Mean-Payoff Parity Games

Notation. For $d \in \mathbb{N}$, by $[d]$ we denote the set $\{0, 1, 2, \dots, d - 1\}$, and by $[d]^+$ we denote the set

$\{1, 2, \dots, d\}$. If (V, E) is a directed graph and $W \subseteq V$, then by $(V, E) \upharpoonright W$ we denote the subgraph (W, F) of (V, E) , where $F = E \cap (W \times W)$.

Definition 1 (Parity, mean-payoff, and mean-payoff parity games). A *game graph* $\mathcal{G} = ((V, E), (V_1, V_2))$ consists of a directed graph (V, E) and a partition (V_1, V_2) of the set V of vertices. All game graphs have the property that every vertex has at least one out-going edge. A *parity game* $\mathcal{P} = (\mathcal{G}, p)$ consists of a game graph \mathcal{G} and a priority function $p: V \rightarrow [d]$ or $p: V \rightarrow [d]^+$, for some $d \in \mathbb{N}$, which maps every vertex to a natural-number-valued priority. A *mean-payoff game* $\mathcal{M} = (\mathcal{G}, r)$ consists of a game graph \mathcal{G} and a reward function $r: V \rightarrow \mathbb{R}$, which maps every vertex to a real-valued reward. A *mean-payoff parity game* $\mathcal{MP} = (\mathcal{G}, p, r)$ consists of a game graph \mathcal{G} , a priority function $p: V \rightarrow [d]$ or $p: V \rightarrow [d]^+$, and a reward function $r: V \rightarrow \mathbb{R}$. We denote by n and m the number of vertices and edges, respectively, i.e., $n = |V|$ and $m = |E|$. ■

Special cases of mean-payoff parity games are mean-payoff *Büchi* and mean-payoff *coBüchi* games, where the priority function p is defined as $p: V \rightarrow [2]$ and $p: V \rightarrow [2]^+$, respectively. A game is played by two players: player 1 and player 2, who form an infinite path in the game graph by moving a token along edges. They start by placing the token on an initial vertex and then they take moves indefinitely in the following way. If the token is on a vertex in V_1 , then player 1 moves the token along one of the edges going out of the vertex. If the token is on a vertex in V_2 , then player 2 does likewise. The result is an infinite path $\omega = \langle v_1, v_2, v_3, \dots \rangle$ in the game graph; we refer to such infinite paths as *plays*. The value of a play is determined by the priorities and rewards of the vertices in the play.

Parity payoff. For a play $\omega = \langle v_1, v_2, v_3, \dots \rangle$ let $\text{Inf}(\omega)$ denote the set of priorities that occur infinitely often in $\langle p(v_1), p(v_2), p(v_3), \dots \rangle$. In a parity game, the play ω is *parity winning* for player 1 if $\min(\text{Inf}(\omega))$ is even; otherwise it is parity winning for player 2.

Mean payoff. Informally, in a mean-payoff game player 1 wins and player 2 loses a payoff that is the “long-run average” of the rewards of the play. Formally, the payoffs for players 1 and 2 of the play $\omega = \langle v_1, v_2, v_3, \dots \rangle$ are $M_1(\omega)$ and $M_2(\omega)$, respectively, defined as follows: $M_1(\omega) = \liminf_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n r(v_i)$ and $M_2(\omega) = -M_1(\omega)$. The objective of each player is to maximize her own payoff.

Mean payoff and parity. Informally, in a mean-payoff parity game player 1 wins the “long-run average” reward if the parity objective is satisfied, else player 1 is penalized infinitely. Intuitively this captures the notion that player 1 has a mean-payoff objective under a parity constraint, and player 1 wins her “long-run average” reward only if she satisfies her parity objective. Formally, given the play ω , the

values $MP_1(\omega)$ and $MP_2(\omega)$ for both players are defined as follows:

$$MP_1(\omega) = \begin{cases} M_1(\omega) & \text{if } \min(\text{Inf}(\omega)) \\ & \text{is even} \\ -\infty & \text{otherwise} \end{cases}$$

and $M_2(\omega) = -M_1(\omega)$. Note that if the play ω is parity winning for player 1, then the value of the play is finite for both players, whereas if the play ω is parity winning for player 2, then the value of the play is $-\infty$ for player 1 and $+\infty$ for player 2.

Given $k \in \mathbb{N} \setminus \{0\}$, and a play $\omega = \langle v_1, v_2, v_3, \dots \rangle$, we denote by ω^k the sequence $\langle v_k, v_{k+1}, \dots \rangle$ of vertices, i.e., the play ω without the prefix of length $k-1$. Note that mean payoff and parity are infinitary objectives and the value of a play ω remains unaffected if a finite prefix is deleted. It follows that for every play ω and every $k \in \mathbb{N} \setminus \{0\}$, we have $MP_1(\omega) = MP_1(\omega^k)$ and $MP_2(\omega) = MP_2(\omega^k)$. This observation will be used in the later sections.

Definition 2 (Strategies). A *strategy* for player 1 maps each finite prefix of a play which ends in V_1 to a successor vertex, which player 1 chooses to extend the play if she follows the strategy. Formally, a strategy for player 1 is a function $\sigma: V^* \cdot V_1 \rightarrow V$ such that $(v, \sigma(x \cdot v)) \in E$ for all $x \in V^*$ and $v \in V_1$. The player-2 strategies are defined analogously. We denote by Σ and Π the sets of all strategies for players 1 and 2, respectively. A strategy is *memoryless* if the strategy is independent of the history of the play. Hence, a player-1 *memoryless* strategy can be expressed as function $\sigma: V_1 \rightarrow V$. Let \mathbb{M} be a set called *memory*. A player-1 strategy σ with memory can be described as a pair of functions: (a) a memory update function $\sigma_u: V \times \mathbb{M} \rightarrow \mathbb{M}$, and (b) a next-move function $\sigma_m: V_1 \times \mathbb{M} \rightarrow V$. The strategy $\sigma = (\sigma_u, \sigma_m)$ is *finite memory* if the memory \mathbb{M} is finite; otherwise it is *infinite memory*. ■

A play $\omega = \langle v_1, v_2, v_3, \dots \rangle$ is *consistent* with a strategy σ for player 1 if $v_{\ell+1} = \sigma(\langle v_1, v_2, \dots, v_\ell \rangle)$ for all $\ell \in \mathbb{N}$ with $v_\ell \in V_1$. A strategy σ is *parity winning* for player 1 from a set $U \subseteq V$ of vertices if every play starting from a vertex in U and consistent with σ is parity winning for player 1. Symmetric definitions apply to player 2. Given a strategy σ for player 1, a strategy π for player 2, and a starting vertex $v \in V$, there is a unique play, denoted $\omega_{\sigma, \pi}(v)$, which starts at v and is consistent with both σ and π .

Definition 3 (Values; optimal and ε -optimal strategies). The *value* of players 1 and 2 at a vertex $v \in V$ in a mean-payoff parity game are denoted $MP_1(v)$ and $MP_2(v)$, respectively, defined as follows:

$$MP_1(v) = \sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} MP_1(\omega_{\sigma, \pi}(v));$$

$$MP_2(v) = \sup_{\pi \in \Pi} \inf_{\sigma \in \Sigma} MP_2(\omega_{\sigma, \pi}(v)).$$

For a real-valued constant $\varepsilon \geq 0$, a player-1 strategy σ is ε -optimal if $\inf_{\pi \in \Pi} MP_1(\omega_{\sigma, \pi}(v)) \geq MP_1(v) - \varepsilon$ for all $v \in V$, that is, the strategy achieves a payoff that lies within ε of the value, against all player-2 strategies. A player-1 strategy σ is *optimal* if it is ε -optimal for $\varepsilon = 0$. The optimal strategies and ε -optimal strategies for player 2 are defined symmetrically.

In a mean-payoff game, the value functions $M_1(v)$ and $M_2(v)$ are obtained by replacing MP_1 and MP_2 with M_1 and M_2 , respectively. The optimal strategies are defined accordingly. ■

Theorem 1 (Memoryless determinacy of parity and of mean-payoff games).

1. [11, 16] For every parity game \mathcal{P} , there is a unique partition (W_1, W_2) of the set of vertices such that there is a memoryless parity-winning strategy for player 1 from W_1 , and a memoryless parity-winning strategy for player 2 from W_2 .
2. [10] For every mean-payoff game \mathcal{M} , memoryless optimal strategies exist for both players 1 and 2. ■

We call the sets W_1 and W_2 the *parity-winning sets* of players 1 and 2, respectively. A game is called *parity winning* for player 1 if every vertex is parity winning for player 1, that is, $W_2 = \emptyset$.

Theorem 2 (Determinacy of mean-payoff parity games). For every mean-payoff parity game \mathcal{MP} and every vertex v , either $MP_1(v) = -\infty$ and $MP_2(v) = \infty$; or $MP_1(v) + MP_2(v) = 0$.

Proof. The result follows from the fact that the mean-payoff parity condition is Borel measurable, and from the determinacy of Borel games [15]. ■

Note that the determinacy theorem implies the existence of ε -optimal strategies in mean-payoff parity games, for every $\varepsilon > 0$. We will show that in mean-payoff parity games optimal strategies exist, and that optimal strategies in general require infinite memory (recall Example 1). To solve a mean-payoff parity game is to determine $MP_1(v)$ for every vertex v of the game. The value $MP_1(v)$ is $-\infty$ for every vertex $v \in W_2$ and it is finite for every vertex $v \in W_1$. Hence by computing the sets W_1 and W_2 , the value $MP_1(v)$ is known for every vertex $v \in W_2$. The sets W_1 and W_2 can be computed by any algorithm to solve parity games. For the remainder of the paper we focus on computing the value $MP_1(v)$ for vertices $v \in W_1$.

3. Closed Subsets of Least and Greatest Value Classes

In this section we present algorithms to identify *closed* subsets of vertices in the least and greatest value classes of a mean-payoff parity game. In the next section we will present an algorithm for solving mean-payoff parity games using the algorithms of this section as subroutines.

Given a mean-payoff parity game \mathcal{MP} , let ℓ be the least finite value for player 1 in the game \mathcal{MP} , i.e., $\ell = \min\{MP_1(v) : v \in W_1\}$. We denote by \mathcal{LV} the *least value class* of \mathcal{MP} , i.e., $\mathcal{LV} = \{v \in W_1 : MP_1(v) = \ell\}$. Similarly, we denote by g the greatest value for player 1 in the game \mathcal{MP} , i.e., $g = \max\{MP_1(v) : v \in W_1\}$, and we denote by \mathcal{GV} the *greatest value class* of \mathcal{MP} , i.e., $\mathcal{GV} = \{v \in W_1 : MP_1(v) = g\}$. A subset $U \subseteq V$ of vertices is *1-closed* if for all vertices $u \in U \cap V_1$, if $(u, w) \in E$, then $w \in U$. In other words, every successor of a player-1 vertex in U belongs again to U . Similarly, a subset $U \subseteq V$ of vertices is *2-closed* if for all vertices $u \in U \cap V_2$, if $(u, w) \in E$, then $w \in U$. We now define several terms that are useful to understand the algorithms.

Definition 4 (Attractor sets). We inductively define the set $Attr_1(T, \mathcal{G})$ of vertices from which player 1 has a strategy to reach a given set $T \subseteq V$ of vertices in a game graph \mathcal{G} . Set $R_0 = T$, and for $k \geq 0$, set

$$R_{k+1} = R_k \cup \left\{ \begin{array}{l} v \in V_1 : (v, u) \in E \text{ for some } u \in R_k \\ v \in V_2 : u \in R_k \text{ for all } (v, u) \in E \end{array} \right\}.$$

Let $Attr_1(T, \mathcal{G}) = \bigcup_k R_k$. The set $Attr_2(T, \mathcal{G})$ is defined symmetrically. ■

Definition 5 (Universal-reach sets). We inductively define the set $UnivReach(T, \mathcal{G})$ of vertices from which every path in a game graph \mathcal{G} reaches a given set $T \subseteq V$ of vertices. Set $U_0 = T$, and for $k \geq 0$, set

$$U_{k+1} = U_k \cup \{v \in V : u \in U_k \text{ for all } (v, u) \in E\}.$$

Let $UnivReach(T, \mathcal{G}) = \bigcup_k U_k$. ■

Proposition 1 Let \mathcal{G} be a game graph, and let $T \subseteq V$ be a set of vertices of \mathcal{G} . Then the complement $\mathcal{H} = \mathcal{G} \upharpoonright (V \setminus Attr_1(T, \mathcal{G}))$ of the attractor set and the complement $\mathcal{H}' = \mathcal{G} \upharpoonright (V \setminus UnivReach(T, \mathcal{G}))$ of the universal-reach set are again game graphs; that is, there is an outgoing edge for every vertex in \mathcal{H} and in \mathcal{H}' . ■

Consider a mean-payoff parity game \mathcal{MP} such that the game is parity winning for player 1. We assume without loss of generality that $p^{-1}(0) \cup p^{-1}(1) \neq \emptyset$, as otherwise, it would be possible to reduce every priority by 2 and continue. In Subsection 3.1 we will present a recursive algorithm to identify a 1-closed subset of \mathcal{LV} where $p^{-1}(0) \neq \emptyset$.

In Subsection 3.2 we will present a recursive algorithm to identify a 2-closed subset of \mathcal{GV} where $p^{-1}(0) = \emptyset$ and $p^{-1}(1) \neq \emptyset$.

3.1 A 1-closed subset of \mathcal{LV}

Consider a mean-payoff parity game $\mathcal{MP} = (\mathcal{G}, p, r)$, where $\mathcal{G} = ((V, E), (V_1, V_2))$, such that the game is parity winning for player 1 and $p^{-1}(0) \neq \emptyset$. Let $F = \text{Attr}_1(p^{-1}(0), \mathcal{G})$ and $H = V \setminus F$. Consider the subgame \mathcal{H} induced by the vertex set H , i.e., $\mathcal{H} = \mathcal{G} \upharpoonright H$. In sequel, we often use game graphs as superscripts of value functions to make the context of the game clear. For a vertex $v \in H$, we denote by $MP_1^{\mathcal{H}}(v)$ the mean-payoff parity game value in the subgame \mathcal{H} . We construct a mean-payoff game $\tilde{\mathcal{M}} = (\tilde{\mathcal{G}}, \tilde{r})$, where $\tilde{\mathcal{G}} = ((\tilde{V}, \tilde{E}), (\tilde{V}_1, \tilde{V}_2))$, from the mean-payoff parity game \mathcal{MP} as follows:

- $\tilde{V} = V \cup \{\tilde{v} : v \in H\}$.
- $\tilde{V}_1 = V_1 \cup (\tilde{V} \setminus V)$ and $\tilde{V}_2 = \tilde{V} \setminus \tilde{V}_1$.
- $\tilde{E} = E \cup \{(v, \tilde{v}) : v \in H\} \cup \{(\tilde{v}, \tilde{v}) : v \in H\}$.
- $\tilde{r}(v) = r(v)$ for $v \in V$, and $\tilde{r}(\tilde{v}) = MP_1^{\mathcal{H}}(v)$ for $v \in H$.

Lemma 1 For every vertex $v \in H$, $M_1^{\tilde{\mathcal{G}}}(\tilde{v}) \leq MP_1^{\mathcal{H}}(v)$.

Proof. For every vertex $v \in H \cap V_2$, player 2 can choose an edge (v, \tilde{v}) such that $M_1^{\tilde{\mathcal{G}}}(\tilde{v}) = MP_1^{\mathcal{H}}(v)$. The desired result is an easy consequence. ■

Let \tilde{l} be the value of the least value class in the game $\tilde{\mathcal{G}}$, and let \hat{l} be the value of the least value class in the game \mathcal{H} . It follows from Lemma 1 that $\hat{l} \geq \tilde{l}$. We denote by $\mathcal{LV}_{\tilde{\mathcal{G}}}$ the least value class in the game graph $\tilde{\mathcal{G}}$. Let $\tilde{n} = |\tilde{V}|$.

Lemma 2 There is an optimal strategy $\tilde{\sigma}$ for player 1 in $\tilde{\mathcal{G}}$ such that for every $k > 0$, for every strategy π of player 2 in $\tilde{\mathcal{G}}$, for every vertex v , if $\omega_{\tilde{\sigma}, \pi}(v) = \langle v_1, v_2, v_3, \dots \rangle$, then $\sum_{i=1}^k r(v_i) \geq (k - \tilde{n}) \cdot \tilde{l}$.

Proof. Any memoryless optimal strategy in $\tilde{\mathcal{G}}$ satisfies the desired condition. Theorem 1 [10] guarantees existence of memoryless optimal strategies in mean-payoff games. ■

Lemma 3 For all $\varepsilon > 0$, there is a strategy σ^* for player 1 in the game \mathcal{G} such that for every vertex $v \in \mathcal{LV}_{\tilde{\mathcal{G}}} \cap V$, we have $\inf_{\pi \in \Pi} MP_1^{\mathcal{G}}(\omega_{\sigma^*, \pi}(v)) \geq \tilde{l} - \varepsilon$.

Proof. For $\varepsilon > 0$, we construct the desired strategy σ^* as follows: the strategy σ^* is played in rounds numbered

1, 2, 3, ... Let $a_0 = 0$; then in round i the strategy is defined as follows:

$$\sigma^* = \begin{cases} \text{Phase 1.} & \text{Play } \tilde{\sigma} \text{ of Lemma 2 for } (a_{i-1} + 2\tilde{n}) \cdot \tilde{l} \cdot 2^i \\ & \text{steps, where } a_{i-1} \text{ is the number of steps} \\ & \text{the play was in Phase 2 in round } i-1. \\ \text{Phase 2.} & \text{While the game is in } H, \text{ play } \hat{\sigma} \\ & \text{(an } \varepsilon\text{-optimal strategy in } \mathcal{H}\text{).} \\ \text{Phase 3.} & \text{When the game reaches } F, \text{ play a} \\ & \text{strategy to reach } p^{-1}(0) \text{ and goto} \\ & \text{Phase 1 of round } i+1. \end{cases}$$

Consider any strategy π for player 2. We consider two cases to prove the desired result.

1. Case 1: If $\text{Inf}(\omega_{\sigma^*, \pi}(v)) \cap F = \emptyset$ (i.e., the play visits F finitely often), then the game stays in H forever after a finite number of steps. Thus the strategy σ^* behaves like $\hat{\sigma}$ after a finite number of steps, and the value of the play is also determined by the infinite suffix. Hence

$$\begin{aligned} MP_1^{\mathcal{G}}(\omega_{\sigma^*, \pi}(v)) &\geq \min_{v \in H} MP_1^{\mathcal{H}}(v) - \varepsilon \\ &= \hat{l} - \varepsilon \geq \tilde{l} - \varepsilon, \end{aligned}$$

by ε -optimality of $\hat{\sigma}$. Observe that if $\hat{\sigma}$ is optimal in \mathcal{H} , then the inequality holds with $\varepsilon = 0$.

2. Case 2: Else $\text{Inf}(\omega_{\sigma^*, \pi}(v)) \cap F \neq \emptyset$. Then the Phase 3 of strategy σ^* ensures that $\text{Inf}(\omega_{\sigma^*, \pi}(v)) \cap p^{-1}(0) \neq \emptyset$. Hence $\min(\text{Inf}(\omega_{\sigma^*, \pi}(v)))$ is even, and the play is parity winning for player 1. Consider the segment of play starting at Phase 2 of round i and ending at Phase 1 of round $i+1$. We denote by b_{i+1} the number of steps that Phase 1 is played in round $i+1$. From the properties of the strategies in Phase 1 and Phase 2 (from Lemma 2) a lower bound on the average reward for the considered segment of the play can be derived as follows:

$$\begin{aligned} \frac{(b_{i+1} - \tilde{n}) \cdot \tilde{l}}{b_{i+1} + a_i + \tilde{n}} &= \tilde{l} - \frac{(a_i + 2\tilde{n}) \cdot \tilde{l}}{b_{i+1} + a_i + \tilde{n}} \\ &\geq \tilde{l} - \frac{(a_i + 2\tilde{n}) \cdot \tilde{l}}{b_{i+1}} = \tilde{l} - \frac{1}{2^i}, \end{aligned}$$

because $b_{i+1} = (a_i + 2\tilde{n}) \cdot \tilde{l} \cdot 2^i$. Hence for all strategies π , $MP_1(\omega_{\sigma^*, \pi}(v)) \geq \lim_{i \rightarrow \infty} (\tilde{l} - \sum_{j=i}^{\infty} \frac{1}{2^j}) = \tilde{l}$. This gives us the desired inequality. ■

Lemma 4 For all $\varepsilon > 0$, there is a strategy π^* for player 2 in the game \mathcal{G} such that for every vertex $v \in \mathcal{LV}_{\tilde{\mathcal{G}}} \cap V$, we have $\sup_{\sigma \in \Sigma} MP_1^{\mathcal{G}}(\omega_{\sigma, \pi^*}(v)) \leq \tilde{l} + \varepsilon$.

Proof. For $\varepsilon > 0$, we construct the desired strategy π^* as follows: for a finite sequence $\vec{w} = \langle v_1, v_2, \dots, v_k \rangle$ of

vertices and $v \in V_2$, the strategy π^* is defined as follows:

$$\pi^*(\vec{w} \cdot v) = \begin{cases} \tilde{\pi}(\vec{w} \cdot v) & \text{if for all } 1 \leq i \leq k \text{ if } v_i \in V_2, \\ & \text{then } \tilde{\pi}(\langle v_1, v_2, \dots, v_i \rangle) \in V, \\ & (\tilde{\pi} \text{ is an optimal strategy in } \tilde{\mathcal{G}}). \\ \hat{\pi}(\vec{w} \cdot v) & \text{otherwise; } (\hat{\pi} \text{ is } \varepsilon\text{-optimal in } \mathcal{H}). \end{cases}$$

Informally, the strategy π^* follows an optimal strategy $\tilde{\pi}$ in $\tilde{\mathcal{G}}$ as long as $\tilde{\pi}$ does not choose an edge in $\tilde{V} \setminus V$; if this happens, it switches to an ε -optimal strategy $\hat{\pi}$ in the game \mathcal{H} . Consider any strategy σ for player 1.

1. If $\text{Inf}(\omega_{\sigma, \tilde{\pi}}(v)) \subseteq V$ in the game $\tilde{\mathcal{G}}$, then the play $\omega_{\sigma, \pi^*}(v)$ in the game graph \mathcal{G} is same as the play $\omega_{\sigma, \tilde{\pi}}(v)$ in the game graph $\tilde{\mathcal{G}}$. Since $\tilde{\pi}$ is an optimal strategy in the mean-payoff game $\tilde{\mathcal{G}}$, we have $MP_1^{\mathcal{G}}(\omega_{\sigma, \pi^*}(v)) = M_1^{\tilde{\mathcal{G}}}(\omega_{\sigma, \tilde{\pi}}(v)) \leq \tilde{l}$.
2. Else $\omega_{\sigma, \tilde{\pi}}(v) = \langle v_1, v_2, \dots \rangle$ reaches a vertex \tilde{v}_ℓ in the game $\tilde{\mathcal{G}}$. Let j be the least index such that $v_j = \tilde{v}_\ell$. Since $\tilde{\pi}$ is an optimal strategy in $\tilde{\mathcal{G}}$ and $v \in \mathcal{LV}_{\tilde{\mathcal{G}}}$, it follows that $MP_1^{\mathcal{H}}(\tilde{v}_\ell) = \tilde{l}$. The strategy π^* then switches to the ε -optimal strategy $\hat{\pi}$ of the game \mathcal{H} . The following inequality is a consequence of ε -optimality of $\hat{\pi}$ in \mathcal{H} :

$$\begin{aligned} MP_1^{\mathcal{G}}(\omega_{\sigma, \pi^*}(v)) &= MP_1^{\mathcal{G}}(\omega_{\sigma, \pi^*}^j(v)) = MP_1^{\mathcal{H}}(\omega_{\sigma, \hat{\pi}}^j(v)) \\ &\leq MP_1^{\mathcal{H}}(\tilde{v}_\ell) + \varepsilon = \tilde{l} + \varepsilon. \end{aligned}$$

The lemma follows from the above inequalities. \blacksquare

Proposition 2 *The set $\mathcal{LV}_{\tilde{\mathcal{G}}} \cap V$ of least values is 1-closed.*

Observe that if the strategies $\hat{\sigma}$ and $\hat{\pi}$ in Lemma 3 and Lemma 4 are optimal in the sub-game \mathcal{H} , then the strategies σ^* and π^* are optimal for all vertices in $\mathcal{LV}_{\tilde{\mathcal{G}}} \cap V$ in the game \mathcal{G} . Also note that the strategy π^* is finite-memory if $\hat{\pi}$ is finite-memory, whereas the strategy σ^* requires infinite-memory to count the steps in the Phase 1 of the rounds. The following Lemma follows from Lemma 3, Lemma 4, and Proposition 2.

Lemma 5 *Let $\mathcal{MP} = (\mathcal{G}, p, r)$ be a mean-payoff parity game such that $p^{-1}(0) \neq \emptyset$ and the game is parity winning for player 1. Then Algorithm 1 identifies a 1-closed subset of the least value class of \mathcal{MP} .*

3.2 A 2-closed subset of \mathcal{GV}

Consider a mean-payoff parity game $\mathcal{MP} = (\mathcal{G}, p, r)$ such that the game is parity winning for player 1 and $p^{-1}(0) = \emptyset$ and $p^{-1}(1) \neq \emptyset$. We present an algorithm to identify a 2-closed subset of the greatest value class in \mathcal{MP} . The algorithm is different from Algorithm 1, as

in the present case the least priority of a vertex is 1, which is odd. Let $F = \text{Attr}_2(p^{-1}(1), \mathcal{G})$ and $H = V \setminus F$. Consider the subgame \mathcal{H} induced by the vertex set H , i.e., $\mathcal{H} = \mathcal{G} \upharpoonright H$. Let \hat{g} be the value of the greatest value class in the game \mathcal{H} .

Lemma 6 *For every vertex $v \in \mathcal{GV}_{\mathcal{H}}$, $MP_1^{\mathcal{G}}(v) = MP_1^{\mathcal{H}}(v)$.*

Proof. Let $\hat{\sigma}$ be an ε -optimal strategy for player 1 in the sub-game \mathcal{H} . For every strategy π of player 2 and every vertex $v \in H$, we have $MP_1^{\mathcal{G}}(\omega_{\hat{\sigma}, \pi}(v)) \geq MP_1^{\mathcal{H}}(v) - \varepsilon$. Hence $MP_1^{\mathcal{G}}(v) \geq MP_1^{\mathcal{H}}(v)$, for every vertex $v \in H$. Observe that if $\hat{\sigma}$ is optimal in \mathcal{H} , then $\hat{\sigma}$ is optimal for all vertices in $\mathcal{GV}_{\mathcal{H}}$ in the game \mathcal{G} . We construct a strategy π for player 2 as follows:

$$\pi = \begin{cases} \text{in } F \text{ the strategy is a strategy to reach } p^{-1}(1) \\ \text{in } H \text{ the strategy is an } \varepsilon\text{-optimal strategy } \hat{\pi} \text{ in } \mathcal{H} \end{cases}$$

Consider any strategy σ for player 1. There are two cases:

1. If the play $\omega_{\sigma, \pi}(v)$ visits F infinitely often, then by the property of π in F we have $\text{Inf}(\omega_{\sigma, \pi}(v)) \cap p^{-1}(1) \neq \emptyset$. Since $p^{-1}(0) = \emptyset$, the minimum priority occurring infinitely often is odd, and hence the play is not parity winning for player 1.
2. Else the play $\omega_{\sigma, \pi}(v)$ visits F only finitely often. Choose a finite integer k such that every vertex in the path $\omega_{\sigma, \pi}^k(v)$ is a vertex in H . Then

$$\begin{aligned} MP_1^{\mathcal{G}}(\omega_{\sigma, \pi}(v)) &= MP_1^{\mathcal{G}}(\omega_{\sigma, \pi}^k(v)) \\ &= MP_1^{\mathcal{H}}(\omega_{\sigma, \hat{\pi}}^k(v)) \leq \hat{g} + \varepsilon, \end{aligned}$$

by ε -optimality of $\hat{\pi}$. Observe that if $\hat{\pi}$ is optimal in \mathcal{H} , then the strategy π is optimal for all vertices in $\mathcal{GV}_{\mathcal{H}}$ in the game \mathcal{G} . The strategy π is finite-memory if $\hat{\pi}$ is finite-memory.

It follows that for every vertex $v \in \mathcal{GV}_{\mathcal{H}}$, we have $MP_1^{\mathcal{G}}(v) = MP_1^{\mathcal{H}}(v)$. \blacksquare

Proposition 3 *The set $\mathcal{GV}_{\mathcal{H}}$ of greatest values is 2-closed.*

Lemma 6 and Proposition 3 gives us the next Lemma.

Lemma 7 *Let $\mathcal{MP} = (\mathcal{G}, p, r)$ be a mean-payoff parity game such that $p^{-1}(0) = \emptyset$ and $p^{-1}(1) \neq \emptyset$, and the game is parity winning for player 1. Then Algorithm 2 identifies a 2-closed subset of the greatest value class of \mathcal{MP} .*

4. Solving Mean-Payoff Parity Games

In this section we present an algorithm to compute the values of a mean-payoff parity game. We first give an informal description of the algorithm; the detailed description is shown in Algorithm 3.

Algorithm 1 ComputeLeastValueClass

Input: a mean-payoff parity game $\mathcal{MP} = (\mathcal{G}, p, r)$ such that $p^{-1}(0) \neq \emptyset$ and the game is parity winning for player 1.

Output: a nonempty 1-closed subset of \mathcal{LV} , and $MP_1(v)$ for all $v \in \mathcal{LV}$.

1. $F = Attr_1(p^{-1}(0), \mathcal{G})$.
 2. $H = V \setminus F$ and $\mathcal{H} = \mathcal{G} \upharpoonright H$.
 3. **MeanPayoffParitySolve**(\mathcal{H}) (Algorithm 3).
 4. Construct the mean-payoff game $\tilde{\mathcal{G}}$ as described in Subsection 3.1 and **Solve** the mean-payoff game $\tilde{\mathcal{G}}$.
 5. Let $\mathcal{LV}_{\tilde{\mathcal{G}}}$ be the least value class in $\tilde{\mathcal{G}}$ and \tilde{l} be the least value.
 6. $\mathcal{LV} = \mathcal{LV}_{\tilde{\mathcal{G}}} \cap V$, and $MP_1(v) = \tilde{l}$ for all $v \in \mathcal{LV}$.
 7. **return** $(\mathcal{LV}, \tilde{l})$.
-

Algorithm 2 ComputeGreatestValueClass

Input: a mean-payoff parity game $\mathcal{MP} = (\mathcal{G}, p, r)$ such that $p^{-1}(0) = \emptyset$ and $p^{-1}(1) \neq \emptyset$, and the game is parity winning for player 1.

Output: a nonempty 2-closed subset \mathcal{GV} , and $MP_1(v)$ for all $v \in \mathcal{GV}$.

1. $F = Attr_2(p^{-1}(1), \mathcal{G})$.
 2. $H = V \setminus F$ and $\mathcal{H} = \mathcal{G} \upharpoonright H$.
 3. **MeanPayoffParitySolve**(\mathcal{H}) (Algorithm 3).
 4. Let $\mathcal{GV}_{\mathcal{H}}$ be the greatest value class in \mathcal{H} and \hat{g} be the greatest value.
 5. $\mathcal{GV} = \mathcal{GV}_{\mathcal{H}}$, and $MP_1(v) = \hat{g}$ for all $v \in \mathcal{GV}$.
 6. **return** (\mathcal{GV}, \hat{g}) .
-

Informal description of Algorithm 3. The algorithm iteratively computes the values of vertices of a mean-payoff parity game. We focus on computing the values in the parity winning set W_1 for player 1. Throughout the computation, the vertices whose values have been determined are denoted by the set W_0 .

Step 6.2.a. If at any iteration i of Algorithm 3 the subgame induced by the vertex set $W_1 \setminus W_0$ is not parity winning for player 1, then let W_2^i be the nonempty parity winning set for player 2. Let w be a vertex with the maximum value for player 1 in W_0 to which a vertex $v \in W_2^i \cap V_1$ has an edge. For every vertex $v \in W_2^i \cap V_1$ which has an edge to a vertex $w' \in W_0$ that has value equal to w , the value of v is set to the value of w , and v is included in W_0 .

Step 6.2.b. Otherwise, the subgame induced by the vertex set $W_1 \setminus W_0$ is parity winning for player 1. If $p^{-1}(0) \cap (W_1 \setminus W_0) \neq \emptyset$, then the least value class L_i with value l_i is computed using Algorithm 1 (Step 6.2.b.1); else the greatest value class G_i is computed using Algorithm 2 (Step 6.2.b.2). Suppose that there is a vertex $v \in L_i \cap V_1$ such that there is an edge (v, w) such that $w \in W_0$ and the value of w is greater than l_i . Let w be a vertex with the maximum value for player 1 in W_0 to which a vertex $v \in L_i \cap V_1$ has an edge. For every vertex $v \in W_2^i \cap V_1$ which has an edge to a vertex $w' \in W_0$ that has value equal to w , the value of v is set to the value of w , and v is included

in W_0 . A similar procedure is applied to vertices $v \in L_i \cap V_2$ with the max operator replaced by min. Otherwise, for all vertices in L_i the value is set to l_i . A similar procedure is applied to G_i in Step 6.2.b.2. The correctness of the Algorithm is proved in the following lemma. We use the notation $p-2: V \rightarrow [d]$ for the function with $(p-2)(v) = p(v) - 2$ for all vertices $v \in V$.

Lemma 8 *For every mean-payoff parity game \mathcal{MP} , Algorithm 3 correctly computes the value function MP_1 .*

Proof. We prove by induction that for every vertex $v \in W_0$, we have $Val(v) = MP_1(v)$. The base case follows from the correctness of Algorithms 1 and 2 for games with only one priority. To prove the inductive case, we show the following two claims.

Claim 1. Let L_i be the least value class in the subgame \mathcal{G}_i , and let l_i be the least value. Let W_0 be the set of vertices of the i -th iteration. Let $g = \max\{Val(w) : w \in W_0 \text{ and } \exists v \in L_i \cap V_1. (v, w) \in E\}$, and let $l = \min\{Val(w) : w \in W_0 \text{ and } \exists v \in L_i \cap V_2. (v, w) \in E\}$.

1. If $g > l_i$, then for every vertex $v \in T_1$, where $T_1 = \{v \in L_i \cap V_1 : \exists w \in W_0. Val(w) = g \text{ and } (v, w) \in E\}$, we have $MP_1(v) = g$.
2. If $l < l_i$, then for every vertex $v \in T_2$, where $T_2 = \{v \in L_i \cap V_2 : \exists w \in W_0. Val(w) = l \text{ and } (v, w) \in E\}$, we have $MP_1(v) = l$.

Algorithm 3 MeanPayoffParitySolve

Input: a mean-payoff party game \mathcal{MP} . **Output:** the value function MP_1 .

1. Compute W_1 and W_2 by any algorithm for solving parity games.
 2. For every vertex $v \in W_2$, set $Val(v) = -\infty$. 3. $W_0 = \emptyset$. 4. $\mathcal{G}_0 = \mathcal{G} \upharpoonright W_1$ and $V^0 = W_1$. 5. $i = 0$.
 6. **repeat**
 - 6.1. **while** $(p^{-1}(0) \cup p^{-1}(1)) \cap V_i = \emptyset$ **do** set $p = p - 2$. **end while**
 - 6.2. Let (W_1^i, W_2^i) be the partition of the parity winning sets in \mathcal{G}_i .
 - 6.2.a. **if** $W_2^i \neq \emptyset$ **then**
 - 6.2.a.1. $g = \max\{Val(w) : w \in W_0 \text{ and } \exists v \in W_2^i \cap V_1. (v, w) \in E\}$.
 - 6.2.a.2. $T_1 = \{v \in L_i \cap V_1 : \exists w \in W_0. Val(w) = g \text{ and } (v, w) \in E\}$.
 - 6.2.a.3. $W_0 = W_0 \cup UnivReach(T_1)$.
 - 6.2.a.4. For every vertex $v \in UnivReach(T_1)$, set $Val(v) = g$.**goto** Step. 6.3.
 - 6.2.b. **else**
 - 6.2.b.1. **if** $p^{-1}(0) \cap V_i \neq \emptyset$ **then** let $(L_i, l_i) = \mathbf{ComputeLeastValueClass}(\mathcal{G}_i)$.
 - 6.2.b.1.a. Subroutine $SetValues(L_i, l_i)$
 - 6.2.b.1.b. $W_0 = W_0 \cup L_i$, and for every vertex $v \in L_i$, set $Val(v) = l_i$.
 - 6.2.b.2. **else** $(G_i, g_i) = \mathbf{ComputeGreatestValueClass}(\mathcal{G}_i)$.
 - 6.2.b.2.a. Subroutine $SetValues(G_i, g_i)$
 - 6.2.b.2.b. $W_0 = W_0 \cup G_i$, and for every vertex $v \in G_i$, set $Val(v) = g_i$.
 - 6.3. $V^{i+1} = V^i \setminus W_0$ and $\mathcal{G}_i = \mathcal{G} \upharpoonright V^i$.
 - 6.4. $i = i + 1$.
 - until** $V_i = \emptyset$ (**end repeat**)
 7. $MP_1 = Val$.
-

3. If $g \leq l_i$ and $l \geq l_i$, then for every vertex $v \in L_i$, we have $MP_1(v) = l_i$.

Proof of Claim 1:

1. For $\varepsilon > 0$, fix an ε -optimal strategy π for player 2 as follows: play an ε -optimal strategy π^i in the subgame \mathcal{G}_i , and switch to an ε -optimal strategy π_w from the set W_0 once the game reaches W_0 . Consider a vertex $v \in T_1$ and a strategy σ for player 1. If the play $\omega_{\sigma, \pi}(v)$ stays in \mathcal{G}_i forever, then $MP_1(\omega_{\sigma, \pi}(v)) = MP_1(\omega_{\sigma, \pi^i}(v)) \leq l_i + \varepsilon$, by ε -optimality of π^i . By the inductive hypothesis, for all vertices $w \in W_0$, we have $MP_1(w) = Val(w)$. Note that g is the maximum value that player 1 can ensure escaping to any vertex in W_0 . If the play $\omega_{\sigma, \pi^i}(v)$ reaches W_0 , then player 2 follows an ε -optimal strategy π_w to ensure that $MP_1(\omega_{\sigma, \pi}(v)) \leq g + \varepsilon$. Since $l_i \leq g$, for all player-1 strategies σ , we have $MP_1(\omega_{\sigma, \pi}(v)) \leq g + \varepsilon$. On the other hand, from all vertices in T_1 , player 1 can chose a vertex $w \in W_0$ such that $(v, w) \in E$ and $Val(w) = MP_1(w) = g$, and play ε -optimal strategies σ_w from w to ensure that $MP_1(v) \geq g$. Thus for all vertices $v \in T_1$, $MP_1(v) = g$. Observe that if π^i is optimal in \mathcal{G}_i and π_w is optimal from W_0 , then the strategy π is optimal for all vertices $v \in T_1$.
2. The argument is similar to the previous case.
3. For $\varepsilon > 0$, fix an ε -optimal strategy σ for player 1 as

follows: play an ε -optimal strategy σ^i in the subgame \mathcal{G}_i , and switch to an ε -optimal strategy σ_w from the set W_0 once the game reaches W_0 . Consider a vertex $v \in L_i$ and a strategy π for player 2. If the play $\omega_{\sigma, \pi}(v)$ stays in \mathcal{G}_i forever, then ε -optimality of σ^i implies $MP_1(\omega_{\sigma, \pi}(v)) = MP_1(\omega_{\sigma^i, \pi}(v)) \geq l_i - \varepsilon$. Since for all edges (v, w) such that $v \in L_i \cap V_2$ and $w \in W_0$, we have $Val(w) = MP_1(w) \geq l_i$, if the play reaches W_0 , then ε -optimality of σ_w from W_0 implies $MP_1(\omega_{\sigma, \pi}(v)) \geq l_i - \varepsilon$. Hence, for all vertices $v \in L_i$, we have $MP_1(v) \geq l_i$. By a symmetric argument for player 2, it follows that $MP_1(v) \leq l_i$, for all vertices $v \in L_i$. Again if σ^i is optimal in \mathcal{G}_i and σ_w is optimal from W_0 , then σ is optimal from all vertices $v \in L_i$.

Claim 2. Let G_i be the greatest value class in the subgame \mathcal{G}_i , and let g_i be the greatest value. Let W_0 be the set of vertices of the i -th iteration. Let $g = \max\{Val(w) : w \in W_0 \text{ and } \exists v \in G_i \cap V_1. (v, w) \in E\}$, and let $l = \min\{Val(w) : w \in W_0 \text{ and } \exists v \in G_i \cap V_2. (v, w) \in E\}$.

1. If $g > g_i$, then for every vertex $v \in T_1$, where $T_1 = \{v \in G_i \cap V_1 : \exists w \in W_0. Val(w) = g \text{ and } (v, w) \in E\}$, we have $MP_1(v) = g$.
2. If $l < g_i$, then for every vertex $v \in T_2$, where $T_2 = \{v \in G_i \cap V_2 : \exists w \in W_0. Val(w) = l \text{ and } (v, w) \in E\}$, we have $MP_1(v) = l$.

Subroutine *SetValues*(J_i, j_i)

1. $g = \max\{Val(w) : w \in W_0 \text{ and } \exists v \in J_i \cap V_1. (v, w) \in E\}$.
 - 2.1 **if** $g > j_i$ **then**
 - 2.2 $T_1 = \{v \in J_i \cap V_1 : \exists w \in W_0. Val(w) = g \text{ and } (v, w) \in E\}$; and $W_0 = W_0 \cup UnivReach(T_1)$.
 - 2.3 For every vertex $v \in UnivReach(T_1)$, set $Val(v) = g$.
 - 2.4 **goto** Step 6.3. of **MeanPayoffParitySolve**.
 3. $l = \min\{Val(w) : w \in W_0 \text{ and } \exists v \in J_i \cap V_2. (v, w) \in E\}$.
 - 4.1 **if** $l < j_i$ **then**
 - 4.2 $T_2 = \{v \in J_i \cap V_2 : \exists w \in W_0. Val(w) = l \text{ and } (v, w) \in E\}$; and $W_0 = W_0 \cup UnivReach(T_2)$.
 - 4.3 For every vertex $v \in UnivReach(T_2)$, set $Val(v) = l$.
 - 4.4 **goto** Step 6.3. of **MeanPayoffParitySolve**.
-

3. If $g \leq g_i$ and $l \geq g_i$, then for all vertices $v \in G_i$, we have $MP_1(v) = g_i$.

The proof of Claim 2 is similar to the proof of Claim 1. We can now prove the correctness of Algorithm 3.

1. *Correctness of Step 6.2.a.* follows from Claim 1, because for every vertex $v \in W_2^i$, we have $MP_1^{G_i}(v) = -\infty$. Also observe that for every vertex $w \in W_0$, we have $Val(w) = MP_1(w) > -\infty$. Since the game \mathcal{MP} is parity winning for player 1, there is a vertex $v \in W_2^i \cap V_1$ such that there is a vertex $w \in W_0$ and $(v, w) \in E$.
2. *Correctness of Steps 6.2.b.1 and 6.2.b.2* follows from Claims 1 and 2.

The correctness of Algorithm 3 follows. \blacksquare

The following theorem gives the complexity of Algorithm 3. Recall that n is the number of vertices of the game graph, m is the number of edges, d is the number of priorities of the parity objective, and MP and Parity denote the complexities required for solving mean-payoff games and parity games, respectively (by any algorithm).

Theorem 3 (Solution of mean-payoff parity games). *Algorithm 3 solves mean-payoff parity games in time $O(n^d \cdot (m + \text{MP} + \text{Parity}))$.*

Proof. Let $T(n, m, d)$ denote the time complexity of Algorithm 3 to solve a mean-payoff parity game with n vertices, m edges and d priorities. Step 6.2. involves solving a parity game, and Step 6.2.b. recursively solves a smaller sub-game using Algorithm 1 and Algorithm 2. The time complexity of Algorithm 1 and Algorithm 2 are $T(n-1, m-1, d-1) + O(m + \text{MP})$ and $T(n-1, m-1, d-1) + O(m)$, respectively. Thus each iteration of Step 6 can be computed in time $T(n-1, m-1, d-1) + O(m + \text{MP} + \text{Parity})$. In every iteration of Step 6, the value of at least one vertex is fixed. Hence, Step 6 is iterated at most n times. The following recurrence describes the running time of Algorithm 3:

$$T(n, m, d) = n \cdot (T(n-1, m-1, d-1) + O(m + \text{MP} + \text{Parity}))$$

The desired time complexity satisfies the recurrence. \blacksquare

Mean-payoff games can be solved in time $O(n^3 \cdot m \cdot W)$, where W is the input size for the reward function r [21]. Parity games can be solved in time $O(d \cdot m \cdot (\frac{n}{\lfloor d/2 \rfloor})^{\lfloor d/2 \rfloor})$, where d is the number of priorities [13]. Moreover, subexponential-time algorithms are known to solve mean-payoff and parity games [3, 2]. The exact time complexity of mean-payoff parity games remains as open as the complexities of mean-payoff games and of parity games.

Theorem 4 (Rationality of mean-payoff parity games). *Let $\mathcal{MP} = (\mathcal{G}, p, r)$ be a mean-payoff parity game such that the reward $r(v)$ is rational for all vertices $v \in V$. Then the value $MP_1(v)$ is rational for all vertices $v \in W_1$.*

Proof. The proof is by induction on the number of priorities. If there is only one priority, then the value is either $-\infty$ (the priority is odd) and $W_1 = \emptyset$, or else it reduces to a mean-payoff game. The recursive characterization of the value function by Algorithm 3 proves the inductive case. \blacksquare

Theorem 5 (Optimal strategies).

1. *Memoryless optimal strategies exist for player 1 in mean-payoff coBüchi games. Optimal strategies require infinite memory for player 1 in mean-payoff Büchi games.*
2. *Infinite-memory optimal strategies exist for player 1 in mean-payoff parity games. Finite-memory optimal strategies exist for player 2 in mean-payoff parity games.*

Proof. (sketch)

1. In mean-payoff coBüchi games Algorithm 2 reduces to solving a mean-payoff game. It follows from Theorem 1 that memoryless optimal strategies exist in mean-payoff games. The result follows from the recursive characterization of Algorithm 3 and induction on the size of the game graph. The result for the mean-payoff Büchi games follows from Example 1.

2. The result follows by induction on the number of priorities and size of the game graph. The base case consists of games with only one priority. The result then follows from the existence of memoryless optimal strategies in mean-payoff games. The inductive case follows from the recursive characterization of the optimal strategies in mean-payoff parity games from the optimal strategies in sub-games, by Lemma 3, Lemma 4, Lemma 6 and Lemma 8. The infinite-memory requirement of optimal strategies for player 1 follows from the special case of mean-payoff Büchi games. ■

5 Conclusion

We studied perfect-information games with a combination of mean-payoff and parity objectives. The determinacy of these games guarantees the existence of ε -optimal strategies, for all $\varepsilon > 0$. We proved the existence of optimal (not only ε -optimal) strategies for mean-payoff parity games, and our characterization of optimal strategies enabled us to present an algorithm for solving these games. We also showed that unlike in the cases of pure mean-payoff or pure parity games, optimal strategies of the combined games in general require infinite memory. The exact complexity of mean-payoff parity games, and the study of combining other classes of quantitative and qualitative objectives, are directions for future research. Such combinations of objectives are interesting for the synthesis of reactive systems with both fairness (qualitative) and resource (quantitative) constraints [5].

Acknowledgments. This research was supported in part by the ONR grant N00014-02-1-0671, the AFOSR MURI grant F49620-00-1-0327, and the NSF ITR grant CCR-0225610.

References

- [1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *JACM*, 49:672–713, 2002.
- [2] H. Bjorklund, S. Sandberg, and S. Vorobyov. A discrete subexponential algorithms for parity games. In *STACS'03*, pages 663–674. LNCS 2607, Springer, 2003.
- [3] H. Bjorklund, S. Sandberg, and S. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. In *MFCS'04*, pages 673–685. LNCS 3153, Springer, 2004.
- [4] J.R. Büchi and L.H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the AMS*, 138:295–311, 1969.
- [5] A. Chakrabarti, L. de Alfaro, T.A. Henzinger, and M. Stoelinga. Resource interfaces. In *EMSOFT 03: Embedded Software*, pages 117–133. LNCS 2855, Springer, 2003.
- [6] A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.
- [7] L. de Alfaro and T. A. Henzinger. Concurrent ω -regular games. In *LICS'00*, pages 141–154. IEEE Computer Society Press, 2000.
- [8] L. de Alfaro and T.A. Henzinger. Interface theories for component-based design. In *EMSOFT 01: Embedded Software*, pages 148–165. LNCS 2211, Springer, 2001.
- [9] D.L. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-independent Circuits*. The MIT Press, 1989.
- [10] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.
- [11] E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In *FOCS'91*, pages 368–377. IEEE Computer Society Press, 1991.
- [12] T.A. Henzinger, O. Kupferman, and S. Rajamani. Fair simulation. *Information and Computation*, 173:64–81, 2002.
- [13] M. Jurdziński. Small progress measures for solving parity games. In *STACS'00*, pages 290–301. LNCS 1770, Springer, 2000.
- [14] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, 1992.
- [15] D. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.
- [16] A. W. Mostowski. Games with forbidden positions. Technical Report 78, University of Gdańsk, 1991.
- [17] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *POPL'89*, pages 179–190. ACM Press, 1989.
- [18] P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230, 1987.
- [19] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, chapter 7, pages 389–455. Springer, 1997.
- [20] J. Vöge and M. Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *CAV'00*, pages 202–215. LNCS 1855, Springer, 2000.
- [21] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *TCS*, 158:343–359, 1996.