Parity Games with Weights

Sven Schewe[1]
University of Liverpool, Liverpool L69 3BX, United Kingdom
sven.schewe@liverpool.ac.uk

Alexander Weinert[2]
Reactive Systems Group, Saarland University, 66123 Saarbrücken, Germany
weinert@react.uni-saarland.de

Martin Zimmermann
Reactive Systems Group, Saarland University, 66123 Saarbrücken, Germany
zimmermann@react.uni-saarland.de

Abstract

Quantitative extensions of parity games have recently attracted significant interest. These extensions include parity games with energy and payoff conditions as well as finitary parity games and their generalization to parity games with costs. Finitary parity games enjoy a special status among these extensions, as they offer a native combination of the qualitative and quantitative aspects in infinite games: the quantitative aspect of finitary parity games is a quality measure for the qualitative aspect, as it measures the limit superior of the time it takes to answer an odd color by a larger even one. Finitary parity games have been extended to parity games with costs, where each transition is labelled with a non-negative weight that reflects the costs incurred by taking it. We lift this restriction and consider parity games with costs with arbitrary integer weights. We show that solving such games is in NP ∩ co-NP, the signature complexity for games of this type. We also show that the protagonist has finite-state winning strategies, and provide tight exponential bounds for the memory he needs to win the game. Naturally, the antagonist may need infinite memory to win. Finally, we present tight bounds on the quality of winning strategies for the protagonist.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects

Keywords and phrases Infinite Games, Quantitative Games, Parity Games

Digital Object Identifier 10.4230/LIPIcs.CSL.2018.35

Related Version Full version available online [28].

1 Introduction

Finite games of infinite duration offer a wealth of challenges and applications that has garnered to a lot of attention. The traditional class of games under consideration were games with a simple parity [19, 12, 11, 21, 2, 31, 15, 16, 29, 18, 25, 27, 26, 3, 17, 13, 20] or payoff [24, 32, 15, 1, 27] objective. These games form a hierarchy with very simple tractable reductions from parity games through mean payoff games [24, 32, 15, 1, 27] and discounted payoff games [32, 15, 27] to simple stochastic games [9].

[1] Supported by the EPSRC projects 'Energy Efficient Control' (EP/M027287/1) and 'Solving Parity Games in Theory and Practice' (EP/P020909/1).
[2] Supported by the project "TriCS" (ZI 1516/1-1) of the German Research Foundation (DFG) and the Saarbrücken Graduate School of Computer Science.

27th EACSL Annual Conference on Computer Science Logic (CSL 2018).
Editors: Dan Ghica and Achim Jung; Article No. 35; pp. 35:1–35:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

More recently, games with a mixture of the qualitative parity condition and further quantitative objectives have been considered, including mean payoff parity games [8] and energy parity games [4]. Finitary parity games [7] take a special role within the class of games with mixed parity and payoff objectives. To win a finitary parity game, Player 0 needs to enforce a play with a bound b such that almost all occurrences of an odd color are followed by a higher even color within at most b steps.

This is interesting, because it provides a natural link between the qualitative and quantitative objective. One aspect that attracted attention is that, as long as one is not interested in optimizing the bound b, these games are the only games of the lot that are known to be tractable [7]. However, the bound b itself is also interesting: It serves as a native quality measure, because it limits the response time [30].

This property calls for a generalization to different cost models, and a first generalization has been made with the introduction of parity games with costs [14]. In parity games with costs, the basic cost function of finitary parity games—where each step incurs the same cost—is replaced with different non-negative costs for different edges. In this paper, we generalize this further to *general* integer costs: We decorate the edges with integer weights. The quantitative aspect in these parity games with weights consists of having to answer almost all odd colors by a higher even color, such that the *absolute* value of the weight of the path to this even color is bounded by a bound b.

In addition to their conceptual charm, we show that parity games with weights are PTIME equivalent to energy parity games. This indicates that these games are part of a natural complexity class, whereas the games with a plain objective appear to form a hierarchy. We use the reduction from parity games with weights to energy parity games to solve them. This reduction goes through intermediate reductions to and from *bounded* parity games with weights. These games have the additional restriction that the limit superior of the absolute weight of initial sequences of unanswered requests in a play is finite. These bounded parity games with weights are then reduced to energy parity games. The other direction of the reduction is through simple gadgets that preserve the main elements of winning strategies in games that are extended in two steps by very simple gadgets. As a result, we obtain the same complexity results for parity games with weights as for energy parity games, i.e., NP ∩ CO-NP, the signature complexity for finite games of infinite duration with parity conditions and their extensions. Thereby, we obtain an argument that these games might be representatives of a natural complexity class, lending a further argument for the relevance of two player games with mixed qualitative and quantitative winning conditions. Furthermore, Daviaud et al. recently showed that parity games with weights can even be solved in pseudo-quasi-polynomial time [10].

Naturally, parity games with weights subsume parity games (as a special case where all weights are zero), finitary parity games (as a special case where all weights are positive), and parity games with costs (as a special case where all weights are non-negative).

Finally, we show that the protagonist has finite-state winning strategies, and provide tight exponential bounds for the memory he needs to win the game. We also present tight bounds on the quality of winning strategies for the protagonist. Naturally, the antagonist may need infinite memory to win.

2 Preliminaries

We denote the non-negative integers by $\mathbb{N}$, the integers by $\mathbb{Z}$, and define $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$. As usual, we have $\infty > n$, $-\infty < n$, $n + \infty = \infty$, and $-\infty - n = -\infty$ for all $n \in \mathbb{Z}$.

An **arena** $\mathcal{A} = (V, V_0, V_1, E)$ consists of a finite, directed graph (V, E) and a partition $\{V_0, V_1\}$ of V into the positions of Player 0 (drawn as ellipses) and Player 1 (drawn as rectangles). The size of $\mathcal{A}$, denoted by $|\mathcal{A}|$, is defined as $|V|$. A **play** in $\mathcal{A}$ is an infinite path $\rho = v_0v_1v_2\cdots$ through (V, E). To rule out finite plays, we require every vertex to be non-terminal. We define $|\rho| = \infty$. Dually, for a finite play prefix $\pi = v_0 \cdots v_j$ we define $|\pi| = j + 1$.

A **game** $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ consists of an arena $\mathcal{A}$ with vertex set V and a set $\mathrm{Win} \subseteq V^\omega$ of winning plays for Player 0. The set of winning plays for Player 1 is $V^\omega \setminus \mathrm{Win}$. A winning condition Win is 0-extendable if, for all $\rho \in V^\omega$ and all $w \in V^*$, $\rho \in \mathrm{Win}$ implies $w\rho \in \mathrm{Win}$. Dually, Win is 1-extendable if, for all $\rho \in V^\omega$ and all $w \in V^*$, $\rho \notin \mathrm{Win}$ implies $w\rho \notin \mathrm{Win}$.

A **strategy** for Player $i \in \{0, 1\}$ is a mapping $\sigma \colon V^*V_i \to V$ such that $(v, \sigma(wv)) \in E$ holds true for all $wv \in V^*V_i$. We say that σ is **positional** if $\sigma(wv) = \sigma(v)$ holds true for every $wv \in V^*V_i$. A play $v_0v_1v_2\cdots$ is **consistent** with a strategy σ for Player i, if $v_{j+1} = \sigma(v_0 \cdots v_j)$ holds true for every j with $v_j \in V_i$. A strategy σ for Player i is a **winning strategy** for $\mathcal{G}$ from $v \in V$ if every play that starts in v and is consistent with σ is won by Player i. If Player i has a winning strategy from v, then we say Player i wins $\mathcal{G}$ from v. The **winning region** of Player i is the set of vertices, from which Player i wins $\mathcal{G}$; it is denoted by $\mathcal{W}_i(\mathcal{G})$. **Solving** a game amounts to determining its winning regions. If $\mathcal{W}_0(\mathcal{G}) \cup \mathcal{W}_1(\mathcal{G}) = V$, then we say that $\mathcal{G}$ is **determined**.

Let $\mathcal{A} = (V, V_0, V_1, E)$ be an arena and let $X \subseteq V$. The i-attractor of X is defined inductively as $\mathrm{Attr}_i(X) = \mathrm{Attr}_i^{|V|}(X)$, where $\mathrm{Attr}_i^0(X) = X$ and

$$\mathrm{Attr}_i^j(X) = \mathrm{Attr}_i^{j-1}(X) \cup \{v \in V_i \mid \exists v' \in \mathrm{Attr}_i^{j-1}(X).\, (v, v') \in E\} \\ \cup \{v \in V_{1-i} \mid \forall (v, v') \in E.\, v' \in \mathrm{Attr}_i^{j-1}(X)\} \ .$$

Hence, $\mathrm{Attr}_i(X)$ is the set of vertices from which Player i can force the play to enter X: Player i has a positional strategy σ_X such that each play that starts in some vertex in $\mathrm{Attr}_i(X)$ and is consistent with σ_X eventually encounters some vertex from X. We call σ_X an attractor strategy towards X. Moreover, the i-attractor can be computed in time linear in $|E|$ [23]. When we want to stress the arena $\mathcal{A}$ the attractor is computed in, we write $\mathrm{Attr}_i^{\mathcal{A}}(X)$.

A set $X \subseteq V$ is a trap for Player i, if every vertex in $X \cap V_i$ has only successors in X and every vertex in $X \cap V_{1-i}$ has at least one successor in X. In this case, Player $1 - i$ has a positional strategy τ_X such that every play starting in some vertex in X and consistent with τ_X never leaves X. We call such a strategy a trap strategy.

▶ **Remark 1.**
1. *The complement of an i-attractor is a trap for Player i.*
2. *If X is a trap for Player i, then $\mathrm{Attr}_{1-i}(X)$ is also a trap for Player i.*
3. *If Win is i-extendable and $(\mathcal{A}, \mathrm{Win})$ determined, then $\mathcal{W}_{1-i}(\mathcal{A}, \mathrm{Win})$ is a trap for Player i.*

A **memory structure** $\mathcal{M} = (M, \mathrm{init}, \mathrm{upd})$ for an arena (V, V_0, V_1, E) consists of a finite set M of memory states, an initialization function $\mathrm{init} \colon V \to M$, and an update function $\mathrm{upd} \colon M \times E \to M$. The update function can be extended to finite play prefixes in the usual way: $\mathrm{upd}^+(v) = \mathrm{init}(v)$ and $\mathrm{upd}^+(wvv') = \mathrm{upd}(\mathrm{upd}^+(wv), (v, v'))$ for $w \in V^*$ and $(v, v') \in E$. A next-move function $\mathrm{Nxt} \colon V_i \times M \to V$ for Player i has to satisfy $(v, \mathrm{Nxt}(v, m)) \in E$ for all $v \in V_i$ and $m \in M$. It induces a strategy σ for Player i with memory $\mathcal{M}$ via $\sigma(v_0 \cdots v_j) = \mathrm{Nxt}(v_j, \mathrm{upd}^+(v_0 \cdots v_j))$. A strategy is called **finite-state** if it can be implemented by a memory structure. We define $|\mathcal{M}| = |M|$. Slightly abusively, we say that the size of a finite-state strategy is the size of a memory structure implementing it.

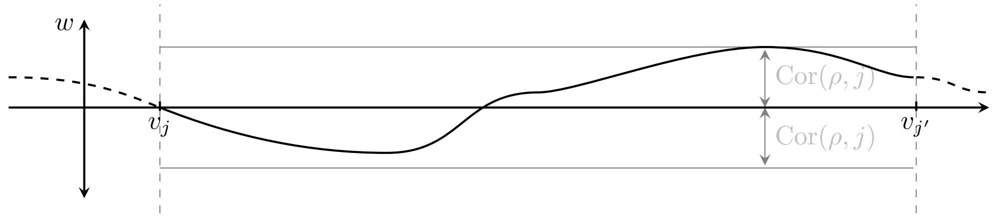

Figure 1 The cost-of-response of some request posed by visiting vertex v_j, which is answered by visiting vertex $v_{j'}$.

3 Parity Games with Weights

Fix an arena $\mathcal{A} = (V, V_0, V_1, E)$. A **weighting** for $\mathcal{A}$ is a function $w\colon E \to \mathbb{Z}$. We define $w(\varepsilon) = w(v) = 0$ for all $v \in V$ and extend w to sequences of vertices of length at least two by summing up the weights of the traversed edges. Given a play (prefix) $\pi = v_0 v_1 v_2 \cdots$, we define the amplitude of π as $\mathrm{Ampl}(\pi) = \sup_{j<|\pi|} |w(v_0 \cdots v_j)| \in \mathbb{N}_\infty$.

A **coloring** of V is a function $\Omega\colon V \to \mathbb{N}$. The classical parity condition requires almost all occurrences of odd colors to be answered by a later occurrence of a larger even color. Hence, let $\mathrm{Ans}(c) = \{c' \in \mathbb{N} \mid c' \geq c \text{ and } c' \text{ is even}\}$ be the set of colors that "answer" a "request" for color c. We denote a vertex v of color c by $^v/_c$.

Fijalkow and Zimmermann introduced a generalization of the parity condition and the finitary parity condition [7], the parity condition with costs [14]. There, the edges of the arena are labeled with *non-negative weights* and the winning condition demands that there exists a bound b such that almost all requests are answered with weight at most b, i.e., the weight of the infix between the request and the response has to be bounded by b.

Our aim is to extend the parity condition with costs by allowing for the full spectrum of weights to be used, i.e., by also incorporating negative weights. In this setting, the weight of an infix between a request and a response might be negative. Thus, the extended condition requires the weight of the infix to be bounded from above and from below.[3] To distinguish between the parity condition with costs and the extension introduced here, we call our extension the parity condition with weights.

Formally, let $\rho = v_0 v_1 v_2 \cdots$ be a play. We define the cost-of-response at position $j \in \mathbb{N}$ of ρ by

$$\mathrm{Cor}(\rho, j) = \min\{\mathrm{Ampl}(v_j \cdots v_{j'}) \mid j' \geq j, \Omega(v_{j'}) \in \mathrm{Ans}(\Omega(v_j))\}$$

where we use $\min \emptyset = \infty$. As the amplitude of an infix only increases by extending the infix, $\mathrm{Cor}(\rho, j)$ is the amplitude of the shortest infix that starts at position j and ends at an answer to the request posed at position j. We illustrate this notion in Figure 1.

We say that a request at position j is answered with cost b, if $\mathrm{Cor}(\rho, j) = b$. Consequently, a request with an even color is answered with cost zero. The cost-of-response of an unanswered request is infinite, even if the amplitude of the remaining play is bounded. In particular, this means that an unanswered request at position j may be "unanswered with finite cost b" (if the amplitude of the remaining play is $b \in \mathbb{N}$) or "unanswered with infinite cost" (if the amplitude of the remaining play is infinite). In either case, however, we have $\mathrm{Cor}(\rho, j) = \infty$.

[3] We discuss other possible interpretations of negative weights in Section 9.

We define the parity condition with weights as

$$\mathrm{WeightParity}(\Omega, w) = \{\rho \in V^\omega \mid \limsup_{j\to\infty} \mathrm{Cor}(\rho, j) \in \mathbb{N}\} .$$

I.e., ρ satisfies the condition if and only if there exists a bound $b \in \mathbb{N}$ such that almost all requests are answered with cost less than b. In particular, only finitely many requests may be unanswered, even with finite cost. Note that the bound b may depend on the play ρ.

We call a game $\mathcal{G} = (\mathcal{A}, \mathrm{WeightParity}(\Omega, w))$ a parity game with weights, and we define $|\mathcal{G}| = |\mathcal{A}| + \log(W)$, where W is the largest absolute weight assigned by w; i.e., we assume weights to be encoded in binary. If w assigns zero to every edge, then $\mathrm{WeightParity}(\Omega, w)$ is a classical (max-) parity condition, denoted by $\mathrm{Parity}(\Omega)$. Similarly, if w assigns positive weights to every edge, then $\mathrm{WeightParity}(\Omega, w)$ is equal to the finitary parity condition over Ω, as introduced by Chatterjee and Henzinger [6]. Finally, if w assigns only non-negative weights, then $\mathrm{WeightParity}(\Omega, w)$ is a parity condition with costs, as introduced by Fijalkow and Zimmermann [14]. In these cases, we refer to $\mathcal{G}$ as a parity game, a finitary parity game, or a parity game with costs, respectively. We recall the characteristics of these games in Table 1 on Page 15.

4 Solving Parity Games with Weights

We now show how to solve parity games with weights. Our approach is inspired by the classic work on finitary parity games [7] and parity games with costs [14]: We first define a stricter variant of these games, which we call bounded parity games with weights, and then show two reductions:

- parity games with weights can be solved in polynomial time with oracles that solve bounded parity games with weights (in this section); and
- bounded parity games with weights can be solved in polynomial time with oracles that solve energy parity games (Section 5).

Furthermore, in Section 8 we polynomially reduce solving energy parity games to solving parity games with weights and thereby show that parity games with weights, bounded parity games with weights, and energy parity games belong to the same complexity class.

The energy parity games that we reduce to are known to be efficiently solvable [4, 10]: they are in NP ∩ co-NP and can be solved in pseudo-quasi-polynomial time.

We first introduce the **bounded parity condition with weights**, which is a strengthening of the parity condition with weights. Hence, it is also induced by a coloring and a weighting:

$$\begin{aligned}\mathrm{BndWeightParity}(\Omega, w) &= \mathrm{WeightParity}(\Omega, w)\\ &\cap \{\rho \in V^\omega \mid \text{no request in } \rho \text{ is unanswered with infinite cost}\} .\end{aligned}$$

Note that this condition allows for a finite number of unanswered requests, as long as they are unanswered with finite cost.

We solve parity games with weights by repeatedly solving bounded parity games with weights. To this end, we apply the following two properties of the winning conditions: We have $\mathrm{BndWeightParity}(\Omega, w) \subseteq \mathrm{WeightParity}(\Omega, w)$ as well as that $\mathrm{WeightParity}(\Omega, w)$ is 0-extendable. Hence, if Player 0 has a strategy from a vertex v such that every consistent play has a suffix in $\mathrm{BndWeightParity}(\Omega, w)$, then the strategy is winning for her from v w.r.t. $\mathrm{WeightParity}(\Omega, w)$. Thus, $\mathrm{Attr}_0(\mathcal{W}_0(\mathcal{A}, \mathrm{BndWeightParity}(\Omega, w))) \subseteq \mathcal{W}_0(\mathcal{A}, \mathrm{WeightParity}(\Omega, w))$. The algorithm that solves parity games with weights repeatedly removes attractors of winning regions of the bounded parity game with weights until a fixed point is reached. We will later formalize this sketch to show that the removed parts are a subset of Player 0's winning region in the parity game with weights.

To show that the obtained fixed point covers the complete winning region of Player 0, we use the following lemma to show that the remaining vertices are a subset of Player 1's winning region in the parity game with weights. The proof is very similar to the corresponding one for finitary parity games and parity games with costs.

▶ **Lemma 2.** *Let* $\mathcal{G} = (\mathcal{A}, \mathrm{WeightParity}(\Omega, w))$ *and let* $\mathcal{G}' = (\mathcal{A}, \mathrm{BndWeightParity}(\Omega, w))$. *If* $\mathcal{W}_0(\mathcal{G}') = \emptyset$, *then* $\mathcal{W}_0(\mathcal{G}) = \emptyset$.

Lemma 2 implies that the algorithm for solving parity games with weights by repeatedly solving bounded parity games with weights (see Algorithm 1) is correct. Note that we use an oracle for solving bounded parity games with weights. We provide a suitable algorithm in Section 5.

Algorithm 1 A fixed-point algorithm computing $\mathcal{W}_0(\mathcal{A}, \mathrm{WeightParity}(\Omega, w))$.

```
k = 0; W_0^k = ∅; A_k = A
repeat
  k = k + 1
  X_k = W_0(A_{k−1}, BndWeightParity(Ω, w))
  W_0^k = W_0^{k−1} ∪ Attr_0^{A_{k−1}}(X_k)
  A_k = A_{k−1} \ Attr_0^{A_{k−1}}(X_k)
until X_k = ∅
return W_0^k
```

The loop terminates after at most $|\mathcal{A}|$ iterations (assuming the algorithm solving bounded parity games with weights terminates), as during each iteration at least one vertex is removed from the arena. The correctness proof relies on Lemma 2 and is similar to the one for finitary parity games [7] and for parity games with costs [14].

▶ **Lemma 3.** *Algorithm 1 returns* $\mathcal{W}_0(\mathcal{A}, \mathrm{WeightParity}(\Omega, w))$

The winning strategy defined in the proof of Lemma 3 can be implemented by a memory structure of size $\max_{k \leq k^*} s_k$, where s_k is the size of a winning strategy σ_k for Player 0 in the bounded parity game with weights solved in the k-th iteration, and where k^* is the value of k at termination. To this end, one uses the fact that the winning regions X_k are disjoint and are never revisited once left. Hence, we can assume the implementations of the σ_k to use the same states.

5 Solving Bounded Parity Games with Weights

After having reduced the problem of solving parity games with weights to that of solving (multiple) bounded parity games with weights, we reduce solving bounded parity games with weights to solving (multiple) energy parity games [4].

Similarly to a parity game with weights, in an energy parity game, the vertices are colored and the edges are equipped with weights. It is the goal of Player 0 to satisfy the parity condition, while, at the same time, ensuring that the weight of every infix, its so-called energy level, is bounded from below. In contrast to a parity game with weights, however, the weights in an energy parity game are not tied to the requests and responses denoted by the coloring.

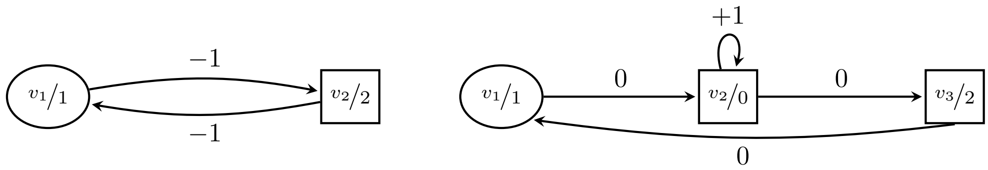

Figure 2 The difference between energy parity games and parity games with weights.

Consider, for example, the games shown in Figure 2. In the game on the left-hand side, players only have a single, trivial strategy. If we interpret this game as a parity game with weights, Player 0 wins from every vertex, as each request is answered with cost one. If we, however, interpret that game as an energy parity game, Player 1 instead wins from every vertex, since the energy level decreases by one with every move. In the game on the right-hand side, the situation is mirrored: When interpreting this game as a parity game with weights, Player 1 wins from every vertex, as she can easily unbound the costs of the requests for color one by staying in vertex v_2 for an ever-increasing number of cycles. Dually, when interpreting this game as an energy parity game, Player 0 wins from every vertex, since the parity condition is clearly satisfied in every play, and Player 1 is only able to increase the energy level, while it is never decreased.

In Section 5.1, we introduce energy parity games formally and present how to solve bounded parity games with weights via energy games in Section 5.2.

5.1 Energy Parity Games

An energy parity game $\mathcal{G} = (\mathcal{A}, \Omega, w)$ consists of an arena $\mathcal{A} = (V, V_0, V_1, E)$, a coloring $\Omega\colon V \to \mathbb{N}$ of V, and an edge weighting $w\colon E \to \mathbb{Z}$ of E. Note that this definition is not compatible with the framework presented in Section 2, as we have not (yet) defined the winner of the plays. This is because they depend on an initial credit, which is existentially quantified in the definition of winning the game $\mathcal{G}$. Formally, the set of winning plays with initial credit $c_0 \in \mathbb{N}$ is defined as

$$\mathrm{EnergyParity}_{c_0}(\Omega, w) = \mathrm{Parity}(\Omega) \cap \{v_0 v_1 v_2 \cdots \in V^\omega \mid \forall j \in \mathbb{N}.\, c_0 + w(v_0 \cdots v_j) \geq 0\} \ .$$

Now, we say that Player 0 wins $\mathcal{G}$ from v if there exists some initial credit $c_0 \in \mathbb{N}$ such that he wins $\mathcal{G}_{c_0} = (\mathcal{A}, \mathrm{EnergyParity}_{c_0}(\Omega, w))$ from v (in the sense of the definitions in Section 2). If this is not the case, i.e., if Player 1 wins $\mathcal{G}_{c_0}$ from v for every c_0, then we say that Player 1 wins $\mathcal{G}$ from v. Note that the initial credit is uniform for all plays, unlike the bound on the cost-of-response in the definition of the parity condition with weights, which may depend on the play.

Unravelling these definitions shows that Player 0 wins $\mathcal{G}$ from v if there is an initial credit c_0 and a strategy σ, such that every play that starts in v and is consistent with σ satisfies the parity condition *and* the accumulated weight over the play prefixes (the energy level) never drops below $-c_0$. We call such a strategy σ a winning strategy for Player 0 in $\mathcal{G}$ from v. Dually, Player 1 wins $\mathcal{G}$ from v if, for every initial credit c_0, there is a strategy τ_{c_0}, such that every play that starts in v and is consistent with τ_{c_0} violates the parity condition *or* its energy level drops below $-c_0$ at least once. Thus, the strategy τ_{c_0} may, as the notation suggests, depend on c_0. However, Chatterjee and Doyen showed that using different strategies is not necessary: There is a uniform strategy τ that is winning from v for every initial credit c_0.

▶ **Proposition 4** ([4]). *Let $\mathcal{G}$ be an energy parity game. If Player 1 wins $\mathcal{G}$ from v, then she has a single positional strategy that is winning from v in $\mathcal{G}_{c_0}$ for every c_0.*

We call such a strategy as in Proposition 4 a winning strategy for Player 1 from v. A play consistent with such a strategy either violates the parity condition, or the energy levels of its prefixes diverge towards $-\infty$.

Furthermore, Chatterjee and Doyen obtained an upper bound on the initial credit necessary for Player 0 to win an energy parity game, as well an upper bound on the size of a corresponding finite-state winning strategy.

▶ **Proposition 5** ([4]). *Let $\mathcal{G}$ be an energy parity game with n vertices, d colors, and largest absolute weight W. The following are equivalent for a vertex v of $\mathcal{G}$:*
1. *Player 0 wins $\mathcal{G}$ from v.*
2. *Player 0 wins $\mathcal{G}_{(n-1)W}$ from v with a finite-state strategy with at most ndW states.*

The previous proposition yields that finite-state strategies of bounded size suffice for Player 0 to win.

Such strategies do not admit long expensive descents, which we show by a straightforward pumping argument.

▶ **Lemma 6.** *Let $\mathcal{G}$ be an energy parity game with n vertices and largest absolute weight W. Further, let σ be a finite-state strategy of size s, and let ρ be a play that starts in some vertex, from which σ is winning, and is consistent with σ. Every infix π of ρ satisfies $w(\pi) > -Wns$.*

Moreover, Chatterjee and Doyen gave an upper bound on the complexity of solving energy parity games, which was recently supplemented by Daviaud et al. with an algorithm solving them in pseudo-quasi-polynomial time.

▶ **Proposition 7** ([4, 10]). *The following problem is in $\mathrm{NP} \cap \mathrm{co\text{-}NP}$ and can be solved in pseudo-quasi-polynomial time: "Given an energy parity game $\mathcal{G}$ and a vertex v in $\mathcal{G}$, does Player 0 win $\mathcal{G}$ from v?"*

5.2 From Bounded Parity Games with Weights to Energy Parity Games

Let $\mathcal{G} = (\mathcal{A}, \mathrm{BndWeightParity}(\Omega, w))$ be a bounded parity game with weights with vertex set V. Without loss of generality, we assume $\Omega(v) \geq 2$ for all $v \in V$. We construct, for each vertex v^* of $\mathcal{A}$, an energy parity game $\mathcal{G}_{v^*}$ with the following property: Player 1 wins $\mathcal{G}_{v^*}$ from some designated vertex induced by v^* if and only if she is able to unbound the amplitude for the request of the initial vertex of the play when starting from v^*. This construction is the technical core of the fixed-point algorithm that solves bounded parity games with weights via solving energy parity games.

The main obstacle towards this is that, in the bounded parity game with weights $\mathcal{G}$, Player 1 may win by unbounding the amplitude for a request from above or from below, while she can only win $\mathcal{G}_{v^*}$ by unbounding the costs from below. We model this in $\mathcal{G}_{v^*}$ by constructing two copies of $\mathcal{A}$. In one of these copies the edge weights are copied from $\mathcal{G}$, while they are inverted in the other copy. We allow Player 1 to switch between these copies arbitrarily. To compensate for Player 1's power to switch, Player 0 can increase the energy level in the resulting energy parity game during each switch.

First, we define the set of polarities $P = \{+, -\}$ as well as $\overline{+} = -$ and $\overline{-} = +$. Given a vertex v^* of $\mathcal{A}$, define the "polarized" arena $\mathcal{A}_{v^*} = (V', V'_0, V'_1, E')$ of $\mathcal{A} = (V, V_0, V_1, E)$ with

- $V' = (V \times P) \cup (E \times P \times \{0, 1\})$,

- $V'_i = (V_i \times P) \cup (E \times P \times \{i\})$ for $i \in \{0, 1\}$, and
- E' contains the following edges for every edge $e = (v, v') \in E$ with $\Omega(v) \notin \mathrm{Ans}(\Omega(v^*))$ and every polarity $p \in P$:
  - $((v, p), (e, p, 1))$: The player whose turn it is at v picks a successor v'. The edge $e = (v, v')$ is stored as well as the polarity p.
  - $((e, p, 1), (v', p))$: Then, Player 1 can either keep the polarity p unchanged and execute the move to v', or
  - $((e, p, 1), (e, p, 0))$: she decides to change the polarity, and another auxiliary vertex is reached.
  - $((e, p, 0), (e, p, 0))$: If the polarity is to be changed, then Player 0 is able to use a self-loop to increase the energy level (see below), before
  - $((e, p, 0), (v', \overline{p}))$: he can eventually complete the polarity switch by moving to v'.
- Furthermore, for every vertex v with $\Omega(v) \in \mathrm{Ans}(\Omega(v^*))$ and every polarity $p \in P$, E' contains the self-loop $((v, p), (v, p))$.[4]

Thus, a play in $\mathcal{A}_{v^*}$ simulates a play in $\mathcal{A}$, unless Player 0 stops the simulation by using the self-loop at a vertex of the form $(e, p, 0)$ ad infinitum, and unless an answer to $\Omega(v^*)$ is reached. We define the coloring and the weighting for $\mathcal{A}_{v^*}$ so that Player 0 loses in the former case and wins in the latter case. Furthermore, the coloring is defined so that all simulating plays that are not stopped have the same color sequence as the simulated play (save for irrelevant colors on the auxiliary vertices in $E \times P \times \{0, 1\}$). Hence, we define

$$\Omega_{v^*}(v) = \begin{cases} \Omega(v') & \text{if } v = (v', p) \text{ with } v' \notin \mathrm{Ans}(\Omega(v^*)) \ , \\ 0 & \text{if } v = (v', p) \text{ with } v' \in \mathrm{Ans}(\Omega(v^*)) \ , \\ 1 & \text{otherwise } . \end{cases}$$

As desired, due to our assumption that $\Omega(v) \geq 2$ for all $v \in V$, the vertices from $E \times P \times \{0, 1\}$ do not influence the maximal color visited infinitely often during a play, unless Player 0 opts to remain in some $(e, p, 0)$ ad infinitum (and thereby violating the parity condition) or an answer to the color of v^* is reached (and thereby satisfying the parity condition).

Moreover, recall that our aim is to allow Player 1 to choose the polarity of edges by switching between the two copies of $\mathcal{A}$ occurring in $\mathcal{A}_{v^*}$. Intuitively, Player 1 should opt for positive polarity in order to unbound the costs incurred by the request posed by v^* from above, while she should opt for negative polarity in order to unbound these costs from below. Since in an energy parity game, it is, broadly speaking, beneficial for Player 1 to move along edges of negative weight, we negate the weights of edges in the copy of $\mathcal{A}$ with positive polarity. Thus, we define

$$w_{v^*}(e) = \begin{cases} -w(v, v') & \text{if } e = ((v, +), ((v, v'), +, 1)) \ , \\ w(v, v') & \text{if } e = ((v, -), ((v, v'), -, 1)) \ , \\ 1 & \text{if } e = ((e, p, 0), (e, p, 0)) \ , \\ 0 & \text{otherwise } . \end{cases}$$

This definition implies that the self-loops at vertices of the form (v, p) with $\Omega(v) \in \mathrm{Ans}(\Omega(v^*))$ have weight zero. Combined with the fact that these vertices have color zero, this allows

[4] Note that this definition introduces some terminal vertices, i.e., those of the form $((v, v'), p, i)$ with $\Omega(v) \in \mathrm{Ans}(\Omega(v^*))$. However, these vertices also have no incoming edges. Hence, to simplify the definition, we just ignore them.

Player 0 to win $\mathcal{G}_{v^*}$ by reaching such a vertex. Intuitively, answering the request posed at v^* is beneficial for Player 0. In particular, if $\Omega(v^*)$ is even, then Player 0 wins $\mathcal{G}_{v^*}$ trivially from (v^*, p), as we then have $\Omega(v^*) \in \mathrm{Ans}(\Omega(v^*))$.

Finally, define the energy parity game $\mathcal{G}_{v^*} = (\mathcal{A}_{v^*}, \Omega_{v^*}, w_{v^*})$. In the following, we are only interested in plays starting in vertex $(v^*, +)$ in $\mathcal{G}_{v^*}$.

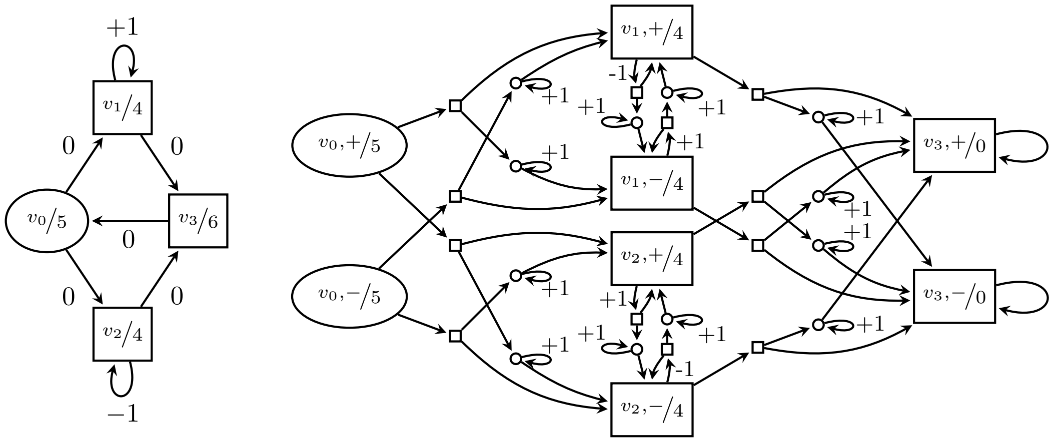

Figure 3 A bounded parity game with weights $\mathcal{G}$ and the associated energy parity game $\mathcal{G}_{v_0}$. The unnamed vertices of Player 1 (Player 0) are of the form $((v, v'), p, 1)$ (of the form $((v, v'), p, 0)$) when between the vertices (v, p) and (v', p'). All missing edge weights in $\mathcal{G}_{v_0}$ are 0.

▶ **Example 8.** Consider the bounded parity game with weights depicted on the left hand side of Figure 3 and the associated energy parity game $\mathcal{G}_{v_0}$ on the right side. First, let us note that all other $\mathcal{G}_v$ for $v \neq v_0$ are trivial in that they all consist of a single vertex (reachable from $(v, +)$), which has even color with a self-loop of weight zero. Hence, Player 0 wins each of these games from $(v, +)$.

Player 1 wins $\mathcal{G}$ from v_0, where a request for color 5 is opened, which is then kept unanswered with infinite cost by using the self-loop at v_1 or v_2 ad infinitum, depending on which successor Player 0 picks.

We show that Player 1 wins $\mathcal{G}_{v_0}$ from $(v_0, +)$: the outgoing edges of $(v_0, +)$ correspond to picking the successor v_1 or v_2 as in $\mathcal{G}$. Before this is executed, however, Player 1 gets to pick the polarity of the successor: she should pick $+$ for v_1 and $-$ for v_2. Now, Player 0 may use the self-loop at her "tiny" vertices ad infinitum. These vertices have color one, i.e., Player 1 wins the resulting play. Otherwise, we reach the vertex $(v_1, +)$ or $(v_2, -)$. From both vertices, Player 1 can enforce a loop of negative weight, which allows him to win by violating the energy condition.

Note that the winning strategy for Player 1 for $\mathcal{G}$ from v is very similar to that for her for $\mathcal{G}_{v_0}$ from $(v_0, +)$. We show that one direction holds in general: A winning strategy for Player 0 for $\mathcal{G}_v$ from $(v, +)$ is "essentially" one for him in $\mathcal{G}$ from v.

Note that the other direction does, in general, not hold. This can be seen by adding a vertex v_{-1} of color 3 with a single edge to v_0. Then, vertices of the form (v_i, p) with $i \in \{1, 2\}$ in $\mathcal{G}_{v_{-1}}$ are winning sinks for Player 0. Hence, he wins $\mathcal{G}_{v_{-1}}$ from (v_{-1}, p) in spite of losing the bounded parity game with weights from v_{-1}.

Hence, the initial request the vertex v inducing $\mathcal{G}_v$ plays a special role in the construction: It is the request Player 1 aims to keep unanswered with infinite cost. To overcome this and to complete our construction, we show a statement reminiscent of Lemma 2: If Player 0 wins $\mathcal{G}_v$ from $(v, +)$ for every v, then she also wins $\mathcal{G}$x from every vertex. With this relation at hand, one can again construct a fixed-point algorithm solving bounded parity games with weights using an oracle for solving energy parity games that is very similar to Algorithm 1.

Formally, we have the following lemma, which forms the technical core of our algorithm that solves bounded parity games with weights by solving energy parity games.

▶ **Lemma 9.** *Let $\mathcal{G}$ be a bounded parity game with weights with vertex set V.*

1. *Let $v^* \in V$. If Player 1 wins $\mathcal{G}_{v^*}$ from $(v^*, +)$, then $v^* \in \mathcal{W}_1(\mathcal{G})$.*
2. *If Player 0 wins $\mathcal{G}_{v^*}$ from $(v^*, +)$ for all $v^* \in V$, then $\mathcal{W}_1(\mathcal{G}) = \emptyset$.*

This lemma is the main building block for the algorithm that solves bounded parity games with weights by repeatedly solving energy parity games, which is very similar to Algorithm 1. Indeed, we just swap the roles of the players: We compute 1-attractors instead of 0-attractors and we change the definition of X_k. Hence, we obtain the following algorithm (Algorithm 2).

Algorithm 2 A fixed-point algorithm computing $\mathcal{W}_1(\mathcal{A}, \mathrm{BndWeightParity}(\Omega, w))$.

$k = 0$; $W_1^k = \emptyset$; $\mathcal{A}_k = \mathcal{A}$
repeat
  $k = k + 1$
  $X_k = \{v^* \mid \text{Player 1 wins the energy parity game } ((\mathcal{A}_{k-1})_{v^*}, \Omega_{v^*}, w_{v^*}) \text{ from } (v^*, +)\}$
  $W_1^k = W_1^{k-1} \cup \mathrm{Attr}_1^{\mathcal{A}_{k-1}}(X_k)$
  $\mathcal{A}_k = \mathcal{A}_{k-1} \setminus \mathrm{Attr}_1^{\mathcal{A}_{k-1}}(X_k)$
until $X_k = \emptyset$
return W_1^k

Algorithm 2 terminates after solving at most a quadratic number of energy parity games. Furthermore, the proof of correctness is analogous to the one for Algorithm 1, relying on Lemma 9. We only need two further properties: the 1-extendability of $\mathrm{BndWeightParity}(\Omega, w)$, and an assertion that $\mathrm{Attr}_1^{\mathcal{A}_{k-1}}(X_k)$ is a trap for Player 0 in $\mathcal{A}_{k-1}$. Both are easy to verify.

After plugging Algorithm 2 into Algorithm 1, Proposition 7 yields our main theorem, settling the complexity of solving parity games with weights.

▶ **Theorem 10.** *The following problem is in* $\mathrm{NP} \cap \mathrm{CO\text{-}NP}$ *and can be solved in pseudo-quasi-polynomial time: "Given a parity game with weights $\mathcal{G}$ and a vertex v in $\mathcal{G}$, does Player 0 win $\mathcal{G}$ from v?"*

6 Memory Requirements

We now discuss the upper and lower bounds on the memory required to implement winning strategies for either player. Recall that we use binary encoding to denote weights, i.e., weights may be exponential in the size of the game. In this section we show polynomial (in n, d, and W) upper and lower bounds on the necessary and sufficient memory for Player 0 to win parity games with weights. Due to the binary encoding of weights, these bounds are exponential in the size of the game. In contrast, Player 1 requires infinite memory.

▶ **Theorem 11.** *Let $\mathcal{G}$ be a parity game with weights with n vertices, d colors, and largest absolute weight W assigned to any edge in $\mathcal{G}$. Moreover, let v be a vertex of $\mathcal{G}$.*

1. *Player 0 has a winning strategy σ from $\mathcal{W}_0(\mathcal{G})$ with $|\sigma| \in \mathcal{O}(nd^2W)$. This bound is tight.*

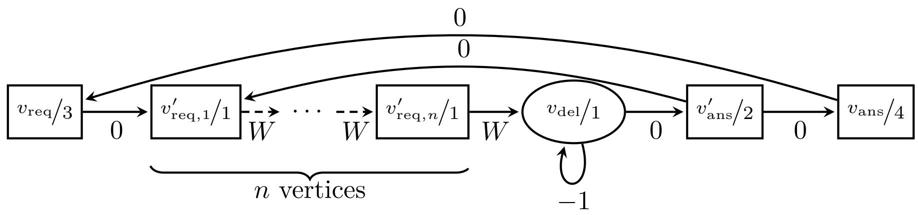

Figure 4 A game of size $\mathcal{O}(n)$ in which Player 0 only wins with strategies of size at least $nW + 1$.

2. *There exists a parity game with weights $\mathcal{G}$, such that Player 1 has a winning strategy from each vertex v in $\mathcal{G}$, but she has no finite-state winning strategy from any v in $\mathcal{G}$.*

The proof of the second item of Theorem 11 is straightforward, since Player 1 already requires infinite memory to implement winning strategies in finitary parity games [7]. Since parity games with weights subsume finitary parity games, this result carries over to our setting. We show the game witnessing this lower bound on the right-hand side of Figure 2.

In contrast, exponential memory is sufficient, but also necessary, for Player 0. To this end, we first prove that the winning strategy for him constructed in the proof of Lemma 9.2 suffers at most a linear blowup in comparison to his winning strategies in the underlying energy parity games. This is sufficient as we have argued in Section 4 that the construction of a winning strategy for Player 0 in a parity game with weights suffers no blowup in comparison to the underlying bounded parity games with weights.

▶ **Lemma 12.** *Let $\mathcal{G}$, n, d, and W be as in Theorem 11. Player 0 has a finite-state winning strategy of size at most $d(6n)(d + 2)(W + 1)$ from $\mathcal{W}_0(\mathcal{G})$.*

Having established an upper bound on the memory required by Player 0, we now proceed to show that this exponential bound is indeed tight, which is witnessed by the games $\mathcal{G}_n$ depicted in Figure 4.

▶ **Lemma 13.** *Let $n, W \in \mathbb{N}$. There exists a parity game with weights $\mathcal{G}_{n,W}$ with n vertices and largest absolute weight W such that Player 0 wins $\mathcal{G}_n$ from every vertex, but each winning strategy for her is of size at least $nW + 1$.*

7 Quality of Strategies

We have shown in the previous section that finite-state strategies of bounded size suffice for Player 0 to win in parity games with weights, while Player 1 clearly requires infinite memory. However, as we are dealing with a quantitative winning condition, we are not only interested in the size of winning strategies, but also in their quality. More precisely, we are interested in an upper bound on the cost of requests that Player 0 can ensure. In this section, we show that he can guarantee an exponential upper bound on such costs. Dually, Player 1 is required to unbound the cost of responses.

▶ **Theorem 14.** *Let $\mathcal{G}$ be a parity game with weights with n vertices, d colors, and largest absolute weight W.*

There exists a $b \in \mathcal{O}((ndW)^2)$ and a strategy σ for Player 0 such that, for all plays ρ beginning in $\mathcal{W}_0(\mathcal{G})$ and consistent with σ, we have $\limsup_{j\to\infty} \mathrm{Cor}(\rho, j) \le b$. This bound is tight.

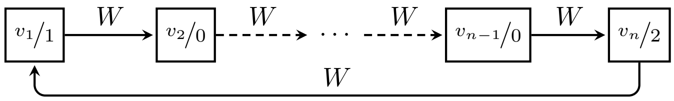

Figure 5 The game $\mathcal{G}_{n,W}$ witnessing an exponential lower bound on the cost that Player 0 can ensure.

We first show that Player 0 can indeed ensure an upper bound as stated in Theorem 14. We obtain this bound via a straightforward pumping argument leveraging the upper bound on the size of winning strategies obtained in Lemma 12.

▶ **Lemma 15.** *Let $\mathcal{G}$, n, d, and W be as in the statement of Theorem 14 and let $s = d(6n)(d+2)(W+1)$. Player 0 has a winning strategy σ such that, for each play ρ that starts in $\mathcal{W}_0(\mathcal{G})$ and is consistent with σ, we have $\limsup_{j\to\infty} \mathrm{Cor}(\rho, j) \leq nsW$.*

Having thus shown that Player 0 can indeed ensure an exponential upper bound on the incurred cost, we now proceed to show that this bound is tight. A simple example shows that there exists a series of parity games with weights, in which Player 0 wins from every vertex, but in which he cannot enforce a sub-exponential cost of any request.

▶ **Lemma 16.** *Let $n, W \in \mathbb{N}$. There exists a parity game with weights $\mathcal{G}_{n,W}$ with n vertices and largest absolute weight W as well as a vertex $v \in \mathcal{W}_0(\mathcal{G})$, such that for each winning strategy for Player 0 from v there exists a play ρ starting in v and consistent with σ with $\limsup_{j\to\infty} \mathrm{Cor}(\rho, j) \geq (n-1)W$.*

Proof. We show the game $\mathcal{G}_{n,W}$ in Figure 5. The arena of $\mathcal{G}_{n,W}$ is a cycle with n vertices of Player 1, where each edge has weight W. Moreover, one vertex is labeled with color two, its directly succeeding vertex is labeled with color one. All remaining vertices have color zero.

Player 0 only has a single strategy in this game and there exist only n plays in $\mathcal{G}_{n,W}$, each starting in a different vertex of $\mathcal{G}_n$. In each play, each request for color one is only answered after $n-1$ steps, each contributing a cost of W. Hence, this request incurs a cost of $(n-1)W$. Moreover, as this request is posed and answered infinitely often in each play, we obtain the desired result. ◀

8 From Energy Parity Games to (Bounded) Parity Games with Weights

We have discussed in Sections 4 and 5 how to solve parity games with weights via solving bounded parity games with weights and how to solve the latter games by solving energy parity games, both steps with a polynomial overhead. An obvious question is whether one can also solve energy parity games by solving (bounded) parity games with weights. In this section, we answer this question affirmatively. We show how to transform an energy parity game into a bounded parity game with weights so that solving the latter also solves the former. Then, we show how to transform a bounded parity game with weights into a parity game with weights with the same relation: Solving the latter also solves the former. Both constructions here are gadget based and increase the size of the arenas only linearly. Hence, all three types of games are interreducible with at most polynomial overhead.

8.1 From Energy Parity Games to Bounded Parity Games with Weights

Note that, in an energy parity game, Player 0 wins if the energy increases without a bound, as long as there is a lower bound. However, in a bounded parity game, he has to ensure an upper and a lower bound. Thus, we show in a first step how to modify an energy parity game so that Player 0 still has to ensure a lower bound on the energy, but can also *throw away* unnecessary energy during each transition, thereby also ensuring an upper bound. The most interesting part of this construction is to determine when energy becomes unnecessary to ensure a lower bound. Here, we rely on Lemma 6.

Formally, let $\mathcal{G} = (\mathcal{A}, \Omega, w)$ be an energy parity game with $\mathcal{A} = (V, V_0, V_1, E)$ where we assume w.l.o.g. that the minimal color in $\Omega(V)$ is strictly greater than 1. Now, we define $\mathcal{G}' = (\mathcal{A}', \Omega', w')$ with $\mathcal{A} = (V, V_0, V_1, E)$ where

- $V' = V \cup E$, $V_0' = V_0 \cup E$, and $V_1' = V_1$,
- $E' = \{(v, e), (e, e), (e, v') \mid e = (v, v') \in E\}$,
- $\Omega'(v) = \Omega(v)$ and $\Omega'(e) = 1$, and
- $w'(v, e) = w(e)$, $w'(e, e) = -1$, and $w(e, v') = 0$ for every $e = (v, v') \in E$.

Intuitively, every edge of $\mathcal{A}$ is subdivided and a new vertex for Player 0 is added, where he can decrease the energy level. The negative weight ensures that he eventually leaves this vertex in order to satisfy an energy condition.

We say that a strategy σ for Player 0 in $\mathcal{A}'$ is corridor-winning for him from some $v \in V$, if there is a $b \in \mathbb{N}$ such that every play ρ that starts in v and is consistent with σ satisfies the parity condition and $\mathrm{Ampl}(\rho) \leq b$. Hence, instead of just requiring a lower bound on the energy level as in the energy parity condition, we also require a uniform upper bound on the energy level (where we w.l.o.g. assume these bounds to coincide).

▶ **Lemma 17.** *Let $\mathcal{G}$ and $\mathcal{G}'$ be as above and let $v \in V$. Player 0 has a winning strategy for $\mathcal{G}$ from v if and only if Player 0 has a corridor-winning strategy for $\mathcal{G}'$ from v.*

Now, we turn $\mathcal{G}'$ into a bounded parity game with weights. In such a game, the cost-of-response of every request has to be bounded, but the overall energy level of the play may still diverge to $-\infty$. To rule this out, we open one unanswerable request at the beginning of each play, which has to be unanswered with finite cost in order to satisfy the bounded parity condition with weights. If this is the case, then the energy level of the play is always in a bounded corridor, i.e., we obtain a corridor-winning strategy.

Formally, for every vertex $v \in V$, we add a vertex $\overline{v}$ to $\mathcal{A}'$ of an odd color c^* that is larger than every color in $\Omega(V)$, i.e., the request can never be answered. Furthermore, $\overline{v}$ has a single outgoing edge to v of weight 0, i.e., it is irrelevant whose turn it is. Call the resulting arena $\mathcal{A}''$, the resulting coloring Ω'', and the resulting weighting w'', and let $\mathcal{G}'' = (\mathcal{A}'', \mathrm{BndWeightParity}(\Omega'', w''))$.

▶ **Lemma 18.** *Let $\mathcal{G}'$ and $\mathcal{G}''$ be as above and let $v \in V$. Player 0 has a corridor-winning strategy for $\mathcal{G}'$ from v if and only if $\overline{v} \in \mathcal{W}_0(\mathcal{G}'')$.*

8.2 From Bounded Parity Games with Weights to Parity Games with Weights

Next, we show how to turn a bounded parity game with weights into a parity game with weights so that solving the latter also solves the former. The construction here uses the same restarting mechanism that underlies the proof of Lemma 2: as soon as a request has incurred a cost of b, restart the play and enforce a request of cost $b + 1$, and so on. Unlike the proof of Lemma 2, where Player 1 could restart the play at any vertex, here we always have to return to a fixed initial vertex we are interested in. While resetting, we have to answer all requests in order to prevent Player 1 to use the reset to prevent requests from being answered. Assume $v^* \in V$ is the initial vertex we are interested in. Then, we subdivide every edge in $\mathcal{A}''$ to allow Player 1 to restart the play by answering all open requests and then moving back to v^*.

Formally, fix a bounded parity game with weights $\mathcal{G} = (\mathcal{A}, \text{BndWeightParity}(\Omega, w))$ with $\mathcal{A} = (V, V_0, V_1, E)$ and a vertex $v^* \in V$. We define the parity game with weights $\mathcal{G}_{v^*} = (\mathcal{A}_{v^*}, \text{WeightParity}(\Omega_{v^*}, w_{v^*}))$ with $\mathcal{A}_{v^*} = (V', V'_0, V'_1, E')$ where

- $V' = V \cup E \cup \{\top\}$, $V'_0 = V_0$, and $V'_1 = V_1 \cup E \cup \{\top\}$,
- $E' = \{(v, e), (e, \top), (e, v') \mid e = (v, v') \in E\} \cup \{(\top, v^*)\}$,
- $\Omega_{v^*}(v) = \Omega(v)$, $\Omega_{v^*}(e) = 0$ for every $e \in E$, and $\Omega_{v^*}(\top) = 2\max(\Omega(V))$, and
- $w_{v^*}(v, e) = w(e)$ for $(v, e) \in V \times E$ and $w_{v^*}(e') = 0$ for every other edge $e' \in E'$.

▶ **Lemma 19.** *Let $\mathcal{G}$ and $\mathcal{G}_{v^*}$ as above. Then, $v^* \in \mathcal{W}_0(\mathcal{G})$ if and only if $v^* \in \mathcal{W}_0(\mathcal{G}_{v^*})$.*

9 Conclusions and Future Work

We have established that parity games with weights and bounded parity games fall into the same complexity class as energy parity games. This is interesting, because, while solving such games has the signature complexity class NP ∩ co-NP, they are not yet considered a class in their own right. It is also interesting because the properties appear to be inherently different: While they both combine the qualitative parity condition with quantified costs, parity games with weights *combine* these aspects on the property level, whereas energy parity games simply look at the combined—and totally unrelated—properties. We show the characteristic properties of parity games and of games with combinations of a parity condition with quantitative conditions relevant for this work in Table 1.

	Complexity	Mem. Pl. 0/Pl. 1	Bounds
Parity Games [3]	quasi-poly.	pos./pos.	–
Energy Parity Games [4, 10]	pseudo-quasi-poly.	$\mathcal{O}(ndW)$/pos.	$\mathcal{O}(nW)$
Finitary Parity Games [7]	poly.	pos./inf.	$\mathcal{O}(nW)$
Parity Games with Costs [14, 22]	quasi-poly.	pos./inf.	$\mathcal{O}(nW)$
Parity Games with Weights	pseudo-quasi-poly.	$\mathcal{O}(nd^2W)$/inf.	$\mathcal{O}((ndW)^2)$

Table 1 Characteristic properties of variants of parity games.

As future work, we are looking into the natural extensions of parity games with weights to Streett games with weights [7, 14], and at the complexity of determining optimal bounds and strategies that obtain them [30]. We are also looking at variations of the problem. The two natural variations are

- to use a one-sided definition (instead of the absolute value) for the amplitude of a play, i.e., using $\text{Ampl}(\pi) = \sup_{j<|\pi|} w(v_0 \cdots v_j) \in \mathbb{N}_\infty$ (instead of $\text{Ampl}(\pi) = \sup_{j<|\pi|} |w(v_0 \cdots v_j)| \in \mathbb{N}_\infty$), and
- to use an arbitrary consecutive subsequence of a play, i.e., $\text{Ampl}(\pi) = \sup_{j \leq k<|\pi|} |w(v_j \cdots v_k)| \in \mathbb{N}_\infty$.

There are good arguments in favor and against using these individual variations—and their combination to $\text{Ampl}(\pi) = \sup_{j \leq k<|\pi|} w(v_j \cdots v_k) \in \mathbb{N}_\infty$—but we feel that the introduction of parity games with weights benefit from choosing one of the four combinations as *the* parity games with weights.

We expect the complexity to rise when changing from maximizing over the absolute value to maximizing over the value, as this appears to be close to pushdown boundedness games [5], and we conjecture this problem to be PSPACE complete.

References

1 Henrik Björklund and Sergei Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. *Discrete Appl. Math.*, 155(2):210–229, 2007. doi:http://dx.doi.org/10.1016/j.dam.2006.04.029.

2 Anca Browne, Edmund M. Clarke, Somesh Jha, David E. Long, and Wilfredo R. Marrero. An improved algorithm for the evaluation of fixpoint expressions. *Theo. Comp. Sci.*, 178(1–2):237–255, 1997. doi:10.1016/S0304-3975(96)00228-9.

3 C. S. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan. Deciding parity games in quasipolynomial time. In *STOC 2017*, pages 252–263. ACM Press, 2017. doi:10.1145/3055399.3055409.

4 Krishnendu Chatterjee and Laurent Doyen. Energy Parity Games. *Theo. Comp. Sci.*, 458:49–60, 2012. doi:10.1016/j.tcs.2012.07.038.

5 Krishnendu Chatterjee and Nathanaël Fijalkow. Infinite-state games with finitary conditions. In *CSL 2013*, volume 23 of *LIPIcs*, pages 181–196. Schloss Dagstuhl–LZI, 2013. doi:10.4230/LIPIcs.CSL.2013.181.

6 Krishnendu Chatterjee and Thomas A. Henzinger. Finitary winning in omega-regular games. In Holger Hermanns and Jens Palsberg, editors, *TACAS 2006*, volume 3920 of *LNCS*, pages 257–271, 2006. doi:10.1007/11691372_17.

7 Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. Finitary winning in omega-regular games. *Trans. Comput. Log.*, 11(1):1:1–1:27, 2009. doi:10.1145/1614431.1614432.

8 Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdzinski. Mean-payoff parity games. In *LICS 2005*, pages 178–187. IEEE Computer Society, 2005. doi:10.1109/LICS.2005.26.

9 Anne Condon. On algorithms for simple stochastic games. In *Advances in Computational Complexity Theory*, pages 51–73. American Mathematical Society, 1993.

10 Laure Daviaud, Marcin Jurdzinski, and Ranko Lazic. A pseudo-quasi-polynomial algorithm for solving mean-payoff parity games. In *LICS 2018*, page (to appear), 2018.

11 E. Allen Emerson and Charanjit S. Jutla. Tree automata, μ-calculus and determinacy. In *FOCS 1991*, pages 368–377. IEEE Computer Society, 1991.

12 E. Allen Emerson and Chin-Laung Lei. Efficient model checking in fragments of the propositional μ-calculus. In *LICS 1986*, pages 267–278. IEEE Computer Society, 1986. doi:10.1109/SFCS.1991.185392.

13 John Fearnley, Sanjay Jain, Sven Schewe, Frank Stephan, and Dominik Wojtczak. An ordered approach to solving parity games in quasi polynomial time and quasi linear space. In *SPIN 2017*, pages 112–121. ACM, 2017. doi:10.1145/3092282.3092286.

14 Nathanaël Fijalkow and Martin Zimmermann. Parity and Streett games with costs. *LMCS*, 10(2), 2014. doi:10.2168/LMCS-10(2:14)2014.

15 Marcin Jurdziński. Deciding the winner in parity games is in UP $\cap$ co-UP. *Information Processing Letters*, 68(3):119–124, November 1998. doi:10.1016/S0020-0190(98)00150-1.

16 Marcin Jurdziński. Small progress measures for solving parity games. In *STACS 2000*, volume 1770 of *LNCS*, pages 290–301, 2000. doi:10.1007/3-540-46541-3_24.

17 Marcin Jurdziński and Ranko Lazić. Succinct progress measures for solving parity games. In *LICS 2017*, pages 1–9. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005092.

18 Marcin Jurdziński, Mike Paterson, and Uri Zwick. A deterministic subexponential algorithm for solving parity games. *SIAM J. on Comp.*, 38(4):1519–1532, 2008. doi:10.1137/070686652.

19 Dexter Kozen. Results on the propositional μ-calculus. *Theo. Comp. Sci.*, 27:333–354, 1983. doi:10.1016/0304-3975(82)90125-6.

20 Karoliina Lehtinen. A modal μ perspective on solving parity games in quasipolynomial time. In *LICS 2018*, page (to appear), 2018.

21 Robert McNaughton. Infinite games played on finite graphs. *Ann. Pure Appl. Logic*, 65(2):149–184, 1993. doi:10.1016/0168-0072(93)90036-D.

22 Fabio Mogavero, Aniello Murano, and Loredana Sorrentino. On promptness in parity games. *Fundam. Inform.*, 139(3):277–305, 2015. URL: https://doi.org/10.3233/FI-2015-1235, doi:10.3233/FI-2015-1235.

23 Anil Nerode, Jeffrey B. Remmel, and Alexander Yakhnis. Mcnaughton games and extracting strategies for concurrent programs. *Ann. Pure Appl. Logic*, 78(1-3):203–242, 1996. doi:10.1016/0168-0072(95)00032-1.

24 Anuj Puri. *Theory of hybrid systems and discrete event systems*. PhD thesis, Computer Science Department, University of California, Berkeley, 1995.

25 Sven Schewe. An optimal strategy improvement algorithm for solving parity and payoff games. In *CSL 2008*, volume 5213 of *LNCS*, pages 368–383, 2008. doi:10.1007/978-3-540-87531-4_27.

26 Sven Schewe. Solving parity games in big steps. *J. of Comp. and Sys. Sci.*, 84:243–262, 2017. doi:10.1016/j.jcss.2016.10.002.

27 Sven Schewe, Ashutosh Trivedi, and Thomas Varghese. Symmetric strategy improvement. In *ICALP 2015*, volume 9135 of *LNCS*, pages 388–400, 2015. doi:10.1007/978-3-662-47666-6_31.

28 Sven Schewe, Alexander Weinert, and Martin Zimmermann. Parity games with weights. *CoRR*, abs/1804.06168, 2018. URL: http://arxiv.org/abs/1804.06168, arXiv:1804.06168.

29 Jens Vöge and Marcin Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *CAV 2000*, pages 202–215. Springer, 2000. doi:10.1007/10722167_18.

30 Alexander Weinert and Martin Zimmermann. Easy to win, hard to master: Optimal strategies in parity games with costs. *LMCS*, 13(3), 2017. doi:10.23638/LMCS-13(3:29)2017.

31 Wieslaw Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theo. Comp. Sci.*, 200(1-2):135–183, 1998. doi:http://dx.doi.org/10.1016/S0304-3975(98)00009-7.

32 Uri Zwick and Mike S. Paterson. The complexity of mean payoff games on graphs. *Theo. Comp. Sci.*, 158(1–2):343–359, 1996. doi:10.1016/0304-3975(95)00188-3.