
PARITY GAMES WITH WEIGHTS

SVEN SCHEWE, ALEXANDER WEINERT, AND MARTIN ZIMMERMANN

University of Liverpool, Liverpool L69 3BX, United Kingdom
e-mail address: sven.schewe@liverpool.ac.uk

German Aerospace Center (DLR), Intelligent and Distributed Systems,
Linder Höhe, 51147 Köln, Germany
e-mail address: alexander.weinert@dlr.de

University of Liverpool, Liverpool L69 3BX, United Kingdom
e-mail address: martin.zimmermann@liverpool.ac.uk

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 labeled 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 $NP \cap co-NP$, the signature complexity for games of this type. We also show that the protagonist has finite-state winning strategies, and provide tight quasi-polynomial bounds for the memory he needs to win the game. Naturally, the antagonist may need infinite memory to win. Moreover, we present tight bounds on the quality of winning strategies for the protagonist.

Furthermore, we investigate the problem of determining, for a given threshold b , whether the protagonist has a strategy of quality at most b and show this problem to be $EXPTIME$ -complete. The protagonist inherits the necessity of exponential memory for implementing such strategies from the special case of finitary parity games.

Key words and phrases: Infinite Games, Quantitative Games, Parity Games.

Supported by the EPSRC projects ‘Energy Efficient Control’ (EP/M027287/1) and ‘Solving Parity Games in Theory and Practice’ (EP/P020909/1).

Supported by the project “TriCS” (ZI 1516/1-1) of the German Research Foundation (DFG) and the Saarbrücken Graduate School of Computer Science. Major parts of this work were performed while the author was employed with the Reactive Systems Group at Saarland University, Germany.

1. INTRODUCTION

Finite games of infinite duration offer a wealth of challenges and applications that has garnered a lot of attention. The traditional class of games under consideration were games with a simple parity [2, 3, 12–14, 16, 17, 19–22, 24, 28–31, 34] or payoff [1, 16, 27, 30, 35] objective. These games form a hierarchy with very simple tractable reductions from parity games through mean-payoff games [1, 16, 27, 30, 35] and discounted-payoff games [16, 30, 35] to simple stochastic games [9].

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, minimizing the bound b itself is also interesting: As the bounds limit the response time, small bounds translate to high quality guarantees [32].

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 [15]. 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 article, 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 amplitude of the accumulated weight on 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., $\text{NP} \cap \text{co-NP}$, 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).

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.

Solving parity games with weights amounts to determining whether there exists a bound b such that the protagonist is able to enforce a play in which the amplitude of the accumulated weight between almost all requests and their corresponding answer is bounded by b . The value of b , however, may be arbitrarily large, subject to the bounds on the quality of winning strategies. Hence, it is natural to consider the threshold problem for parity games with weights: “Given a parity game with weights \mathcal{G} and a bound $b \in \mathbb{N}$, is the protagonist able to enforce that the paths between almost all requests and their respective answers have an amplitude of at most b ?” It is known that the threshold problem for finitary parity games and parity games with costs is PSPACE-complete [32]. In this work, we show that the complexity increases even further in the case of parity games with weights, as the threshold problem for such games is EXPTIME-complete. The complexity of strategies witnessing the answer to the threshold problem, however, does not increase: Both players require exponential memory in order to ensure or violate the bound b , respectively, if they are able to do so at all.

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). For pronomial convenience, we refer to Player 0 as he, and to Player 1 as she. The size of \mathcal{A} , denoted by $|\mathcal{A}|$, is defined as $|V|$. A **play** in \mathcal{A} is an infinite path $\rho = v_0 v_1 v_2 \cdots$ 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** $\mathcal{G} = (\mathcal{A}, \text{Win})$ consists of an arena \mathcal{A} with vertex set V and a set $\text{Win} \subseteq V^\omega$ of winning plays for Player 0. The set of winning plays for Player 1 is $V^\omega \setminus \text{Win}$. A winning condition Win is 0-extendable if, for all $\rho \in V^\omega$ and all $w \in V^*$, $\rho \in \text{Win}$ implies $w\rho \in \text{Win}$. Dually, Win is 1-extendable if, for all $\rho \in V^\omega$ and all $w \in V^*$, $\rho \notin \text{Win}$ implies $w\rho \notin \text{Win}$. Finally, Win is prefix-independent, if it is both 0-extendable and 1-extendable.

A **strategy** for Player $i \in \{0, 1\}$ is a mapping $\sigma: V^*V_i \rightarrow V$ 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 $v_0 v_1 v_2 \cdots$ 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 $\text{Attr}_i(X) = \text{Attr}_i^{|V|}(X)$, where $\text{Attr}_i^0(X) = X$ and

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

Hence, $\text{Attr}_i(X)$ 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 $\text{Attr}_i(X)$ 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 $|E|$ [26]. When we want to stress the arena \mathcal{A} the attractor is computed in, we write $\text{Attr}_i^{\mathcal{A}}(X)$.

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 2.1.

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

A **memory structure** $\mathcal{M} = (M, \text{init}, \text{upd})$ for an arena (V, V_0, V_1, E) consists of a finite set M of memory states, an initialization function $\text{init}: V \rightarrow M$, and an update function $\text{upd}: M \times E \rightarrow M$. The update function can be extended to finite play prefixes in the usual way: $\text{upd}^+(v) = \text{init}(v)$ and $\text{upd}^+(wvv') = \text{upd}(\text{upd}^+(wv), (v, v'))$ for $w \in V^*$ and $(v, v') \in E$. A next-move function $\text{Nxt}: V_i \times M \rightarrow V$ for Player i has to satisfy $(v, \text{Nxt}(v, m)) \in E$ for all $v \in V_i$ and $m \in M$. It induces a strategy σ for Player i with memory \mathcal{M} via $\sigma(v_0 \cdots v_j) = \text{Nxt}(v_j, \text{upd}^+(v_0 \cdots v_j))$. 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.

3. PARITY GAMES WITH WEIGHTS

Fix an arena $\mathcal{A} = (V, V_0, V_1, E)$. A **weighting** for \mathcal{A} is a function $w: E \rightarrow \mathbb{Z}$. 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) $\pi = v_0v_1v_2 \cdots$, we define the amplitude of π as $\text{Ampl}(\pi) = \sup_{j < |\pi|} |w(v_0 \cdots v_j)| \in \mathbb{N}_\infty$.

A **coloring** of V is a function $\Omega: V \rightarrow \mathbb{N}$. 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 $\text{Ans}(c) = \{c' \in \mathbb{N} \mid c' \geq c \text{ and } c' \text{ is even}\}$ 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 [15]. There, the edges of the arena are labeled with *non-negative weights* and the winning condition demands that there

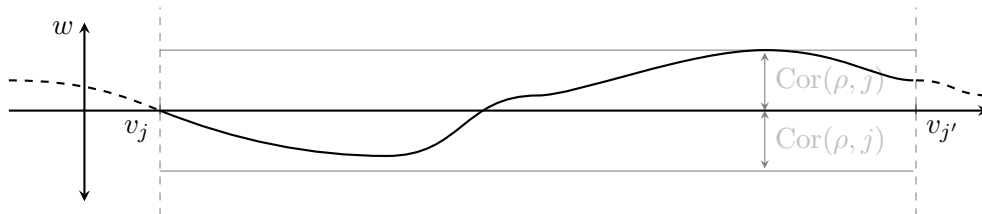


Figure 1: The cost-of-response of some request posed by visiting vertex v_j , which is answered by visiting vertex $v_{j'}$.

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.¹ To distinguish between the parity condition with costs and the extension introduced here, we call our extension the parity condition with weights.

Formally, let $\rho = v_0v_1v_2 \cdots$ be a play. We define the cost-of-response at position $j \in \mathbb{N}$ of ρ by

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

where we use $\min \emptyset = \infty$. As the amplitude of an infix only increases by extending the infix, $\text{Cor}(\rho, j)$ 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 $\text{Cor}(\rho, j) = b$. 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 $\text{Cor}(\rho, j) = \infty$.

We define the parity condition with weights as

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

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 $\mathcal{G} = (\mathcal{A}, \text{WeightParity}(\Omega, w))$ 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 $\text{WeightParity}(\Omega, w)$ is a classical (max-) parity condition, denoted by $\text{Parity}(\Omega)$. Similarly, if w assigns positive weights to every edge, then $\text{WeightParity}(\Omega, w)$ 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 $\text{WeightParity}(\Omega, w)$ is a parity condition with costs, as introduced by Fijalkow and Zimmermann [15]. In these cases, we refer to \mathcal{G} as a parity game, a finitary

¹We discuss other possible interpretations of negative weights in Section 10.

	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 [15, 25]	quasi-poly.	pos./inf.	$\mathcal{O}(nW)$

Table 1: Characteristic properties of variants of parity games.

parity game, or a parity game with costs, respectively. Dually, every parity game, finitary parity game, and parity game with costs is a parity game with weights. We recall the characteristics of these special cases in Table 2.

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 [15]: 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, we provide a polynomial time reduction from solving energy parity games to solving parity games with weights in Section 8. We 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, as they are in $\text{NP} \cap \text{co-NP}$ due to Chatterjee and Doyen [4]. Moreover, they are LOGSPACE -equivalent to mean-payoff parity games as introduced by Chatterjee, Henzinger, and Jurdziński [8]. Hence they can be solved in pseudo-quasi-polynomial time, due to recent advances by Daviaud, Jurdziński, and Lazić [10].

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:

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

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 $\text{BndWeightParity}(\Omega, w) \subseteq \text{WeightParity}(\Omega, w)$ as well as that $\text{WeightParity}(\Omega, w)$ is 0-extendable. Hence, if Player 0 has a strategy from a vertex v such that every consistent play has a suffix in $\text{BndWeightParity}(\Omega, w)$, then the strategy is winning for her from v

w.r.t. $\text{WeightParity}(\Omega, w)$. Thus,

$$\text{Attr}_0(\mathcal{W}_0(\mathcal{A}, \text{BndWeightParity}(\Omega, w))) \subseteq \mathcal{W}_0(\mathcal{A}, \text{WeightParity}(\Omega, w)) .$$

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

Proof. As bounded parity conditions with weights are Borel, bounded parity games with weights are determined [23]. Hence, $\mathcal{W}_0(\mathcal{G}') = \emptyset$ implies that, for every vertex v of \mathcal{A} , Player 1 has a strategy τ_v that is winning in \mathcal{G}' from v .

We combine these strategies into a single strategy τ for Player 1 that is winning in \mathcal{G} from every vertex of \mathcal{A} . This strategy is controlled by a vertex v^* (initialized with the starting vertex of the play) and a counter κ ranging over \mathbb{N} (initialized with zero). The strategy τ mimics the strategy τ_{v^*} from v^* until a request is followed by an infix without an answer and with amplitude κ . This implies that the cost-of-response of this request is at least κ . If such a situation is encountered, then v^* is set to the current vertex and κ is incremented. Furthermore, the history of the play is discarded at this point in the play, and τ behaves henceforth like τ_{v^*} when starting at v^* when this happens.

We now show that τ is winning for Player 1 from every vertex in \mathcal{G} . Consider a play ρ that is consistent with this strategy. If, on the one hand, κ is updated infinitely often along ρ , then ρ contains, for every $b \in \mathbb{N}$, a request that has a cost-of-response that is larger than b . Hence, it violates the parity condition with weights.

If, on the other hand, κ is only updated finitely often, then ρ has a suffix ρ' that starts in some v , which is consistent with τ_v . As τ_v is winning for Player 1 from v in \mathcal{G}' , ρ' violates the bounded parity condition with weights. Also, because κ is updated only finitely often during the suffix, there is a bound b such that the amplitude of every suffix of ρ' that starts at a request is bounded by b . Hence, the only way for ρ' to violate the bounded parity condition with weights is to violate the parity condition. Thus, the full play ρ also violates the parity condition, and therefore also the parity condition with weights, which is a strengthening of the parity condition. Therefore, τ is indeed winning for Player 1 from every vertex in \mathcal{G} . \square

Lemma 4.1 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.

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 4.1 and is similar to the ones for finitary parity games [7] and for parity games with costs [15].

Lemma 4.2. *Algorithm 1 returns $\mathcal{W}_0(\mathcal{A}, \text{WeightParity}(\Omega, w))$*

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

Input: Parity game with weights \mathcal{G} with arena \mathcal{A} , coloring Ω and weighting w

- 1: $k = 0$; $W_0^k = \emptyset$; $\mathcal{A}_k = \mathcal{A}$
 - 2: **repeat**
 - 3: $k = k + 1$
 - 4: $X_k = \mathcal{W}_0(\mathcal{A}_{k-1}, \text{BndWeightParity}(\Omega, w))$
 - 5: $W_0^k = W_0^{k-1} \cup \text{Attr}_0^{\mathcal{A}_{k-1}}(X_k)$
 - 6: $\mathcal{A}_k = \mathcal{A}_{k-1} \setminus \text{Attr}_0^{\mathcal{A}_{k-1}}(X_k)$
 - 7: **until** $X_k = \emptyset$
 - 8: **return** W_0^k
-

Proof. Let $\mathcal{G} = (\mathcal{A}, \text{WeightParity}(\Omega, w))$ and let k^* be the final iteration when running the algorithm on \mathcal{G} , i.e., its output is $W_0^{k^*} = \bigcup_{0 < k < k^*} \text{Attr}_0^{\mathcal{A}_{k-1}}(X_k)$. First, we consider Player 0 and show $W_0^{k^*} \subseteq \mathcal{W}_0(\mathcal{G})$. For every vertex v that is in some X_k , Player 0 has a strategy σ_v for $\mathcal{G}_k = (\mathcal{A}_{k-1}, \text{BndWeightParity}(\Omega, w))$ that is winning from v . Furthermore, for every attractor $\text{Attr}_0^{\mathcal{A}_{k-1}}(X_k)$, he has a positional attractor strategy σ_k . Now, we compose these strategies to a strategy σ for Player 0 in \mathcal{A} via

$$\sigma(v_0 \cdots v_j) = \begin{cases} \sigma_k(v_j) & \text{if } v_j \in \text{Attr}_0^{\mathcal{A}_{k-1}}(X_k) \setminus X_k, \\ \sigma_{v_{j'}}(v_{j'} \cdots v_j) & \text{if } v_j \in X_k. \end{cases}$$

In the second case, $v_{j'} \cdots v_j$ is the longest suffix of $v_0 \cdots v_j$ that only contains vertices from X_k , the set of vertices from which Player 0 has a winning strategy for \mathcal{G}_k .

Consider a play $\rho = v_0 v_1 v_2 \cdots$ in \mathcal{A} that starts in $W_0^{k^*}$ and that is consistent with σ .

For every j there is a unique k_j in the range $0 < k_j < k^*$ such that $v_j \in \text{Attr}_0^{\mathcal{A}_{k_j-1}}(X_{k_j})$. As $\text{BndWeightParity}(\Omega, w)$ is 1-extendable, Items 2 and 3 of Remark 2.1 imply that each $\text{Attr}_0^{\mathcal{A}_{k-1}}(X_k)$ is a trap for Player 1 in \mathcal{A}_{k-1} . Hence, we obtain $k_0 > k_1 > k_2 > \cdots$. As the k_j are always greater than zero, the sequence stabilizes eventually. This implies that ρ has a suffix $\rho' = v_j v_{j+1} v_{j+2} \cdots$ that is consistent with σ_{v_j} .

Hence, due to σ_{v_j} being a winning strategy for Player 0 in \mathcal{G}_k from v_j , we obtain $\rho' \in \text{BndWeightParity}(\Omega, w)$. Hence, $\rho \in \text{WeightParity}(\Omega, w)$, since the bounded parity condition with weights is a strengthening of its unbounded variant and due to 0-extendability of $\text{WeightParity}(\Omega, w)$. Hence, σ is indeed winning from $W_0^{k^*}$.

Now, consider Player 1. We show $V \setminus W_0^{k^*} \subseteq \mathcal{W}_1(\mathcal{G})$. Then, determinacy of parity games with weights (due to their winning conditions being Borel [23]) yields $W_0^{k^*} = \mathcal{W}_0(\mathcal{G})$ and $V \setminus W_0^{k^*} = \mathcal{W}_1(\mathcal{G})$.

Due to X_{k^*} being empty and bounded parity games with weights being determined (again due to their winning conditions being Borel), Player 1 wins the bounded parity game with weights \mathcal{G}_{k^*} from every vertex. Applying Lemma 4.1 shows that she also wins the parity game with weights $(\mathcal{A}_{k^*-1}, \text{WeightParity}(\Omega, w))$ from every vertex. Finally, as $V \setminus W_0^{k^*} = \mathcal{W}_1(\mathcal{A}_{k^*-1}, \text{WeightParity}(\Omega, w))$ is a trap for Player 0 in \mathcal{A} by construction, he also wins $\mathcal{G} = (\mathcal{A}, \text{WeightParity}(\Omega, w))$ from every vertex in $V \setminus W_0^{k^*}$. \square

The winning strategy for Player 0 defined in the proof of Lemma 4.2 can be implemented by a memory structure of size $\max_{k < k^*} s_k$, 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^*

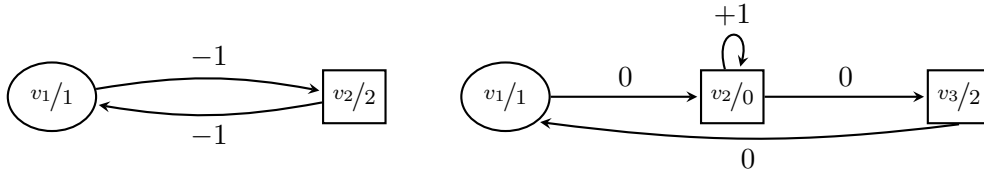


Figure 2: The difference between energy parity games and parity games with weights.

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].

Similar 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 accumulated weight of every prefix, 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.

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 reversed: 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. Finally, Sections 5.3.1 and 5.3.2 are dedicated to the correctness proof of the construction.

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 $\Omega: V \rightarrow \mathbb{N}$ of V , and an edge weighting $w: E \rightarrow \mathbb{Z}$ 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

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

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 $\mathcal{G}_{c_0} = (\mathcal{A}, \text{EnergyParity}_{c_0}(\Omega, w))$ 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, a priori, 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 [4] 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 5.1 ([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 5.1 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 5.2 ([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 via a straightforward pumping argument.

Lemma 5.3. *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 $w(\pi) > -Wns$.*

Proof. Let σ be implemented by $\mathcal{M} = (M, \text{init}, \text{upd})$ and let $\rho = v_0 v_1 v_2 \dots$. We assume towards a contradiction that there is an infix $\pi = v_j \dots v_{j'}$ with $w(\pi) \leq -Wns$.

We obtain a lower bound of $ns + 1$ on the number of different energy levels attained during π (as W is the maximal absolute value of the occurring weights). Moreover, we obtain $ns + 1$ non-empty prefixes of π with (1) increasing length and (2) strictly decreasing energy levels.

Thus, there are positions j_0, j_1 with $j \leq j_0 < j_1 \leq j'$ with $v_{j_0} = v_{j_1}$, $\text{upd}^+(v_0 \dots v_{j_0}) = \text{upd}^+(v_0 \dots v_{j_1})$, and $w(v_{j_0} \dots v_{j_1}) < 0$. Hence, the play $v_0 \dots v_{j_0-1} (v_{j_0} \dots v_{j_1-1})^\omega$ obtained by repeating the loop between v_{j_0} and v_{j_1} ad infinitum is consistent with σ and violates the energy condition. However, it starts in a vertex, from which σ is a winning strategy for Player 0 in the energy parity game. This yields the desired contradiction. \square

Moreover, Chatterjee and Doyen gave an upper bound on the complexity of solving energy parity games, which was recently supplemented by Daviaud et al. [10] with an algorithm solving them in pseudo-quasi-polynomial time.

Proposition 5.4 ([4,10]). *The following problem is in $\text{NP} \cap \text{CO-NP}$ 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 $\mathcal{G} = (\mathcal{A}, \text{BndWeightParity}(\Omega, w))$ 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 the energy parity game \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 $\Omega(v) \notin \text{Ans}(\Omega(v^*))$ 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 $\Omega(v) \in \text{Ans}(\Omega(v^*))$ and every polarity $p \in P$, E' contains the self-loop $((v, p), (v, p))$.²

²This definition introduces some terminal vertices, i.e., those of the form $((v, v'), p, i)$ with $\Omega(v) \in \text{Ans}(\Omega(v^*))$. However, these vertices also have no incoming edges. Hence, to simplify the definition, we just ignore them.

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

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

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 violates the parity condition) or an answer to the color of v^* is reached (and thereby satisfies 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 it is, broadly speaking, beneficial for Player 1 to move along edges of negative weight in an energy parity game, 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 $\Omega(v) \in \text{Ans}(\Omega(v^*))$ have weight zero. Combined with the fact that these vertices have color zero, this allows 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 $\Omega(v^*) \in \text{Ans}(\Omega(v^*))$.

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^*} .

Example 5.5. 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

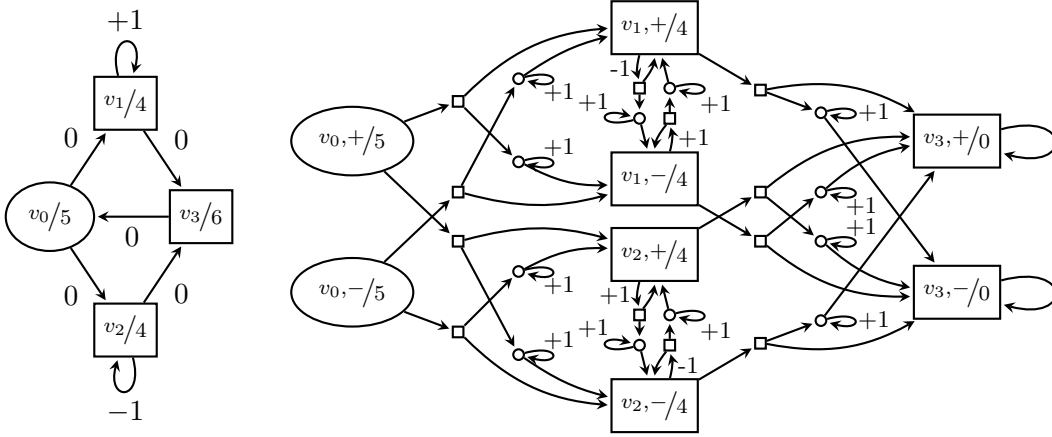


Figure 3: A bounded parity game with weights \mathcal{G} (left) and the associated energy parity game \mathcal{G}_{v_0} (right). The unnamed vertices of Player 1 (Player 0) are of the form $((v, v'), p, 1)$ (of the form $((v, v'), p, 0)$) when between the vertices (v, p) and (v', p') . All missing edge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

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 of 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 4.1: If Player 0 wins \mathcal{G}_v from $(v, +)$ for every v , then she also wins \mathcal{G} 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 5.6. *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 $\mathcal{W}_1(\mathcal{G}) = \emptyset$.*

Before we prove this lemma, we first note that it 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 Algorithm 2.

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

Input: Bounded parity game with weights \mathcal{G} with arena \mathcal{A} , coloring Ω and weighting w

- 1: $k = 0$; $W_1^k = \emptyset$; $\mathcal{A}_k = \mathcal{A}$
- 2: **repeat**
- 3: $k = k + 1$
- 4: $X_k = \{v^* \mid \text{Player 1 wins the energy parity game } ((\mathcal{A}_{k-1})_{v^*}, \Omega_{v^*}, w_{v^*}) \text{ from } (v^*, +)\}$
- 5: $W_1^k = W_1^{k-1} \cup \text{Attr}_1^{\mathcal{A}_{k-1}}(X_k)$
- 6: $\mathcal{A}_k = \mathcal{A}_{k-1} \setminus \text{Attr}_1^{\mathcal{A}_{k-1}}(X_k)$
- 7: **until** $X_k = \emptyset$
- 8: **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 5.6. We only need two further properties: the 1-extendability of $\text{BndWeightParity}(\Omega, w)$, and an assertion that $\text{Attr}_1^{\mathcal{A}_{k-1}}(X_k)$ is a trap for Player 0 in \mathcal{A}_{k-1} . Both are easy to verify.

After plugging Algorithm 2 into Algorithm 1, Proposition 5.4 yields our main theorem, settling the complexity of solving parity games with weights.

Theorem 5.7. *The following problem is in $\text{NP} \cap \text{CO-NP}$ 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 ?”

It remains to prove Lemma 5.6. We do so in the following section.

5.3. Proof of Lemma 5.6. We prove the two assertions of Lemma 5.6 separately from each other: We first show Item 1 of Lemma 5.6, before continuing to show Item 2. In order to prepare for this, however, we first introduce some notation. Let $v^* \in V$ and consider \mathcal{G}_{v^*} . We distinguish three types of plays in \mathcal{G}_{v^*} :

Type -1: Plays that have a suffix $(e, p, 0)^\omega$ for some $e \in E$ and some $p \in P$.

Type 0: Plays that visit infinitely many vertices from both $V \times P$ and $E \times P \times \{0, 1\}$.

Type 1: Plays that have a suffix $(v, p)^\omega$. Note that this implies $\Omega(v) \in \text{Ans}(\Omega(v^*))$.

Clearly, plays of Type -1 are losing for Player 0 due to the coloring of \mathcal{G}_{v^*} labeling vertices of the form $(e, p, 0)$ with the odd color one. Dually, plays of Type 1 are losing for Player 1, since $\Omega(v) \in \text{Ans}(\Omega(v^*))$ implies that (v, p) carries color zero and its only outgoing edge is a self-loop of weight 0. We formalize this observation in the following remark.

Remark 5.8. *Let ρ' be a play in \mathcal{G}_{v^*} that starts in (v^*, p) .*

- (1) *If ρ' is consistent with a winning strategy for Player 0 from (v^*, p) , then ρ' is not a play of Type -1.*
- (2) *If ρ' is consistent with a winning strategy for Player 1 from (v^*, p) , then ρ' is not a play of Type 1.*

In order to remove the added vertices of the form $E \times P \times \{0, 1\}$ from plays in \mathcal{G}_{v^*} , we define the homomorphism $\text{unpol}: (V')^* \cup (V')^\omega \rightarrow V^* \cup V^\omega$ induced by $\text{unpol}(v, p) = v$ and $\text{unpol}(e, p, i) = \varepsilon$ for $v \in V$, $e \in E$, $p \in P$, and $i \in \{0, 1\}$. Let $\rho' \in (V')^* \cup (V')^\omega$. We call $\text{unpol}(\rho')$ the **unpolarization** of ρ' .

Remark 5.9. *Let ρ' be a play of Type 0 in some \mathcal{G}_{v^*} . We have $\rho' \in \text{Parity}(\Omega_{v^*})$ if and only if $\text{unpol}(\rho') \in \text{Parity}(\Omega)$.*

5.3.1. *Proof of Item 1 of Lemma 5.6.* Recall that we need to show that Player 1 wins the bounded parity game with weights \mathcal{G} from v^* if she wins the energy-parity game \mathcal{G}_{v^*} from $(v^*, +)$. Thus, let τ_{v^*} be a winning strategy for Player 1 from $(v^*, +)$ in \mathcal{G}_{v^*} . We define a winning strategy τ for her from v^* in \mathcal{G} such that τ mimics the moves made by τ_{v^*} . To this end, τ keeps track of a play prefix \mathcal{G}_{v^*} . Formally, we define τ together with a simulation function h that satisfies the following invariant:

If π is a non-empty play prefix in \mathcal{A} that starts in v^* , is consistent with τ , and ends in some v , then $h(\pi)$ is a play prefix in \mathcal{A}_{v^*} that starts in $(v^*, +)$, is consistent with τ_{v^*} , and ends in some (v, p) . Furthermore, $\text{unpol}(h(\pi)) = \pi$.

Recall that, if h has the properties described above, then, due to the structure of \mathcal{A}_{v^*} , for each π , given $h(\pi)$, the strategy τ_{v^*} prescribes a move to some vertex $((v, v'), p, 1)$, where $(v, v') \in E$. We can mimic this choice by moving to v' in \mathcal{G} .

We now define h and τ formally and begin with $h(v^*) = (v^*, +)$, which clearly satisfies the invariant. Now let $\pi = v_0 \cdots v_j$ be some non-empty play prefix in \mathcal{A} beginning in v^* and consistent with τ such that $h(\pi)$ is defined. Due to the invariant, $h(\pi)$ ends in (v_j, p_j) for some $p_j \in P$.

If $v_j \in V_1$, let v_{j+1} be such that $h(\pi) \cdot ((v_j, v_{j+1}), p_j, 1)$ is consistent with τ_{v^*} and define $\tau(\pi) = v_{j+1}$. Such a v_{j+1} exists, because (v_j, p_j) , the last vertex of $h(\pi)$, satisfies $\Omega(v_j) \notin \text{Ans}(\Omega(v^*))$ due to the invariant, Item 2 of Remark 5.8, and because the answering vertices are sinks. If, however, $v_j \in V_0$, then let v_{j+1} be an arbitrary successor of v_j in \mathcal{A} . In either case, it remains to define $h(\pi \cdot v_{j+1})$.

Since we want to simulate the move from v_j to v_{j+1} in $h(\pi \cdot v_{j+1})$, we first move from (v_j, p_j) to $((v_j, v_{j+1}), p_j, 1)$. Moreover, in order to satisfy the invariant, we aim to simulate the play prefix $\pi \cdot v_{j+1}$ such that $h(\pi \cdot v_{j+1})$ is consistent with τ_{v^*} . This strategy may prescribe for Player 1 to either preserve the polarity p_j , or to switch it during the simulated move from v_j to v_{j+1} .

In the former case, i.e., if $\tau_{v^*}(h(\pi) \cdot ((v_j, v_{j+1}), p_j, 1)) = (v_{j+1}, p_j)$, we define

$$h(\pi \cdot v_{j+1}) = h(\pi) \cdot ((v_j, v_{j+1}), p_j, 1) \cdot (v_{j+1}, p_j) .$$

In the latter case, Player 0 gets an opportunity to recharge the energy by taking the self-loop of the vertex $((v_j, v_{j+1}), p_j, 0)$ finitely often. We opt to let her recover the energy lost so far in the play prefix, i.e., we pick $c_j = \max\{0, -w(h(\pi) \cdot ((v_j, v_{j+1}), p_j, 1))\}$ and define

$$h(\pi \cdot v_{j+1}) = h(\pi) \cdot ((v_j, v_{j+1}), p_j, 1) \cdot ((v_j, v_{j+1}), p_j, 0)^{c_j+1} \cdot (v_{j+1}, \bar{p}_j)$$

in this case. Since $h(\pi \cdot v_{j+1})$ is consistent with τ_{v^*} in either case, we satisfy the invariant in either case. This completes the definition of τ and h .

It remains to show that τ is indeed winning from v^* in \mathcal{G} . To this end, let $\rho = v_0 v_1 v_2 \cdots$ be a play in \mathcal{A} that starts in v^* and that is consistent with τ . We show $\rho \notin \text{BndWeightParity}(\Omega, w)$ by examining the play ρ' in \mathcal{A}_{v^*} , which is the limit of the $h(\pi)$ for increasing prefixes π of ρ . Due to the invariant, ρ' starts in $(v^*, +)$ and is consistent with τ_{v^*} . Moreover, due to the construction of h , we obtain $\text{unpol}(\rho') = \rho$. Finally, we have that ρ' is a play of Type 0 in \mathcal{G}_{v^*} . Hence, due to Remark 5.9, ρ satisfies the parity condition if and only if ρ' satisfies the parity condition.

As the play ρ' is consistent with the winning strategy τ_{v^*} for Player 1, we have $\rho' \notin \text{EnergyParity}(\Omega_{v^*}, w_{v^*})$, i.e., ρ' either violates the parity condition or the energy condition. Hence, as argued above, if ρ' violates the parity condition, then so does ρ , i.e., ρ is indeed winning for Player 1.

Now assume that ρ' violates the energy condition. Due to the structure of \mathcal{A}_{v^*} and the construction of h we have

$$\rho' = \prod_{j=0,1,2,\dots}(v_j, p_j) \cdot ((v_j, v_{j+1}), p_j, 1) \cdot ((v_j, v_{j+1}), p_j, 0)^{m_j}$$

for some $m_j \in \mathbb{N}$. Since ρ' violates the energy condition, we have $\inf_{j \in \mathbb{N}} w((v_0, p_0) \cdots (v_j, p_j) \cdot ((v_j, v_{j+1}), p_j, 1)) = -\infty$. The restriction to play prefixes of this form suffices due to the structure of \mathcal{A}_{v^*} and, in turn, the structure of ρ' . Moreover, since Player 1 wins \mathcal{G}_{v^*} from $(v^*, +)$, the initial vertices v^* and $(v^*, +)$ of ρ and ρ' , respectively, have the same odd color. Also, as ρ' is a play of Type 0, the request for the color $\Omega(v^*)$ is never answered in ρ or ρ' . We show that the request for $\Omega(v^*)$ in ρ is unanswered with infinite cost, which concludes the proof.

To this end, we split ρ' into infixes of constant polarity. Given a vertex $v = (v', p)$ or $v = ((v', v''), p, i)$, we call p the **polarity** of v . Let $\rho' = \mu'_0 \mu'_1 \mu'_2 \cdots$, where each μ'_j is a maximal finite (or infinite) infix (or suffix) of ρ' , such that all vertices in μ'_j have the same polarity. We call an infix μ'_j of ρ' an **equi-polarity infix** (EPI) of ρ' .

Since the polarity remains constant throughout each μ'_j , Player 0 only resets the energy via repeatedly traversing a self-loop of a vertex in \mathcal{A}_{v^*} at the last vertex visited in μ'_j , if at all. Hence, the energy levels attained during μ'_j and $\text{unpol}(\mu'_j)$ are closely related.

Remark 5.10. *Let μ' be an EPI beginning in (v_j, p_j) and let $\mu = \text{unpol}(\mu') = v_j v_{j+1} v_{j+2} \cdots$. For each j' with $j \leq j' < j + |\mu|$, we have $|w(v_j \cdots v_{j'})| = |w((v_j, p_j) \cdots (v_{j'}, p_{j'}))|$.*

In particular, Remark 5.10, the structure of \mathcal{A}_{v^*} , and the definition of h imply that we have $\text{Ampl}(\text{unpol}(\mu')) = \text{Ampl}(\mu')$ for all EPIs μ' of ρ' . Thus, if there exist only finitely many EPIs of ρ' , let μ' be the infinite final EPI of ρ' , let $\mu = \text{unpol}(\mu')$, and note that, due to $\text{Ampl}(\rho') = \infty$, we have $\text{Ampl}(\mu') = \infty$. Due to Remark 5.10, we obtain $\text{Ampl}(\mu) = \infty$, which implies that the request posed at the initial position of ρ is unanswered with infinite cost due to the reasoning above, as μ is a suffix of ρ .

If, however, there exist infinitely many EPIs of ρ' , assume towards a contradiction that the cost of answering the request posed at the initial position of ρ is finite. By construction of ρ' , the energy level is non-negative at the end of each EPI. Since ρ' violates the energy condition, for each bound $b \in \mathbb{N}$ there exists an EPI μ' of ρ' with a prefix of weight strictly smaller than $-b$. We obtain $\text{Ampl}(\text{unpol}(\mu')) > b$ via Remark 5.10. This contradicts the cost of answering the request posed at the initial position of ρ being bounded and concludes the proof of Item 1 of Lemma 5.6.

5.3.2. Proof of Item 2 of Lemma 5.6. To prove Item 2 of Lemma 5.6, we show that Player 0 wins the bounded parity game with weights \mathcal{G} from every vertex, if he wins each energy-parity game \mathcal{G}_{v^*} from $(v^*, +)$. This suffices due to determinacy of \mathcal{G} . To this end, we construct a strategy σ for Player 0 in \mathcal{G} that is winning for him from each vertex $v \in V$. As winning regions are disjoint, this implies the desired result.

For each energy parity game $\mathcal{G}_v = (\mathcal{A}_v, \Omega_v, w_v)$ we have $n' = |\mathcal{A}_v| \in \mathcal{O}(|\mathcal{A}|^2)$, we have $d' = |\Omega_v(V')| = |\Omega(V)| + 2$, and we have $W' = \max(w(E')) = \max(w(E) \cup \{1\})$, where E and E' are the sets of edges in \mathcal{A} and the \mathcal{A}_v , respectively. Note that the values n', d' ,

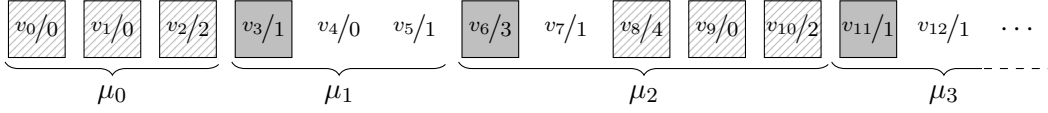


Figure 4: A play ρ , its induced color sequence, its most relevant requests, and the ESIs of ρ .

and W' of \mathcal{G}_v are independent of the vertex v , which explains our notation. Due to the assumption of the statement and Proposition 5.2, for each $v \in V$, there exists a finite-state strategy σ_v with at most $n'd'W'$ states that is winning for Player 0 from $(v, +)$ in \mathcal{G}_v .

We construct the winning strategy σ for Player 0 in \mathcal{G} by “stitching together” the individual σ_v . To this end, given a play prefix, we identify the request which should be answered most urgently. Say this request was opened by visiting vertex v . The strategy σ then mimics the moves made by σ_v when starting in $(v, +)$. Once the request for $\Omega(v)$ is answered, σ makes arbitrary moves until a new request is opened.

Formally, given a play prefix $\pi = v_0 \cdots v_j$, we say that a request for color c is open in π if there exists a position j' with $0 \leq j' \leq j$ such that $\Omega(v_{j'}) = c$ and, for all positions j'' with $j' \leq j'' \leq j$, we have $\Omega(v_{j''}) \notin \text{Ans}(\Omega(v_{j'}))$. Clearly there is never an open request for an even color.

If there is no open request in π , the position of the most relevant request is undefined and we write $\text{posMRR}(\pi) = \perp$. Otherwise, let c be the maximal color for which there is an open request in π . We define $\text{posMRR}(\pi)$ as the smallest position j' , such that the request for color c is open in all prefixes of π of length greater than j' .

As an example, consider the play prefix shown in Figure 4 using the notation $v/\Omega(v)$. We mark a position j with solid background if $\text{posMRR}(v_0 \cdots v_j) = j$ and with dashed background if $\text{posMRR}(v_0 \cdots v_j) = \perp$. Otherwise, i.e., if $\perp \neq \text{posMRR}(v_0 \cdots v_j) < j$, we leave j unmarked. For those positions, $\text{posMRR}(v_0 \cdots v_j)$ is equal to the largest (i.e., last visited) earlier position marked in solid background.

In order to leverage the moves made by the strategies for the \mathcal{G}_v in \mathcal{G} , we need to simulate play prefixes in the latter game in the former ones. To this end, we again define σ together with a simulation function h . This function h maps a play prefix consistent with σ to a sequence of vertices from V' (not necessarily a play prefix) such that we are able to leverage the choices made by the σ_v in order to define σ . Our aim is to define h such that it satisfies the following invariant:

Let $\pi = v_0 \cdots v_j$ be a play in \mathcal{A} consistent with σ . Then $h(\pi)$ ends in some (v_j, p_j) . Moreover, if $\text{posMRR}(\pi) = j' \neq \perp$, then $h(\pi)$ has a (unique) suffix $(v_{j'}, +) \cdots (v_j, p_j)$ that is consistent with $\sigma_{v_{j'}}$ and satisfies $\text{unpol}((v_{j'}, +) \cdots (v_j, p_j)) = v_{j'} \cdots v_j$.

We define h and σ inductively and begin with $h(v) = (v, +)$ for each $v \in V$, which clearly satisfies the invariant. Now let $\pi = v_0 \cdots v_j$ be a nonempty play prefix in \mathcal{A} and consistent with σ such that $h(\pi)$ is defined. We again first determine a successor of v_j , defining $\sigma(\pi)$ along the way if v_j is a vertex of Player 0.

If $v_j \in V_1$, let v_{j+1} be an arbitrary successor of v_j in \mathcal{A} . If, however, $v_j \in V_0$, we distinguish two cases based on whether or not $\text{posMRR}(\pi)$ is defined. If $\text{posMRR}(\pi) = \perp$, again let v_{j+1} be an arbitrary successor of v_j . If, however, $\text{posMRR}(\pi) = j' \neq \perp$, then the invariant of h yields a suffix $(v_{j'}, +) \cdots (v_j, p_j)$ of $h(\pi)$ that is consistent with $\sigma_{v_{j'}}$. Let v_{j+1} be the unique vertex such that $(v_{j'}, +) \cdots (v_j, p_j) \cdot ((v_j, v_{j+1}), p_j, 1)$ is consistent

with $\sigma_{v_{j'}}$. Such a vertex v_{j+1} exists, because the request posed by visiting $v_{j'}$ is open in π due to $\text{posMRR}(\pi) = j'$ and since the color sequences induced by $v_{j'} \cdots v_j$ and $(v_{j'}, +) \cdots (v_j, p_j)$ coincide, save for the irrelevant intermediate vertices of colors zero and one. Hence, (v_j, p_j) is not an accepting sink in $\mathcal{G}_{v_{j'}}$. Since $v_j \in V_0$, the vertex v_{j+1} is unique. In both cases, we define $\sigma(\pi) = v_{j+1}$. This concludes the definition of σ .

It remains to define $h(\pi \cdot v_{j+1})$ such that it satisfies the above invariant. To this end, we use one of two operations. Firstly, we define the **discontinuous extension of $h(\pi)$ with v_{j+1}** as

$$h(\pi) \cdot (v_{j+1}, +) .$$

Secondly, we define a simulated extension of $h(\pi)$ such that we obtain $h(\pi \cdot v_{j+1})$ by simulating the move from v_j to v_{j+1} in some \mathcal{G}_v . Formally, we define the **simulated extension of $h(\pi)$ with v_{j+1} and charge m** as

$$h(\pi) \cdot ((v_j, v_{j+1}), p_j, 1) \cdot ((v_j, v_{j+1}), p_j, 0)^m \cdot (v_{j+1}, p_{j+1}) ,$$

where $p_{j+1} = p_j$ if $m = 0$ and $p_{j+1} = \overline{p_j}$ otherwise. This ensures that the extension is a play infix in some \mathcal{G}_v .

In order to define $h(\pi \cdot v_{j+1})$ we again distinguish whether or not $\text{posMRR}(\pi)$ is defined. If $\text{posMRR}(\pi) = \perp$, we define $h(\pi \cdot v_{j+1})$ to be the discontinuous extension of $h(\pi)$ with v_{j+1} . This clearly satisfies the first condition of the invariant. Moreover, the second condition of the invariant is satisfied as well: If $\text{posMRR}(\pi \cdot v_{j+1}) = \perp$, this condition holds true vacuously. Otherwise, we have $\text{posMRR}(\pi \cdot v_{j+1}) = j+1$ and observe that the suffix $(v_{j+1}, +)$ of $h(\pi \cdot v_{j+1})$ satisfies the second condition of the invariant.

If, however, $\text{posMRR}(\pi) \neq \perp$, then let $\text{posMRR}(\pi) = j'$. By definition of posMRR we have $\text{posMRR}(\pi \cdot v_{j+1}) \in \{\perp, j', j+1\}$. We distinguish two sub-cases and first define $h(\pi \cdot v_{j+1})$ for the case that $\text{posMRR}(\pi \cdot v_{j+1}) \in \{\perp, j+1\}$. In this case, the move to v_{j+1} either answers the most relevant request in π , or the request posed by visiting v_{j+1} is itself the most relevant request of $\pi \cdot v_{j+1}$: We have $\text{posMRR}(\pi \cdot v_{j+1}) = \perp$ in the former case and $\text{posMRR}(\pi \cdot v_{j+1}) = j+1$ in the latter one. In either case, we define $h(\pi \cdot v_{j+1})$ to be the discontinuous extension of $h(\pi)$ with v_{j+1} and observe that the first condition of the invariant holds true. If $\text{posMRR}(\pi \cdot v_{j+1}) = \perp$, the second condition of the invariant holds true vacuously. If, however, $\text{posMRR}(\pi \cdot v_{j+1}) = j+1$, then the suffix $(v_{j+1}, +)$ witnesses that the second condition of the invariant holds true.

Now assume that the move to v_{j+1} neither opens a new most relevant request, nor answers the existing one, i.e., that we have $\text{posMRR}(\pi \cdot v_{j+1}) = j'$. In this case, we extend the suffix of $h(\pi)$ that is consistent with $\sigma_{v_{j'}}$ by simulating the move from v_j to v_{j+1} . Recall that we picked the vertex v_{j+1} such that $(v_{j'}, +) \cdots (v_j, p_j) \cdot ((v_j, v_{j+1}), p_j, 1)$ is consistent with $\sigma_{v_{j'}}$. As we can freely choose whether or not Player 1 switches the polarity in the simulation, we follow the intuition stated during the construction of the polarized arena: Recall that both players currently play “with respect to” the request for $\Omega(v_{j'})$ opened by visiting $v_{j'}$. Hence, we opt to let Player 1 move to positive polarity if the cost of the request for $\Omega(v_{j'})$ so far is nonnegative, and let him move to negative polarity otherwise. To this end, we use the sign function Sgn , which is defined as

$$\text{Sgn}(n) = \begin{cases} + & \text{if } n \geq 0 \text{ ,} \\ - & \text{otherwise .} \end{cases}$$

If $\text{Sgn}(w(v_{j'} \cdots v_{j+1})) = p_j$, we define $h(\pi \cdot v_{j+1})$ to be the simulated extension of $h(\pi)$ with v_{j+1} and charge 0, thus implementing the intuition as described above. Otherwise, i.e., if $\text{Sgn}(w(v_{j'} \cdots v_{j+1})) = \overline{p_j}$, let $m > 0$ such that $(v_{j'}, +) \cdots (v_j, p_j) \cdot ((v_j, v_{j+1}), p_j, 1) \cdot ((v_j, v_{j+1}), p_j, 1)^m \cdot (v_{j+1}, \overline{p_j})$ is consistent with $\sigma_{v_{j'}}$. Such an m exists, as otherwise the play $(v_{j'}, +) \cdots (v_j, p_j) \cdot ((v_j, v_{j+1}), p_j, 1) \cdot ((v_j, v_{j+1}), p_j, 0)^\omega$ of Type -1 that starts in $(v_{j'}, +)$ would be consistent with the winning strategy $\sigma_{v_{j'}}$ from $(v_{j'}, +)$ for Player 0, contradicting Item 1 of Remark 5.8. We define $h(\pi \cdot v_{j+1})$ to be the simulated extension of $h(\pi)$ with v_{j+1} and charge m . Since we have $\text{posMRR}(v_0 \cdots v_{j+1}) = j'$ by assumption, either definition of $h(\pi \cdot v_{j+1})$ satisfies the invariant. This completes the definition of h .

It remains to show that the strategy σ is indeed winning for Player 0 from v^* . To this end, fix some play $\rho = v_0 v_1 v_2 \cdots$ consistent with σ starting in v^* , and let ρ' be the limit of the $h(\pi)$ for increasing prefixes π of ρ . By the construction of h , we obtain $\text{unpol}(\rho') = \rho$. Hence, the play ρ' is of the form $(v_0, p_0) \cdots (v_1, p_1) \cdots (v_2, p_2) \cdots$. We call a position j of ρ a **discontinuity** of ρ if either $j = 0$ or if $h(v_0 \cdots v_j)$ is the discontinuous extension of $h(v_0 \cdots v_{j-1})$ by v_j .

Let j and j' be adjacent discontinuities of ρ . We call the infix $v_j \cdots v_{j'-1}$ of ρ an **equi-strategic infix** (ESI) of ρ . Moreover, if there only exist finitely many discontinuities of ρ , let j^* be its final discontinuity. We call the suffix $v_{j^*} v_{j^*+1} v_{j^*+2} \cdots$ of ρ the **terminal** ESI of ρ .

Remark 5.11. *Let $\nu = v_j v_{j+1} v_{j+2} \cdots$ be an ESI.*

- (1) *If μ is finite, then the infix μ' of ρ' starting at position $|h(v_0 \cdots v_j)|$ and ending at position $|h(v_0 \cdots v_{j+|\mu|-1})|$ starts in $(v_j, +)$, ends in some $(v_{j+|\mu|-1}, p)$, and is consistent with σ_{v_j} .*
- (2) *If μ is infinite, then the suffix μ' of ρ' starting at position $|h(v_0 \cdots v_j)|$ starts in $(v_j, +)$ and is consistent with σ_{v_j} .*

For each position j of ρ we define $\text{ESI}(j) = k$ if the k -th ESI of ρ contains v_j . Moreover, if $\mu = v_j v_{j+1} v_{j+2} \cdots$ is an ESI of ρ , then we call $\Omega(v_j)$ the **characteristic color** of μ . By the construction of h , if the characteristic color of an ESI μ is even, then μ consists only of a single vertex. If, however, the characteristic color c of an ESI μ is odd, then we have $\Omega(v) \leq c$ for all vertices v in μ . Moreover, let c' be the characteristic color of the ESI succeeding μ , if μ is not the terminal ESI of ρ . Due to the construction of h , we have $c' > c$. If c' is even, this observation implies $c' \in \text{Ans}(\Omega(v))$ for all vertices v in μ . As the number of colors in \mathcal{G} is finite, this in turn implies that the number of ESIs between a request and its response (if a response exists at all) is bounded.

Remark 5.12. *Let j be some position in ρ and let $k = \text{ESI}(j)$. Moreover, let d be the number of colors in \mathcal{G} .*

- (1) *If the request at position j is first answered at position j' , then $\text{ESI}(j') < k + d$*
- (2) *If the request at position j is unanswered in ρ , then ρ contains less than $k + d$ many ESIs.*

Proof. 1. Let $\text{ESI}(j') = k'$. If $\Omega(v_j)$ is even, we obtain $k = k'$ and the claim is trivial. Thus, assume that $\Omega(v_j)$ is odd. Since the request at position j is answered in the k' -th ESI, define the sequence $j_0 < \cdots < j_{k'-k}$ inductively such that $j_0 = \text{posMRR}(v_0 \cdots v_j)$ and such that, for each ℓ with $0 \leq \ell \leq k' - k$ the position j_ℓ is the unique position in ρ with $\text{ESI}(j_\ell) \neq \text{ESI}(j_{\ell-1})$. We claim that the requests opened by visiting the v_{j_ℓ} are answered at position j' at the earliest and show this by induction over ℓ .

First consider $\ell = 0$. Since $\Omega(v_j)$ is odd, we obtain $j_0 \neq \perp$ and, hence, that $\Omega(v_{j_0})$ is odd. Assume towards a contradiction that the request at position j_0 is answered at position $j'' < j'$. If $j'' < j$, then this contradicts our choice of j_0 as the position of the most relevant request of the play prefix $v_0 \cdots v_j$: Since $\Omega(v_j)$ is odd, we would then obtain $\text{posMRR}(v_0 \cdots v_j) > j'' > 0 = \text{posMRR}(v_0 \cdots v_j)$. If, however, $j < j'' < j'$, we obtain that the request at position j is answered at position j'' , since $\Omega(v_{j_0}) \geq \Omega(v_j)$ by definition of the most relevant requests. Hence, we obtain $j'' \geq j'$.

Now consider j_ℓ for $0 < \ell \leq k' - k$: Since we have $\Omega(v_{j_\ell}) > \Omega(v_{j_{\ell-1}})$, an answer to the request posed by v_{j_ℓ} would answer the request posed by $v_{j_{\ell-1}}$ as well. Via the induction hypothesis we obtain that v_{j_ℓ} is answered at position j' at the earliest.

Thus, the only possibility to switch the mimicked strategy and enter a new ESI is by visiting a vertex of larger odd color than the vertex at the initial position of the current ESI. Hence, the color of the v_{j_ℓ} is strictly monotonically increasing, which implies $k' < k + d$.

2. Since the request at position j is unanswered, we obtain that $\Omega(v_j)$ is odd. Again define the sequence $j_0 < j_1 < j_2 < \cdots$ inductively such that $j_0 = \text{posMRR}(v_0 \cdots v_j)$ and such that, for each $\ell > 0$, the position j_ℓ is the unique position in ρ with $\text{ESI}(j_\ell) \neq \text{ESI}(j_{\ell-1})$. Due to similar reasoning as in the previous case, we obtain that the request posed at each j_ℓ is unanswered. Thus, the only possibility to enter a new ESI is again by visiting a vertex of higher odd color than the vertex at the initial position of the current ESI, which again implies that the number of ESIs in ρ is bounded by $k + d$. \square

Recall that the bounded parity condition with weights requires the play ρ to not only satisfy the parity condition, but also that the cost of almost all requests is bounded and that there exists no unanswered request with infinite cost in ρ . We first show that ρ satisfies the classical parity condition. In a second step, we then show that there exists a bound on the cost of each (answered or unanswered) request in ρ . The former condition, i.e., that ρ satisfies the parity condition, is in large parts implied by Remark 5.12.

Lemma 5.13. *The play ρ satisfies the parity condition.*

Proof. If ρ contains no unanswered requests, then it vacuously satisfies the parity condition. Hence, let j be the position of such an unanswered request in ρ . Due to Remark 5.12, we obtain that there exist only finitely many ESIs in ρ . Let $\mu = v_{j^*} v_{j^*+1} v_{j^*+2} \cdots$ be the terminal ESI of ρ . By the construction of h , there exists a suffix μ' of ρ' with $\text{unpol}(\mu') = \mu$. Due to Item 2 of Remark 5.11, the suffix μ' begins in $(v_{j^*}, +)$ and is consistent with the winning strategy $\sigma_{v_{j^*}}$ for Player 0 from $(v_{j^*}, +)$ in $\mathcal{G}_{v_{j^*}}$. Moreover, μ' is a play of Type 0 due to μ' being the terminal ESI of ρ and due to being consistent with $\sigma_{v_{j^*}}$. Hence, we obtain that μ satisfies the parity condition via Remark 5.9, which in turn implies that ρ satisfies the parity condition. \square

It remains to show that the costs of requests in ρ are bounded. Recall that we defined $n' = |\mathcal{A}_v|$, d' as the largest color of a vertex in the \mathcal{G}_v , and W' as the largest absolute weight of an edge. We claim that the costs of the most relevant requests in ρ are bounded by $(n'd'W')^2$. This implies that the cost of all requests is bounded: Due to Remark 5.12 we obtain that the number of ESIs between a request and its response, if one exists, is bounded by d . Hence, it suffices to show that each ESI contributes at most a bounded amount to the cost of answering a request.

Lemma 5.14. *Let $\mu = v_j v_{j+1} v_{j+2} \cdots$ be an ESI of ρ . For each j' with $j \leq j' < j + |\mu|$ we have $|w(v_j \cdots v_{j'})| \leq d'(n'd'W')^2$.*

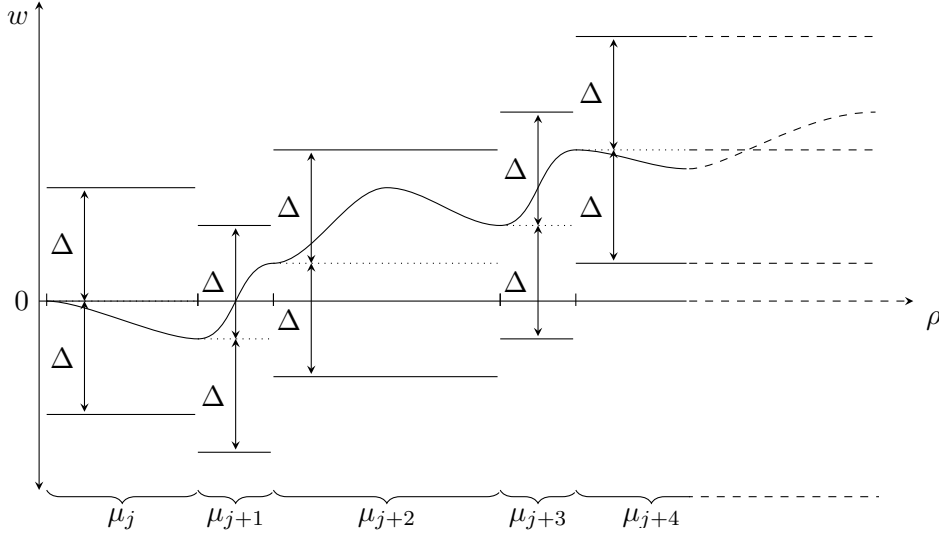


Figure 5: Bounds on the cost of a request over time given by Lemma 5.14. We write $\Delta = d'(n'W')^2$.

Proof. Towards a contradiction, let j' be a position with $j \leq j' < j + |\mu|$, such that we have $|w(v_j \cdots v_{j'})| > d'(n'W')^2$. We assume $w(v_j \cdots v_{j'}) > d'(n'W')^2$, i.e., that the infix $v_j \cdots v_{j'}$ violates the claimed bound from above. The other case is dual. Since each traversed edge adds a cost of at most W' , there exists a minimal position j'' such that, for all k with $j'' \leq k \leq j'$, we have $w(v_j \cdots v_k) > 0$. Let $\pi' = (v_{j''}, p_{j''}) \cdots (v_{j'}, p_{j'})$ be an infix of ρ' such that $\text{unpol}(\pi') = v_{j''} \cdots v_{j'}$. The polarity remains positive throughout π' due to the construction of h , while we have $w_{v_j}(\pi') < -d'(n'W')^2$. Moreover, by definition of ESIs, the infix π' is an infix of a play that starts in $(v_j, +)$ and is consistent with σ_{v_j} . This, however, contradicts Lemma 5.3 on Page 10. \square

Due to Lemma 5.14, each ESI strictly in-between a request and its response contributes at most $d'(n'W')^2$ to the cost incurred by the request. Similarly, the ESI containing the request and its response also contribute at most $d'(n'W')^2$ each to the cost of answering the given request. Hence, via Remark 5.12, we obtain that each (answered or unanswered) request in ρ incurs a cost of at most $(d'n'W')^2$. We illustrate this argument in Figure 5. Hence, σ is a winning strategy for Player 0 from v^* in \mathcal{G} , as each play that starts in v^* and is consistent with σ satisfies the parity condition due to Lemma 5.13 and because no such play contains a request that is unanswered with infinite cost, which concludes the proof of Item 2 of Lemma 5.6.

Before we conclude this section, we formalize the above observation about the winning strategy for Player 0 uniformly bounding the costs of requests in the following corollary.

Corollary 5.15. *Let \mathcal{G} be a bounded parity game with weights with n vertices, d colors, and largest absolute weight W . There exists a strategy σ for Player 0 that is winning from $\mathcal{W}_0(\mathcal{G})$, such that in each play ρ consistent with σ , each request is answered or unanswered with cost at most $((d+2)(2n+4n^2)(W+1))^2$.*

Using arguments from Section 7, this bound can be improved to $((d+2)(6n)(W+1))^2$. However, as we only use Corollary 5.15 later to obtain some upper bound on the quality of such strategies, we refrain from repeating these arguments here.

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 quasi-polynomial bounds on the necessary and sufficient memory for Player 0 to win parity games with weights. In contrast, Player 1 requires infinite memory.

Theorem 6.1. *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} .*

- (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.*
- (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} .*

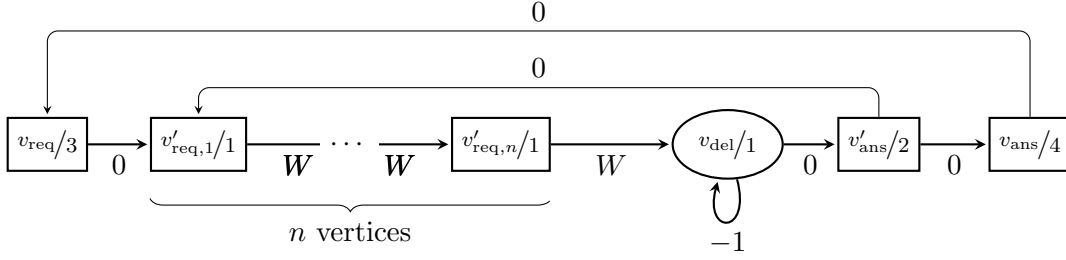
The proof of the second item of Theorem 6.1 is straightforward, since Player 1 already requires infinite memory to implement winning strategies in finitary parity games [7]. Since parity games with weights subsume finitary parity games, this result carries over to our setting. We show the game witnessing this lower bound on the right-hand side of Figure 2 on Page 9.

In contrast, quasi-polynomial memory is sufficient, but also necessary, for Player 0. To show this claim, we first prove that the winning strategy for him in a bounded parity game with weights constructed in the proof of Item 2 of Lemma 5.6 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 6.2. *Let \mathcal{G} be a bounded parity game with weights and let n , d , and W be defined analogously to Theorem 6.1. Player 0 has a finite-state winning strategy of size at most $d(6n)(d+2)(W+1)$ from $\mathcal{W}_0(\mathcal{G})$.*

Proof. Let V and E be the vertex set of \mathcal{G} and recall that we have defined $P = \{+, -\}$. In the proof of Item 2 of Lemma 5.6, we have constructed an energy parity game \mathcal{G}_v with vertices $(V \times P) \cup (E \times P \times \{0, 1\})$ for each vertex v of \mathcal{G} . We have then constructed a winning strategy σ for Player 0 for \mathcal{G} out of winning strategies for him in the \mathcal{G}_v . As it is straightforward to implement σ via the disjoint union of memory structures implementing the constituent strategies, this approach yields an upper bound of $n(2n+4n^2)(d+2)(W+1)$ on the size of σ due to the upper bound on the size of winning strategies for Player 0 in energy parity games from Proposition 5.2.

In the construction of the \mathcal{G}_v , however, we only store the edges chosen by the players in the vertices of the form $E \times P \times \{0, 1\}$ for didactic purposes. In fact, it suffices to store the target vertex of an edge instead, resulting in a vertex set of size $6n$ of the \mathcal{G}_v . Moreover, recall that the definition of the \mathcal{G}_v only takes the color of v into account: If the vertices v and v' have the same color, then the games \mathcal{G}_v and $\mathcal{G}_{v'}$ are isomorphic. Further, Chatterjee



Proof. Figure 6: A game of size $\mathcal{O}(n)$ in which Player 0 only wins with strategies of size at least $nW + 1$.

and Doyen have shown that, if Player 0 wins an energy parity game \mathcal{G}' with n' vertices, d' colors, and largest absolute weight W' , then he has a uniform strategy of size $n'd'W'$ that is winning from all vertices from which he wins \mathcal{G}' [4]. Hence, it suffices to combine at most d strategies, each of size $(6n)(d+2)(W+1)$, in order to obtain a winning strategy for Player 0 in \mathcal{G} . \square

Having established an upper bound on the memory required by Player 0, we now proceed to show that this bound is tight.

Lemma 6.3. *Let $n, W \in \mathbb{N}$. There exists a parity game with weights $\mathcal{G}_{n,W}$ with $\mathcal{O}(n)$ vertices and largest absolute weight W such that Player 0 wins $\mathcal{G}_{n,W}$ from every vertex, but each winning strategy for him is of size at least $nW + 1$.*

We show the game $\mathcal{G}_{n,W}$ in Figure 6. This game has $n + 4$ vertices and the largest absolute weight of an edge is W . Hence, we have $|\mathcal{G}_{n,W}| = (n + 4) \log W \in \mathcal{O}(n \log W)$. The only vertices with more than one successor are $v_{\text{del}} \in V_0$ and $v'_{\text{ans}} \in V_1$. Thus, the only choice of Player 0 in $\mathcal{G}_{n,W}$ consists in determining how often to take the self-loop of vertex v_{del} upon each visit. Dually, the only choice of Player 1 consists of deciding whether or not to move from v'_{ans} to $v'_{\text{req},1}$, or to continue to v_{ans} .

Player 0 wins $\mathcal{G}_{n,W}$ from each vertex by taking the self-loop of v_{del} nW times upon each visit to v_{del} and by subsequently moving to v'_{ans} . Each request in each play that is consistent with this strategy is answered or unanswered with cost at most nW , independent of the choices of Player 1 in v'_{ans} . Moreover, as the only way to visit v_{req} is to move there from v_{ans} , the play visits v_{ans} infinitely often if and only if it visits v_{req} infinitely often. Further, the play visits $v'_{\text{req},1}$ and v'_{ans} infinitely often. Hence, almost all requests are answered, i.e., this strategy is winning for Player 0 from all vertices. This strategy can be implemented by a counter that counts the number of self-loops of v_{del} taken so far, which is reset to zero upon leaving v_{del} . As this counter is bounded by nW , the strategy is of size $nW + 1$.

It remains to show that each finite-state winning strategy for Player 0 has at least $nW + 1$ memory states. Towards a contradiction, let σ be a winning strategy for Player 0 from some vertex v with less than $nW + 1$ memory states. We implement a strategy for Player 1 using a counter κ that is initialized with one if $v = v_{\text{req}}$ and with zero otherwise. Upon each visit to v_{req} we increment κ . After each visit to v_{req} , the strategy τ prescribes moving from v'_{ans} to $v'_{\text{req},1}$ for the first κ visits to v'_{ans} , and it prescribes moving from v'_{ans} to v_{ans} at the $(\kappa + 1)$ -th visit to v'_{ans} . Hence, after the $(\kappa + 1)$ -th visit to v'_{ans} , the vertex v_{req} is visited again, κ is incremented, and the behavior of τ described above repeats with incremented κ .

Let ρ be a play consistent with σ and τ . Since σ is winning for Player 0, the play ρ does not remain in v_{del} from some point onwards ad infinitum, as this would violate the parity condition and thus contradict σ being winning from v for Player 0. Hence, playing consistently with τ , Player 1 enforces a play that starts with a (possibly empty) finite prefix that ends before the first visit to v_{req} and that continues with infinitely many rounds, each starting in v_{req} . The j -th round is of the form

$$v_{\text{req}} \cdot \prod_{k=0, \dots, j} (v'_{\text{req},1} \cdots v'_{\text{req},n} \cdot (v_{\text{del}})^{\ell_{j,k}} \cdot v'_{\text{ans}}) \cdot v_{\text{ans}} \ .$$

We first show $\ell_{j,k} < nW + 1$ for all j, k . Towards a contradiction, assume $\ell_{j,k} \geq nW + 1$ for some $j, k \in \mathbb{N}$. Since σ is of size less than $nW + 1$, a straightforward pumping argument shows that the play ρ' consisting of the finite prefix of ρ ending before the first visit to v_{req} concatenated with the first $j - 1$ rounds of ρ , but ending with the infinite suffix

$$v_{\text{req}} \cdot \prod_{k'=0, \dots, k-1} (v'_{\text{req},1} \cdots v'_{\text{req},n} \cdot (v_{\text{del}})^{\ell_{j,k'}} \cdot v'_{\text{ans}}) \cdot v'_{\text{req},1} \cdots v'_{\text{req},n} \cdot (v_{\text{del}})^\omega \ ,$$

is consistent with σ . This, however, contradicts σ being a winning strategy for Player 0 from v as v_{del} has the odd color one, i.e., as the resulting play violates the parity condition. Hence, we obtain $\ell_{j,k} < nW + 1$ for all j, k , which, in turn, yields $w((v_{\text{del}})^{\ell_{j,k}}) > -nW$.

Since each edge $(v'_{\text{req},n'}, v'_{\text{req},n'+1})$ for $1 \leq n' < n$ as well as the edge $(v'_{\text{req},n}, v'_{\text{del}})$ has weight W , we obtain

$$w(v'_{\text{req},1} \cdots v'_{\text{req},n} \cdot (v_{\text{del}})^{\ell_{j,k}} \cdot v'_{\text{ans}}) > 0$$

for all j, k . This, in turn, implies

$$w(v_{\text{req}} \cdot \prod_{k=0, \dots, j} (v'_{\text{req},1} \cdots v'_{\text{req},n} \cdot (v_{\text{del}})^{\ell_{j,k'}} \cdot v'_{\text{ans}})) \geq j + 1$$

for each j . Since, as argued above, the play ρ consistent with σ consists of infinitely many rounds, we obtain that for each $b \in \mathbb{N}$ there exist infinitely many requests in ρ that are answered with cost at least b . Hence, the costs-of-requests along ρ diverge, which contradicts σ being a winning strategy for Player 0. \square

This concludes the study of memory requirements for both players in parity games with weights. For Player 0, the results from Theorem 6.1 also hold true for bounded parity games with weights: Lemma 6.2 directly yields an upper bound on the memory required by Player 0 in order to win in bounded parity games with weights, while it is easy to see that Player 0 also wins the games constructed in the proof of Lemma 6.3 when interpreting them as bounded parity games with weights, but only using a strategy of size $nW + 1$.

For Player 1, however, these results do not directly carry over to the setting of bounded parity games with weights. She has, in fact, a positional winning strategy for the game witnessing necessity of infinite memory for her in parity games with weights shown in Figure 2 on Page 9, when interpreted as a bounded parity game with weights. Recall that, in the proof of the first item of Lemma 5.6, for a given parity game with weights \mathcal{G} , we construct a strategy τ that is winning for Player 1 from a vertex v out of a winning strategy τ_v for her in an induced energy parity game \mathcal{G}_v from a designated vertex v' .

While τ_v can be assumed to be positional as shown by Chatterjee and Doyen [4] (cf. Proposition 5.1), the strategy τ keeps track of play prefixes in \mathcal{G}_v and thus requires potentially infinitely many memory states. In particular, in order to win \mathcal{G}_v from v' , recall that it may be necessary to switch between two copies of the arena of \mathcal{G} . Whether or not to perform this switch is governed by the accumulated weight of the play prefix in \mathcal{G} thus far.

Hence, our construction from the proof of that lemma does not directly allow us to obtain positional or finite-state winning strategies for Player 1 in bounded parity games

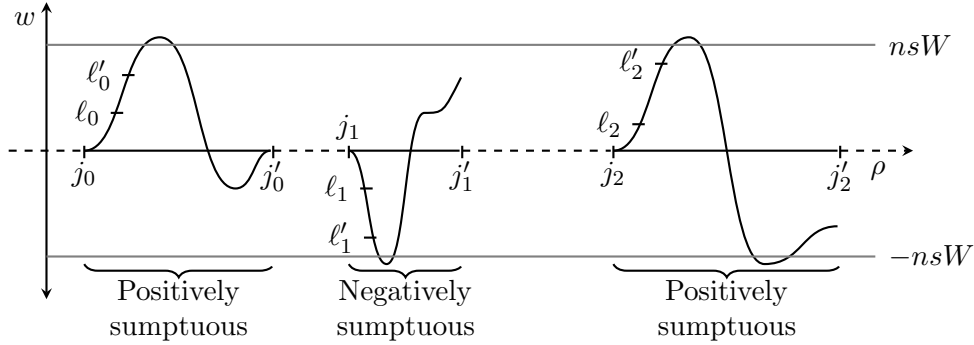


Figure 7: Illustration of the approach to the proof of Lemma 7.2.

with weights. It remains open whether Player 1 requires infinite memory to win in bounded parity games with weights. Since she, however, has finite-state winning strategies in the special case of bounded parity games with costs as shown by Fijalkow and Zimmermann [15], we conjecture that she requires at most finite memory for winning strategies in bounded parity games with weights as well.

We now turn our attention to the quantitative properties of this winning condition. To this end, we provide tight bounds on the costs of requests that Player 0 can guarantee in a parity game with weights \mathcal{G} , if she wins \mathcal{G} at all.

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 7.1. *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 $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) \leq b$. This bound is tight.

We first show that Player 0 can indeed ensure an upper bound as stated in Theorem 7.1. We obtain this bound via a straightforward pumping argument leveraging the upper bound on the size of winning strategies obtained in Lemma 6.2.

Lemma 7.2. *Let \mathcal{G} , n , d , and W be as in the statement of Theorem 7.1 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 $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) \leq nsW$.*

Proof. Let σ be a winning strategy for Player 0 in \mathcal{G} from $\mathcal{W}_0(\mathcal{G})$ of size at most s . Due to Lemma 6.2, such a strategy exists. Let $\rho = v_0 v_1 v_2 \dots$ be a play that starts in $\mathcal{W}_0(\mathcal{G})$ and is consistent with σ . We call a position $j \in \mathbb{N}$ **sumptuous** if $nsW < \text{Cor}(\rho, j) < \infty$. Each

sumptuous position j has some odd color c , and the request for c posed by visiting v_j is eventually answered.

Assume towards a contradiction that there exist infinitely many sumptuous positions. We define a sequence of positions that begins with the first sumptuous position j_0 . Let j'_0 be the minimal position that satisfies $\Omega(v_{j'_0}) \in \text{Ans}(\Omega(v_{j_0}))$. This position exists since $\text{Cor}(\rho, j_0) < \infty$ due to the definition of sumptuous positions. We continue by defining j_1 as the smallest sumptuous position greater than j'_0 and by defining j'_1 as the minimal position that satisfies $\Omega(v_{j'_1}) \in \text{Ans}(\Omega(v_{j_1}))$. Continuing in this manner, we obtain a sequence $j_0 < j'_0 < j_1 < j'_1 < j_2 < j'_2 < \dots$, where j_0 is the first sumptuous position of ρ , each j'_k for $k \geq 0$ is the minimal position that satisfies both $j'_k > j_k$ and $\Omega(v_{j'_k}) \in \text{Ans}(\Omega(v_{j_k}))$, and each j_k for $k > 0$ is the smallest sumptuous position greater than j'_{k-1} . Since there exist infinitely many sumptuous positions by assumption and since each request posed at a sumptuous position is answered by definition, the sequence $j_0 < j'_0 < j_1 < j'_1 < j_2 < j'_2 < \dots$ is indeed infinite.

Due to the definition of sumptuous positions and the j'_k , we have $\text{Ampl}(v_{j_k} \dots v_{j'_k}) > nsW$ for each $k \in \mathbb{N}$. Since ρ is consistent with the finite-state strategy σ of size s , we claim that in each such $v_{j_k} \dots v_{j'_k}$ there exists an infix that can be repeated arbitrarily often while retaining consistency with σ . To identify such infixes, we partition the sumptuous positions j_k : We call a position j_k **positively sumptuous** if there exists a j' with $j_k \leq j' \leq j'_k$ such that $w(v_{j_k} \dots v_{j'}) > nsW$ and **negatively sumptuous** otherwise. In the latter case, there exists a j' with $j_k \leq j' \leq j'_k$ such that $w(v_{j_k} \dots v_{j'}) < -nsW$. See Figure 7 for an illustration. In particular, note that the third sumptuous position is positively sumptuous, although it exceeds both the bounds nsW and $-nsW$.

Let σ be implemented by $(M, \text{init}, \text{upd})$. As each edge contributes cost at most W to $\text{Ampl}(v_{j_k} \dots v_{j'_k})$, this implies that there exist positions ℓ_k and ℓ'_k with $j_k < \ell_k < \ell'_k < j'_k$ such that

- $v_{\ell_k} = v_{\ell'_k}$,
- $\text{upd}^+(v_0 \dots v_{\ell_k}) = \text{upd}^+(v_0 \dots v_{\ell'_k})$,
- $w(v_{\ell_k} \dots v_{\ell'_{k-1}}) > 0$, if j_k is positively sumptuous, and such that
- $w(v_{\ell_k} \dots v_{\ell'_{k-1}}) < 0$, if j_k is negatively sumptuous.

The positions j_k , ℓ_k , ℓ'_k , and j'_k split ρ into infixes $\rho = \prod_{k=0,1,2,\dots} \pi_{k,I} \cdot \pi_{k,II} \cdot \pi_{k,III} \cdot \pi_{k,IV}$, where $\pi_{k,I}$, $\pi_{k,II}$, $\pi_{k,III}$, and $\pi_{k,IV}$ start at j_k , ℓ_k , ℓ'_k , and j'_k , respectively. Due to the definition of ℓ_k and ℓ'_k , the play $\rho' = \prod_{k=0,1,2,\dots} \pi_{k,I} \cdot (\pi_{k,II})^k \cdot \pi_{k,III} \cdot \pi_{k,IV}$ is consistent with σ . The costs-of-response of the requests opened by visiting the v_{j_k} , however, diverge due to $|w(\pi_{k,II})| = |w(v_{\ell_k} \dots v_{\ell'_{k-1}})| > 0$. Hence, ρ' violates the parity condition with weights, which contradicts that σ is a winning strategy of Player 0. \square

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 7.3. *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 for each vertex $v \in \mathcal{W}_0(\mathcal{G})$ and for each winning strategy for Player 0 from v there exists a play ρ starting in v and consistent with σ with $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) \geq (n-1)W$.*

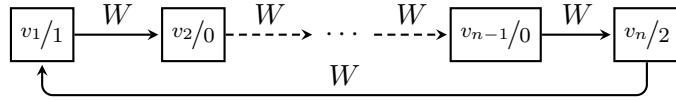


Figure 8: The game $\mathcal{G}_{n,W}$ witnessing an exponential lower bound on the cost that Player 0 can ensure.

Proof. We show the game $\mathcal{G}_{n,W}$ in Figure 8. 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 $\mathcal{G}_{n,W}$. 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. \square

While Player 0 is able to bound the costs from above by $d(6n^2)(d + 1)W(W + 1) \in \mathcal{O}((ndW)^2)$ due to Lemma 7.2, we only obtain a sequence of examples witnessing a lower bound of $(n - 1)W \in \Omega(nW)$ on these costs due to Lemma 7.3. Thus, while both the upper and the lower bound are exponential in the size of the game, there remains a polynomial gap between the two bounds.

The upper bound on the costs is due to a pumping argument leveraging the upper bound on the memory Player 0 requires to implement a winning strategy due to Lemma 6.2. We have a lower bound of $(n - 4)W \in \Omega(nW)$ on the memory required by Player 0 to implement a winning strategy due to Lemma 6.2. Thus, even if we are able to improve the upper bounds on the memory requirements for Player 0, the pumping argument leveraged in the proof of Lemma 7.2 can only improve the upper bound on the cost incurred by Player 0 to $d(n - 4)W^2 \in \mathcal{O}(ndW^2)$. Hence, the proof methods used in this section do not enable us to completely close this gap between the upper and the lower bound on the cost Player 0 is able to enforce.

All bounds obtained in this section also hold for bounded parity games with weights: The upper bound from Lemma 7.2 holds for bounded parity games with weights since the bounded parity condition with weights strengthens its unbounded variant. Analogously, the games constructed for the proof of the lower bound on the incurred cost in Lemma 7.3 yield the same lower bound for bounded parity games with weights.

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. We then 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. In an energy parity game, Player 0 wins if the energy increases without a bound, as long as there is a lower bound. In a bounded parity game, in contrast, he has to ensure both 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 5.3.

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 , which starts in v and is consistent with σ , satisfies the parity condition and $\text{Ampl}(\rho) \leq b$. 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 8.1. *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 he has a corridor-winning strategy for \mathcal{G}' from v .*

Proof. We first show the direction from right to left. To this end, assume that Player 0 wins \mathcal{G} from v . Due to Proposition 5.2, he has a finite-state winning strategy σ for \mathcal{G} from v , say of size s . Furthermore, there is an initial credit c_0 such that every play prefix π that starts in v and is consistent with σ satisfies $w(\pi) \geq -c_0$. Finally, define $b = Wns$, where n and W denote the number of vertices and the largest absolute weight occurring in \mathcal{G} , respectively.

We define a strategy σ' for Player 0 in \mathcal{G}' such that it mimics the behavior of σ and additionally ensures that the energy level of a play prefix never exceeds b by more than W . Formally, consider a play prefix π' in \mathcal{G}' starting in v . If π' ends in some $v' \in V$, then we define $\sigma'(\pi) = (v', \sigma(f(\pi)))$ where $f: (V \cup E)^* \cup (V \cup E)^\omega \rightarrow V^* \cup V^\omega$ is the homomorphism induced by $f(v) = v$ and $f(e) = \varepsilon$. On the other hand, assume π' ends in some $e = (v', v'') \in E$. If $w(\pi') \leq b$, then we define $\sigma'(\pi') = v''$, otherwise, we define $\sigma'(\pi') = e$, i.e., the self-loop is used until the energy level of the play prefix is exactly b . This completes the definition of σ' .

Now, consider a play ρ' in \mathcal{G}' that starts in v and is consistent with σ' . By definition of σ' , ρ' visits infinitely many vertices in V . Hence, by construction of \mathcal{A} , $\rho = f(\rho')$ is a play in \mathcal{G} that starts in v . Further, ρ is consistent with σ , as σ' mimics σ . Hence, $f(\rho')$ satisfies

the parity condition. As the vertices removed from ρ' all have color one, and as all colors in $\Omega(V)$ are greater than one, we conclude that ρ' satisfies the parity condition as well.

To conclude, we show that every prefix π' of ρ' satisfies $-c_0 \leq w'(\pi') \leq b + W$. This implies that σ' is indeed a corridor-winning strategy from v . The upper bound $b + W$ is satisfied by construction of σ' : As soon as the weight exceeds b , it is decreased to b by the strategy. As this correction happens after each transition, the bound b can be exceeded by at most W , the largest absolute weight of an edge.

To conclude, we consider two cases: first, assume ρ' has no prefix whose energy level exceeds b , then we have $w'(\pi') = w(f(\pi)) \geq -c_0$ for every prefix π' of ρ' . Second, assume that ρ' has at least one prefix whose weight exceeds b . Let π be the shortest such prefix. For every prefix π' shorter than π we obtain $-c_0 \leq w'(\pi') \leq b + W$ via the above argument. We show that every prefix longer than π has non-negative weight, which concludes the proof.

Towards a contradiction, assume that there is a longer suffix with negative weight. Then there is an infix of ρ' of weight strictly smaller than $-b$, such that Player 0 never uses a self-loop in \mathcal{A}' to throw away energy. Hence, $f(\rho)$ also has an infix with weight strictly smaller than $-b$. This, however, contradicts Lemma 5.3 and thus concludes the proof of this direction.

We now show the other direction of the lemma. To this end, let σ' be a corridor-winning strategy for Player 0 in \mathcal{G}' from v . Further, let f be defined as above.

We define a strategy σ for Player 0 from v in \mathcal{G} that is obtained by simulating play prefixes in \mathcal{G}' . To this end, we again use a simulation function h that maps a play prefix $v_0 \cdots v_j$ in \mathcal{G} that starts in v and is consistent with σ to a play prefix $h(v_0 \cdots v_j)$ in \mathcal{G}' that starts in v , is consistent with σ' , and ends in v_j .

Hence, we define $h(v) = v$. Now, assume we have a play prefix $v_0 \cdots v_j$ in \mathcal{G} that starts in v and is consistent with σ . From our construction, we obtain a play prefix $h(v_0 \cdots v_j)$ in \mathcal{G}' that starts in v , is consistent with σ' , and ends in v_j . If $v_j \in V_0 \subseteq V'_0$, then let $\sigma'(h(v_0 \cdots v_j)) = (v_j, v_{j+1})$. We define $\sigma(v_0 \cdots v_j) = v_{j+1}$, which is a legal move due to the construction of \mathcal{A}' . If $v_j \in V_1$, then let v_{j+1} be an arbitrary successor of v_j in \mathcal{A} .

In both cases, we have to define $h(v_0 \cdots v_j v_{j+1})$. As σ' is a corridor-winning strategy for Player 0 from v in \mathcal{G}' , there is a unique play of the form $h(v_0 \cdots v_j)(v_j, v_{j+1})^m v_{j+1}$ that is consistent with σ' . We define $h(v_0 \cdots v_j v_{j+1})$ to be equal to this play, which satisfies the properties required above.

Let b be the uniform bound on the amplitude of plays in \mathcal{G}' consistent with σ' starting in v . Now, fix a play ρ in \mathcal{G} starting in v and consistent with σ . Furthermore, let ρ' be the limit of the $h(\pi)$ for increasing prefixes of ρ . By construction, ρ' starts in v as well and is consistent with σ' . Hence, ρ' visits infinitely many vertices from V and never gets stuck in a self-loop throwing away energy. This implies $f(\rho') = \rho$. Furthermore, as ρ' satisfies the parity condition, ρ does as well: the colors removed by applying f are inconsequential in this situation.

Let π_j be the prefix of length j of ρ . A straightforward induction proves that the energy level of π_j is greater or equal to that of $h(\pi_j)$. As the latter is bounded from below by b , we conclude that σ is winning for Player 0 in \mathcal{G} from v with initial credit b . \square

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 \bar{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, \bar{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 $\mathcal{G}'' = (\mathcal{A}'', \text{BndWeightParity}(\Omega'', w''))$.

Lemma 8.2. *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 $\bar{v} \in \mathcal{W}_0(\mathcal{G}'')$.*

Proof. We again first show the direction from right to left. To this end, let σ' be a corridor-winning strategy for Player 0 in \mathcal{G}' from v . Further, let b be the corresponding uniform bound on the amplitude of plays that start in v and are consistent with σ' . We define a strategy σ'' for Player 0 from \bar{v} via $\sigma''(\bar{v}\pi) = \sigma'(\pi)$.

Let $\bar{v}\rho$ be a play that is consistent with σ'' . By construction, ρ starts in v and is consistent with σ' . Hence, it satisfies the parity condition and its amplitude is bounded by b . Thus, almost all requests in ρ are answered with cost at most b and there is no unanswered request of infinite cost. This implies that $\bar{v}\rho$ satisfies the bounded parity condition with weights. Hence, $\bar{v} \in \mathcal{W}_0(\mathcal{G}'')$.

We now show the obverse direction of the lemma. To this end, let σ'' and b be a winning strategy for Player 0 in \mathcal{G}'' from \bar{v} and a bound such that every request in a play starting in \bar{v} and consistent with σ'' is answered or unanswered with cost at most b . Due to Corollary 5.15, such a strategy σ'' exists. We define a strategy σ' for Player 0 from v in \mathcal{G}' via $\sigma'(\pi) = \sigma''(\bar{v}\pi)$.

Let ρ be a play starting in v that is consistent with σ' . By construction, $\bar{v}\rho$ is consistent with σ'' . Hence, $\bar{v}\rho$ satisfies the parity condition and every request is answered or unanswered with cost at most b . In particular, this holds true for the unanswered request posed by visiting \bar{v} . Hence, the amplitude of $\bar{v}\rho$ (and thus also that of ρ) is bounded by b .

Thus, ρ satisfies the parity condition and the energy level of all its prefixes is between $-b$ and b . As ρ is picked arbitrarily, we have that σ' is corridor-winning from v . \square

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 4.1: 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 4.1, however, 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 8.3. *Let \mathcal{G} and \mathcal{G}_{v^*} as above. Then, $v^* \in \mathcal{W}_0(\mathcal{G})$ if and only if $v^* \in \mathcal{W}_0(\mathcal{G}_{v^*})$.*

Proof. First, we show the direction from left to right. To this end, let σ be a winning strategy for Player 0 for \mathcal{G} from v^* such that there is a b , such that every request in a play that starts in v^* and is consistent with σ is answered or unanswered with cost at most b . Due to Corollary 5.15, such a strategy σ exists.

We define a winning strategy σ' for Player 0 from v^* in \mathcal{G}'_{v^*} as follows: Given a play or a play prefix π' in \mathcal{A}_{v^*} that does not end in vertex \top , let $\text{sfx}_{\top}(\pi')$ be the longest suffix of π' that does not contain \top . Hence, if π' starts in v^* , then $\text{sfx}_{\top}(\pi')$ starts in v^* as well, as v^* is the unique successor of \top . Further, let $f: (V \cup E)^* \cup (V \cup E)^\omega \rightarrow V^* \cup V^\omega$ be the homomorphism induced by $f(v) = v$ for $v \in V$ and $f(e) = \varepsilon$ for $e \in E$. Now, if π' is a play (prefix) in \mathcal{A}_{v^*} that does not visit \top , then $f(\pi')$ is a play (prefix) in \mathcal{A} of the same weight that induces the same sequence of colors (save for the occurrences of the inconsequential minimal color zero at the vertices from E that are deleted by f).

Let π' be a play prefix in \mathcal{A}_{v^*} that ends in a vertex $v \in V'_0 = V_0$. We define $\sigma'(\pi') = (v, \sigma(f(\text{sfx}_{\top}(\pi'))))$ and show that σ' is winning for Player 0 in \mathcal{G}'_{v^*} from v^* . To this end, let ρ' be a play in \mathcal{A}_{v^*} starting in v^* that is consistent with σ' . We consider two cases, depending on whether or not the play ρ' visits the vertex \top infinitely often.

If ρ' visits \top only finitely often, then $\text{sfx}_{\top}(\rho')$ is an infinite play in \mathcal{G}'_{v^*} starting in v^* . By definition of f and construction of σ' , the infinite play $f(\text{sfx}_{\top}(\rho'))$ in \mathcal{A} starts in v^* and is consistent with σ . Hence, $f(\text{sfx}_{\top}(\rho'))$ satisfies the bounded parity condition with weights. Since this condition strengthens the parity condition with weights and since the latter condition is 0-extendable, we conclude that $f(\rho')$ satisfies the parity condition with weights as well. This, in turn, implies that the complete play ρ' satisfies the parity condition with weights due to the construction of \mathcal{A}_{v^*} .

Now, assume that the play ρ' visits \top infinitely often. Then, ρ' is of the form $\pi'_0 \top \pi'_1 \top \pi'_2 \top \dots$, where none of the π'_j visits \top . Hence, by definition of \mathcal{A}' , f , and σ' , each play prefix $f(\pi'_j)$ in \mathcal{A} starts in v^* and is consistent with σ . Furthermore, every request in each π'_j is answered by the next visit to the vertex \top at the latest, i.e., ρ' satisfies the parity condition. Thus, it suffices to show that the cost-of-response of all requests in ρ' is bounded. This follows immediately from the fact that σ only admits answered or unanswered requests of cost at most b when starting in v^* and that each $f(\pi'_j)$ starts in v^* and is consistent with σ . This property is inherited by the π'_j due to the construction of \mathcal{A}_{v^*} . Thus, ρ' satisfies the parity condition with weights, i.e., σ' is indeed winning for Player 0 from v^* .

To show the direction from right to left, we proceed by contraposition. Due to the determinacy of both games, it suffices to show that $v^* \in \mathcal{W}_1(\mathcal{G})$ implies $v^* \in \mathcal{W}_1(\mathcal{G}_{v^*})$. Hence, let τ be a winning strategy for Player 1 in \mathcal{G} from v . Further, let sfx_{\top} and f be defined as above.

Now, we define a strategy τ' for Player 1 from v^* in \mathcal{G}_{v^*} that is controlled by a counter κ , which is initialized with zero, and which is incremented during a play every time the costs of some request exceed κ . We construct our strategy such that each time κ is updated, Player 1 restarts the play by moving to \top and then to v^* .

Assume we have a play prefix π' in \mathcal{A}_{v^*} that ends in a vertex of Player 1 and have to define $\tau'(\pi')$. We consider several cases depending on the last vertex of π' . If π' ends in \top , then we define $\tau'(\pi') = v^*$, which is the only successor of \top .

If π' ends in $v \in V_1 \subseteq V'_1$, then we define $\tau'(\pi') = (v, \tau(f(\text{sfx}_{\top}(\pi'))))$, i.e., we discard everything up to and including the last occurrence of \top . Finally, if π' ends in $e = (v, v') \in E \subseteq V'_1$, then we consider two cases. Let κ be the current counter value. If $\text{sfx}_{\top}(\pi')$ contains a request such that the remaining part of π' that starts at this request has amplitude greater than κ , then we define $\tau'(\pi') = \top$ and increment κ . Otherwise, we define $\tau'(\pi') = v'$ and leave κ unchanged.

It remains to show that τ' is winning in \mathcal{G}_{v^*} from v^* . To this end, let ρ' be a play in \mathcal{G}_{v^*} that starts in v^* and is consistent with τ' . If ρ' visits \top infinitely often, then ρ' contains, for every $b \in \mathbb{N}$, a (different) request that is answered or unanswered with cost at least b . Hence, ρ' violates the parity condition with costs.

Finally, if ρ' visits \top only finitely often, then there is a $b \in \mathbb{N}$ (the final value of κ , which is incremented only finitely often in this case) such that every request in ρ' is answered or unanswered with cost at most b . Furthermore, let ρ be the suffix of ρ' that starts after the last occurrence of \top . As in the previous case, $f(\rho)$ is a play in \mathcal{A} that starts in v^* and is consistent with τ . As ρ and $f(\rho)$ have essentially the same evolution of the weights (save for the removed edges of weight zero) and the same color sequence (save for the removed vertices of color zero), every request in $f(\rho)$ is answered or unanswered with cost at most b . However, as ρ is consistent with τ , it violates the bounded parity condition with weights. This is, in this situation, only possible by violating the parity condition. Hence ρ , and thus also ρ' , violates the parity condition as well. Therefore, ρ' in particular violates the parity condition with weights.

In both cases, ρ' is winning for Player 1, i.e., τ' has the desired properties. \square

8.3. Relation to Mean-Payoff Parity Games. Recall that we have shown in Section 4 and Section 5 how to solve parity games with weights by solving polynomially many energy parity games of polynomial size. Subsequently, in Section 8.1 and Section 8.2 we have shown the converse direction, i.e., how to solve energy parity games by solving polynomially many parity games with weights, again of polynomial size. In summary, we have shown the problem of solving parity games with weights and that of solving energy parity games with weights to be polynomial time equivalent.

Due to this strong connection between the two games, we moreover obtain a connection to another widely used class of games, so-called mean-payoff parity games as introduced by Chatterjee, Henzinger, and Jurdziński [8]. A mean-payoff parity game is played on a colored arena with weights. It is the task of Player 0 to not only satisfy the parity condition induced by the coloring, but also to ensure that the average weight of the traversed edges is non-negative. Formally, given an arena with vertex set V and set of edges E , a coloring Ω of V , and a weight function w over E , the mean-payoff parity condition is defined as

$$\text{MeanPayoffParity}(\Omega, w) = \{v_0 v_1 v_2 \cdots \in V^\omega \mid \liminf_{j \rightarrow \infty} \frac{1}{j} w(v_0 \cdots v_j) \geq 0\} \cap \text{Parity}(\Omega) .$$

A game $(\mathcal{A}, \text{MeanPayoffParity}(\Omega, w))$ is called a mean-payoff parity game.

Chatterjee and Doyen [4] showed that the problem of solving energy parity games and that of solving mean-payoff parity games are logarithmic space equivalent.

Proposition 8.4 ([4]). *Let $\mathcal{G} = (\mathcal{A}, \text{MeanPayoffParity}(\Omega, w))$ be a mean-payoff parity game with n vertices and let $\mathcal{G}' = (\mathcal{A}, \text{EnergyParity}(\Omega, w'))$, where $w'(e) = w(e) + \frac{1}{1+n}$ for all edges e of \mathcal{A} . Then $\mathcal{W}_0(\mathcal{G}) = \mathcal{W}_0(\mathcal{G}')$.*

Hence, we can use the techniques underlying Theorem 5.7 to reduce the problem of solving parity games with weights to the problem of solving mean-payoff parity games. In fact, it is Proposition 8.4 that underpins the proof of the second part of Proposition 5.4, i.e., the proof that energy parity games can be solved in pseudo-quasi-polynomial time.

Corollary 8.5. *The following decision problems are polynomial time equivalent:*

Given a parity game with weights \mathcal{G} and a vertex v of \mathcal{G} , does Player 0 have a winning strategy from v in \mathcal{G} ?

Given a mean-payoff parity game \mathcal{G} and a vertex v of \mathcal{G} , does Player 0 have a winning strategy from v in \mathcal{G} ?

Solving parity games with weights by iteratively solving mean-payoff parity games as sketched above, however, does not allow us to obtain the memory bounds from Theorem 6.1. This is due to Player 0, in general, requiring infinite memory in order to win a mean-payoff parity game [8].

9. THE THRESHOLD PROBLEM

In Section 3, we have defined parity games with weights as a generalization of parity games with costs. Up to this point, we have only considered the problem of solving parity games with weights, i.e., deciding for a given parity game with weights \mathcal{G} and a vertex v of \mathcal{G} , whether Player 0 has a strategy σ such that all plays ρ starting in v and consistent with σ satisfy $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) < \infty$. We have shown this problem to be a member of $\text{NP} \cap \text{co-NP}$ and that it can be solved in pseudo-quasi-polynomial time.

While our algorithm for deciding the above decision problem yields a strategy witnessing the ability of Player 0 to eventually ensure a finite cost-of-response, it does not provide any guarantee on the “quality” of the strategy beyond the very general bound $b \in \mathcal{O}((ndW)^2)$ due to Theorem 7.1. In particular, it may be the case that there exists a strategy σ' such that all plays ρ starting in v and consistent with σ satisfy $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) \leq b' < b$. Hence, in this section, we investigate the so-called threshold problem for parity game with weights, i.e., the problem to decide, given a parity game with weights \mathcal{G} , a vertex v of \mathcal{G} , and a bound $b \in \mathbb{N}$, whether there exists a strategy σ for Player 0 such that for all plays ρ starting in v and consistent with σ satisfy $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) \leq b$.

Weinert and Zimmermann [32] have shown that the threshold problem for finitary parity games as well as for parity games with costs is PSPACE-complete and that exponential memory is both necessary and sufficient for both players to implement witnessing strategies. In this work, we show that the threshold problem for parity games with weights is EXPTIME-complete and that the memory bounds remain unchanged in comparison to the case of parity games with costs.

In order to simplify notation for the remainder of this section, if \mathcal{G} is clear from the context, for a given strategy σ for Player 0 and a vertex v of \mathcal{G} we define

$$\text{Cost}_v(\sigma) = \sup_{\rho} \limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) \text{ ,}$$

where ρ ranges over all plays ρ of \mathcal{G} that start in v and that are consistent with σ . Dually, for a strategy τ for Player 1 and a vertex v , we define $\text{Cost}_v(\tau) = \inf_{\rho} \limsup_{j \rightarrow \infty} \text{Cor}(\rho, j)$, where ρ ranges over all plays ρ of \mathcal{G} that start in v and that are consistent with τ .

Remark 9.1. *Let σ be a strategy for Player 0, let τ be a strategy for Player 1, and let v be a vertex. Then, $\text{Cost}_v(\sigma) \geq \text{Cost}_v(\tau)$.*

The main theorem of this section settles the complexity of the threshold problem for parity games with weights.

Theorem 9.2. *The following decision problem is EXPTIME-complete:*

Given a parity game with weights \mathcal{G} , some vertex v_I of \mathcal{G} , and a bound $b \in \mathbb{N}$, does Player 0 have a strategy σ with $\text{Cost}_{v_I}(\sigma) \leq b$ in \mathcal{G} ?

The remainder of this section is organized as follows: First, in Section 9.1 we show the threshold problem for parity games with weights to be in EXPTIME, before showing EXPTIME-hardness of the problem in Section 9.2. We conclude this section by showing tight bounds on the memory requirements of witnessing strategies for both players in Section 9.3.

9.1. ExpTime-Membership. In order to solve the threshold problem for parity games with weights, we follow the approach used by Weinert and Zimmermann [32] to solve the same problem for parity games with costs. Given a parity game with costs \mathcal{G} and a threshold b , they construct a classical parity game \mathcal{G}_b that satisfies the following property:

Player 0 has a strategy of cost at most b from some vertex v in \mathcal{G} if and only if he has a winning strategy from some designated vertex v' in \mathcal{G}_b .

We show how to lift the construction of Weinert and Zimmermann to the setting of parity games with weights and show that this construction yields EXPTIME-membership of the threshold problem for parity games with weights. For the remainder of this section, we fix some parity game with weights \mathcal{G} with n vertices, d odd colors and a threshold $b \in \mathbb{N}$.

The idea behind the construction of \mathcal{G}_b is to track, for each odd color c of \mathcal{G} , an overapproximation of the costs incurred by all open requests for c in the current play prefix. The authors then showed that, if Player 1 is able to violate the bound b at least n times, then she can do so infinitely often. Thus, by additionally equipping \mathcal{G}_b with an n -bounded counter that counts the number of violations of the given threshold, we obtain the desired property given above.

Recall that, in parity games with costs, all weights are nonnegative. Hence, it suffices to track an upper bound on the cost of open requests, as these costs are implicitly bounded from below by zero. In the setting of parity games with weights, in contrast, we have to track both an upper as well as a lower bound on the cost of open requests. To this end, we first define the set of intervals

$$I = \{(l, h) \mid -b \leq l \leq h \leq b\} .$$

Clearly, we have

$$|I| = (2b + 1) + \dots + 1 = (2b + 1)(2b + 2)/2 = (4b^2 + 4b + 2b + 2)/2 = 2b^2 + 3b + 1 .$$

Using the set of intervals, we now define request functions that enable the overapproximation of the costs of open requests described above. To this end, we denote the set of odd colors occurring in \mathcal{G} by D . A (b -bounded) request function $r: D \rightarrow \{\perp\} \cup I$ is a function mapping each odd color of \mathcal{G} either to

- \perp , denoting that currently no request for color c is open, or to
- some $(l, h) \in I$, denoting that
 - there exists an open request for color c that has accumulated weight l , that
 - there exists an open request for color c that has accumulated weight h , and that
 - all requests for color c have accumulated weight at least l and at most h .

Given some request function r , we define the lower and upper residual request functions r_\downarrow and r_\uparrow as $r_\downarrow(c) = l$ and $r_\uparrow(c) = h$, if $r(c) = (l, h)$, and as $r_\downarrow(c) = r_\uparrow(c) = \perp$ if $r(c) = \perp$. We write R to denote the set of all request functions. We have $|R| = (|I|+1)^{|D|} = (2b^2+3b+2)^d$, i.e., there exist exponentially many request functions when measured in the size of the game \mathcal{G} , but only polynomially many when measured in the bound b .

As stated above, Weinert and Zimmermann [32] showed that it suffices for Player 1 to violate the threshold b n times in order to witness that she can do so infinitely often. Hence, we now define a memory structure comprising request functions and an “overflow counter” that, together with the game \mathcal{G} , induces the desired parity game \mathcal{G}_b .

Recall that we fixed some parity game with weights \mathcal{G} with n vertices and d odd colors as well as a bound $b \in \mathbb{N}$. Using the set R of request functions defined above, we define the set of memory states $M = \{0, \dots, n\} \times R$. As we aim to track the cost of open requests using the functions from R , we define the initial memory element $\text{init}(v) = (0, r_v)$, where r_v is defined as

$$r_v(c) = \begin{cases} (0, 0) & \text{if } \Omega(v) \text{ is odd and } c = \Omega(v), \text{ and} \\ \perp & \text{otherwise.} \end{cases}$$

We define the update function $\text{upd}: M \times E \rightarrow M$ implementing the above intuition. Let $m = (o, r) \in M$ and let $e = (v, v') \in E$ with $w(e) = w$. This update function updates the memory state via $\text{upd}(m, e) = (o', r')$ by performing the following steps in order:

Weight: First, we resolve the effect of traversing the edge e with weight w by defining r'_I as

$$r'_I(c) = \begin{cases} (r_\downarrow(c) + w(e), r_\uparrow(c) + w(e)) & \text{if } r(c) \neq \perp, \text{ and} \\ \perp & \text{otherwise.} \end{cases}$$

Overflow: In a second step, we check whether some request has violated the bound b during the move to v' and update the overflow counter if this is the case. Thus, if there exists a color c such that either $(r'_I)_\downarrow(c) < -b$ or $(r'_I)_\uparrow(c) > b$, then we define $r'_{II}(c) = \perp$ for all $c \in D$ and set o' to the minimum of $o + 1$ and n . Otherwise, we define $r'_{II} = r'_I$ and $o' = o$.

Request: Finally, we resolve the effect of moving to the vertex v' with color $\Omega(v')$ as follows: If $\Omega(v')$ is even, then we define

$$r'_{III}(c) = \begin{cases} \perp & \text{if } c \leq \Omega(v'), \text{ and} \\ r'_{II}(c) & \text{otherwise.} \end{cases}$$

If, however, $\Omega(v')$ is odd, then we define

$$r'_{III}(c) = \begin{cases} (\min\{(r'_{II})_\downarrow(\Omega(v')), 0\}, \max\{(r'_{II})_\uparrow(\Omega(v')), 0\}) & \text{if } c = \Omega(v'), \text{ and} \\ r'_{II}(c) & \text{otherwise.} \end{cases}$$

In either case, we define $r' = r'_{III}$, which concludes the definition of $m' = (o', r')$. The resulting o' is at most n and the resulting function r' is an element of R . We combine these elements in the memory structure $\mathcal{M} = (M, \text{init}, \text{upd})$.

Recall that throughout this section we fixed a parity game with weights \mathcal{G} . Let $\mathcal{G} = (\mathcal{A}, \text{WeightParity}(\Omega, w))$. We define the (b) -threshold game of \mathcal{G}

$$\mathcal{G}_b = (\mathcal{A}', \text{Parity}(\Omega')) ,$$

with $\mathcal{A}' = (V', V'_0, V'_1, E')$, where $V' = V \times M$, $V'_i = V_i \times M$ for $i \in \{0, 1\}$, where $((v, m), (v', m')) \in E'$ if and only if $(v, v') \in E$ and $\text{upd}(m, (v, v')) = m'$, and where

$$\Omega'(v, o, r) = \begin{cases} \Omega(v) & \text{if } o < n, \text{ and} \\ 1 & \text{otherwise,} \end{cases}$$

which concludes the definition of \mathcal{G}_b .

Via a straightforward adaptation of results by Weinert and Zimmermann [32] we obtain that this construction indeed satisfies the above property, i.e., that it suffices to solve \mathcal{G}_b in order to solve the threshold problem for \mathcal{G} :

Lemma 9.3. *Let v^* be a vertex of \mathcal{G} . Player 0 has a strategy σ with $\text{Cost}_{v^*}(\sigma) \leq b$ if and only if he wins \mathcal{G}_b from $(v^*, \text{init}(v^*))$.*

We split the proof of Lemma 9.3 into several lemmas. The direction from right to left is relatively straightforward: Since \mathcal{G}_b is a parity game, if Player 0 wins \mathcal{G}_b from $(v^*, \text{init}(v^*))$, then she does so with a positional strategy σ' . This strategy assigns to each vertex (v, o, r) of Player 0 in \mathcal{G}_b a unique successor (v', o', r') , where the values of o' and r' are deterministic updates of o and r via the update function upd . Hence, σ' can be interpreted as picking only a successor vertex v' of v with respect to the current memory state (o, r) . Thus, the choices of σ' can be mimicked in \mathcal{G} .

Lemma 9.4. *If Player 0 wins \mathcal{G}_b from $(v^*, \text{init}(v^*))$, then she has a strategy σ in \mathcal{G} with $\text{Cost}_{v^*}(\sigma) \leq b$.*

Proof. Let $\sigma' : V'_0 \rightarrow V'$ be a positional winning strategy for Player 0 from $(v^*, \text{init}(v^*))$ in \mathcal{G}_b . Since \mathcal{G}_b is a parity game, such a strategy exists by assumption. We define the finite-state strategy σ for Player 0 in \mathcal{G} as the unique strategy induced by the memory structure \mathcal{M} and the next-move function Nxt defined as $\text{Nxt}(v, m) = v'$, if $\sigma'(v, m) = (v', m')$ for some m' . It remains to show $\text{Cost}_{v^*}(\sigma) \leq b$.

Let $\rho = v_0 v_1 v_2 \dots$ be a play in \mathcal{G} that begins in v^* and that is consistent with σ . Let

$$\rho' = (v_0, o_0, r_0)(v_1, o_1, r_1)(v_2, o_2, r_2) \dots$$

be the unique play in \mathcal{G}_b that satisfies both $(o_0, r_0) = \text{init}(v_0)$ as well as $(o_j, r_j) = \text{upd}((o_{j-1}, r_{j-1}), (v_{j-1}, v_j))$ for all $j > 0$.

A straightforward induction shows that ρ' is consistent with σ' . As ρ' moreover starts in $(v^*, \text{init}(v^*))$ by definition, it is winning for Player 0, i.e., it satisfies the parity condition. This in particular implies that the overflow counter along ρ' never saturates, i.e., that we have $o_j < n$ for all $j \in \mathbb{N}$, since vertices with saturated overflow counter form a losing sink for Player 0. Hence, the plays ρ and ρ' coincide on their color sequences. Since ρ' is winning for Player 0 in \mathcal{G}' , it satisfies the parity condition, which in turn implies that ρ satisfies the parity condition. It remains to show that almost all requests in ρ are answered with cost at most b .

As argued above, the overflow counter along ρ' eventually stabilizes at some value less than n . Moreover, since ρ satisfies the parity condition, all but finitely many requests are answered. Hence, there exists a position j such that $o_{j'} = o_j < n$ and such that $\text{Cor}(\rho, j') <$

∞ for every $j' > j$. Assume towards a contradiction that there exists some $j' \geq j$ with $b < \text{Cor}(\rho, j') < \infty$. Then, due to the construction of \mathcal{G}_b , the overflow counter is incremented along the play ρ' at some point after j' . This, however, contradicts the choice of j . Hence, we obtain that almost all requests in ρ are answered with cost at most b . \square

We now turn our attention to the other direction of the statement of Lemma 9.3, i.e., we aim to show that, if Player 0 has a strategy σ in \mathcal{G} with $\text{Cost}_{v^*}(\sigma) \leq b$, then she wins \mathcal{G}_b from $(v^*, \text{init}(v^*))$. We show this claim via contraposition: Assume that Player 0 does not win \mathcal{G}_b from $(v^*, \text{init}(v^*))$. Since \mathcal{G}_b is a parity game, it is determined. Hence, Player 1 wins \mathcal{G}_b from $(v^*, \text{init}(v^*))$, say with the positional strategy τ' . Such a positional strategy for him exists again due to \mathcal{G}_b being a parity game. We construct a strategy τ for Player 1 in \mathcal{G} that enforces $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) > b$ for each play ρ starting in v^* and consistent with τ . Since this implies $\text{Cost}_{v^*}(\sigma) > b$ for each strategy σ of Player 0 (see Remark 9.1), this suffices to show the desired statement.

Recall that the overflow counter along each play starting in $(v^*, \text{init}(v^*))$ is monotonically increasing and bounded from above by the number n of vertices in \mathcal{G} . Hence, the value of the overflow counter either stabilizes at some value less than n , or it eventually saturates at value n . In the former case, τ' has to ensure that the resulting play violates the parity condition. Hence, it suffices to mimic the moves made by τ' in \mathcal{G} ad infinitum in this case, which results in a play with infinitely many unanswered requests in \mathcal{G} . In the latter case, however, mimicking τ' does not yield a strategy in \mathcal{G} with the desired property, as τ' does not necessarily prescribe “meaningful” moves in \mathcal{G}_b once the overflow counter saturates. In order to leverage τ' even after saturation of the overflow counter, we intervene whenever the overflow counter is incremented, by resetting it to the smallest possible value from which τ' is still winning.

Formally, we define the set \mathcal{R} that contains all vertices (v, o, r) that are visited by some play that starts in $(v^*, \text{init}(v^*))$ and that is consistent with τ' . Recall that we defined r_v as the function denoting the requests opened by visiting the vertex v . Given a vertex v , we then define $o_v = \min(\{n\} \cup \{o \mid (v, o, r_v) \in \mathcal{R}\})$. In particular, we have $o_{v^*} = 0$, since $(v^*, \text{init}(v^*)) = (v^*, 0, r_v) \in \mathcal{R}$.

Lemma 9.5. *The strategy τ' is winning for Player 1 from (v, o_v, r_v) in \mathcal{G}_b for all $v \in V$.*

Proof. If $o_v = n$, then all plays starting in (v, o_v, r_v) violate the parity condition by construction of the arena of \mathcal{G}_b . Thus, for the remainder of this proof, assume $o_v < n$. Let $(v, o_v, r_v)\rho$ be a play starting in (v, o_v, r_v) that is consistent with τ' . Moreover, let π be a play prefix starting in $(v^*, \text{init}(v^*))$, consistent with τ' , and ending in (v, o_v, r_v) . Such a play prefix exists due to definition of o_v and due to the assumption of $o_v < n$, which implies $(v, o_v, r_v) \in \mathcal{R}$.

Since τ' is positional, the play $\pi\rho$ starts in $(v^*, \text{init}(v^*))$ and is consistent with τ' . Thus, $\pi\rho$ violates the parity condition, which implies that ρ violates the parity condition due to prefix-independence. \square

We now define a new memory structure \mathcal{M}' implementing the strategy τ . Recall that we defined the memory structure $\mathcal{M} = (M, \text{init}, \text{upd})$ during the construction of the threshold game \mathcal{G}_b . Using the components of that memory structure, we define $M' = M = [n + 1] \times R$, $\text{init}' = \text{init}$, and

$$\text{upd}'((o, r), (v, v')) = \begin{cases} (o, r') & \text{if } \text{upd}((o, r), (v, v')) = (o, r'), \text{ and} \\ (o_v', r') & \text{if } \text{upd}((o, r), (v, v')) = (o + 1, r'). \end{cases}$$

and combine these elements into the memory structure $\mathcal{M}' = (M', \text{init}', \text{upd}')$.

In the second case of the definition of upd' , we have $r' = r_{v'}$ by definition of upd . Finally, we define the next-move function Nxt' via $\text{Nxt}'(v, m) = v'$, if $\tau'(v, m) = (v', m')$ for some $m' \in M$ and let τ be the finite-state strategy implemented by \mathcal{M}' and Nxt' . We claim that for each play ρ starting in v^* that is consistent with τ we have $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) > b$.

To show this claim, let $\rho = v_0 v_1 v_2 \dots$ be some play in \mathcal{G} that starts in v^* and that is consistent with τ . Moreover, let

$$\rho' = (v_0, o_0, r_0)(v_1, o_1, r_1)(v_2, o_2, r_2) \dots$$

be the unique play in $\mathcal{A} \times \mathcal{M}'$ that satisfies $(o_0, r_0) = \text{init}'(v_0)$ as well as $(o_j, r_j) = \text{upd}'((o_{j-1}, r_{j-1}), (v_{j-1}, v_j))$ for all $j > 0$. We say that j is a reset position if $j = 0$ or if

$$\text{upd}'((o_{j-1}, r_{j-1}), (v_{j-1}, v_j)) = (o_{j-1} + 1, r_j) ,$$

i.e., if the second case in the definition of upd' is applied.

The play ρ' is not necessarily a play in \mathcal{G}_b , since \mathcal{G}_b is defined with respect to \mathcal{M} instead of \mathcal{M}' , but every infix of ρ' that starts at a reset position and does not contain another one, is a play infix in \mathcal{G}_b that is consistent with τ' . At every reset position, instead of incrementing the overflow counter, we set it to o_v .

Via a straightforward induction leveraging the same arguments as Weinert and Zimmermann [32] we obtain $o_j < n$ for all $j \in \mathbb{N}$, i.e., the overflow counter along ρ' never saturates. Intuitively, this implies that the strategy τ always uses “meaningful” moves of τ' for its choice of move and thus allows us to subsequently argue that τ is indeed winning for Player 1. Moreover, while the set M' of memory states as defined above is of size $(n+1)|R|$, the above observation allows us to implement the strategy τ' using a set of memory states of size $n|R|$ by omitting the memory states indicating a saturated overflow counter.

It remains to show that we indeed have $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) > b$. As argued above, this then directly implies $\text{Cost}_{v^*}(\sigma) > b$ for each strategy σ of Player 0, concluding the proof of the direction from left to right of Lemma 9.3.

Proof of Lemma 9.3. The direction from right to left is encapsulated in Lemma 9.4.

For the direction from left to right, recall that we defined a strategy τ for Player 1, that we picked ρ beginning in v^* and consistent with τ arbitrarily and that we defined $\rho' = (v_0, o_0, r_0)(v_1, o_1, r_1)(v_2, o_2, r_2) \dots$. First assume that the overflow counter of ρ' eventually stabilizes, i.e., there exists some $j \in \mathbb{N}$ such that $o_{j'} = o_j$ for all $j' > j$. Then, there exists a suffix of ρ' that is consistent with τ' , which therefore violates the parity condition. Hence, it suffices to note that the color sequences induced by ρ' and by ρ coincide in this case since the overflow counter along ρ' never saturates as argued above, and due to the construction of \mathcal{G}_b , in which vertices of the form (v, o, r) inherit the coloring of the vertex v for $o < n$. Thus, ρ violates the parity condition and, in turn, also the parity condition with weights with respect to any bound.

Now assume that the overflow counter of ρ' does not stabilize. Then, there are infinitely many reset positions in ρ' . We obtain that there exist a request that incurs cost greater than b between any two adjacent such positions as a direct consequence of the construction of the arena of \mathcal{G}_b . Hence, we obtain $\limsup_{j \rightarrow \infty} \text{Cor}(\rho, j) > b$, which concludes this direction of the proof. \square

Lemma 9.3 allows us to decide the threshold problem for a parity game with weights with n vertices and d odd colors by solving a classical parity game of size $\mathcal{O}(nb^d)$, i.e., the

size of the parity game depends on both the size of the parity game with weights as well as the threshold. Lemma 7.2, however, allows us to bound the parameter b : If b is at least $nd(6n)(d+1)(W+1)W$, then solving the threshold problem reduces to solving the parity game with weights. We formalize these observations as Algorithm 3.

Algorithm 3 An algorithm deciding whether or not Player 0 has a strategy σ with $\text{Cost}_v(\sigma) \leq b$ in a parity game with weights \mathcal{G} .

Input: Parity game with weights \mathcal{G} with n vertices, d odd colors, and largest absolute weight W , vertex v of \mathcal{G} , bound $b \in \mathbb{N}$.

- 1: **if** $b \geq nd(6n)(d+1)(W+1)W$ **then**
- 2: **return** $v \in \mathcal{W}_0(\mathcal{G})$ /* Requires solving \mathcal{G} , e.g. via Algorithm 1 */
- 3: **else**
- 4: $\mathcal{G}_b = b$ -threshold game of \mathcal{G} /* \mathcal{G}_b is explicitly constructed */
- 5: **return** $(v, 0, r_v) \in \mathcal{W}_0(\mathcal{G}_b)$ /* Requires solving \mathcal{G}_b */
- 6: **end if**

Recall that our aim in this section is to argue that the threshold problem for parity games with weights is in EXPTIME. To this end, we have constructed Algorithm 3, the correctness of which follows directly from Lemma 9.3. It remains to argue that this algorithm runs in exponential time.

To this end, recall that parity games can be solved in polynomial time in the number of vertices and in exponential time in the number of odd colors [3, 16]. Moreover, recall that the (b) -threshold game \mathcal{G}_b has $\mathcal{O}(nb^d)$ many vertices, where n and d are the number of vertices and the number of odd colors in \mathcal{G} , respectively. As we are able to bound the value of b from above by a polynomial in n , d , and the largest absolute weight W of \mathcal{G} due to Lemma 7.2, we obtain that we are able to solve \mathcal{G}_b in exponential time, which we formalize in the following theorem.

Theorem 9.6. *The following problem is in EXPTIME:*

Given a parity game with weights \mathcal{G} , a vertex v of \mathcal{G} , and a bound $b \in \mathbb{N}$, does Player 0 have a strategy σ with $\text{Cost}_v(\sigma) \leq b$?

Proof. We show that Algorithm 3 witnesses the claimed membership in EXPTIME. The correctness of this algorithm follows directly from Lemma 7.2 and Lemma 9.3. It remains to show that Algorithm 3 terminates in exponential time. Let n be the number of vertices of \mathcal{G} , let d be the number of odd colors of \mathcal{G} , and let W be the largest absolute weight in \mathcal{G} .

If $b \geq nd(6n)(d+1)(W+1)W$, then the dominating factor for the runtime of Algorithm 3 is the call to a solver for parity games with weights in Line 2. This solver only has to solve the given parity game with weights as discussed in Section 4. Due to Theorem 5.7, the problem of solving these games is in $\text{NP} \cap \text{CO-NP}$. Since $\text{NP} \subseteq \text{EXPTIME}$, we obtain the desired runtime in this case.

If, however, $b < nd(6n)(d+2)(W+1)W$, Algorithm 3 constructs and solves the b -threshold game \mathcal{G}_b of \mathcal{G} in Line 4 and Line 5, respectively. Let n' be the number of vertices of \mathcal{G}_b . By construction of \mathcal{G}_b we obtain

$$n' = n|\{0, \dots, n\} \times R| = n(n+1)(2b^2 + 3b + 2)^d \in \mathcal{O}(n^2b^{2d}) ,$$

where R denotes the set of request functions as defined above.

Due to our case analysis based on the cardinality of b in Line 1, we furthermore obtain $b \in \mathcal{O}(n^2 d^2 W)$, which in turn implies

$$n' \in \mathcal{O}(n^2 (n^2 d^2 W)^{2d}) = \mathcal{O}(n^{2+4d} d^{4d} W^{2d}) .$$

As we assume weights to be given in binary encoding, we additionally obtain $W \in \mathcal{O}(2^{|\mathcal{G}|})$, which finally implies

$$n' \in \mathcal{O}(n^{2+4d} d^{4d} (2^{|\mathcal{G}|})^{2d}) = n^{2+4d} d^{4d} 2^{2d|\mathcal{G}|} ,$$

i.e., \mathcal{G}_b contains only exponentially many vertices and $d' = \max\{1, d\} \in \mathcal{O}(d)$ many colors in terms of $|\mathcal{G}|$. Recall that parity games can be solved in polynomial time in the number of vertices and in exponential time in the logarithm of the number of colors using, e.g., the recent algorithm by Calude et al. [3]. Hence, \mathcal{G}_b can indeed be solved in exponential time in $|\mathcal{G}|$, which implies membership of the above problem in EXPTIME. \square

Algorithm 3 furthermore yields an algorithm determining the optimal b such that Player 0 has a strategy of cost at most b from a given v : Given a parity game with weights \mathcal{G} with n vertices, d odd colors, and largest absolute weight W , and a vertex v of \mathcal{G} , we first solve \mathcal{G} and determine whether or not $v \in \mathcal{W}_0(\mathcal{G})$. If this is not the case, then no such bound b exists. Otherwise, the minimal b with the above property can be determined with a binary search over the range $0, \dots, nd(6n)(d+1)(W+1)W$. This binary search requires deciding at most $\log(nd(6n)(d+1)(W+1)W)$, i.e., polynomially many instances of the threshold problem, each of which takes at most exponential time. Hence, the optimal b such that Player 0 has a strategy of cost at most b can be determined in exponential time in the size of \mathcal{G} .

This concludes the proof of EXPTIME-membership of the threshold problem for parity games with weights. It remains to show that this bound is tight. To this end, we show in the next section that the threshold problem is EXPTIME-hard via a reduction from the problem of solving countdown games.

9.2. ExpTime-Hardness. In the previous section we have shown EXPTIME-membership of the threshold problem for parity games with weights.

In this section we provide a matching lower bound by showing that the threshold problem is EXPTIME-complete. To this end, we reduce the EXPTIME-hard problem of solving countdown games to the threshold problem for parity games with weights. Countdown games were introduced by Jurdziński, Laroussinie, and Sprston [18].

In a countdown game, some initial credit is fixed at the beginning of a play. Both players then move in alternation in an arena whose edges are labeled with negative weights. In each turn, first Player 0 announces some weight, before Player 1 has to move along some edge of that weight, reducing the initial credit by the weight of the traversed edge. If the credit at some point hits zero, Player 0 wins. If, however, the credit at some point turns negative, Player 1 wins. Since each edge has strictly negative weight, each play is won by either player after finitely many moves.

When formulated in our framework of arenas and winning conditions, a countdown game $\mathcal{G} = (\mathcal{A}, \text{Countdown}(w, c_I))$ consists of an arena $\mathcal{A} = (V, V_0, V_1, E)$, a weighting w of \mathcal{A} , and some initial credit $c_I \in \mathbb{N}$, which satisfies the following conditions:

- (1) There exists a designated sink vertex $v_\perp \in V_1$,
- (2) we have

- $E \subseteq (V_0 \times V_1) \cup (V_1 \times V_0) \cup (\{v_\perp\} \times \{v_\perp\})$,
 - $V_0 \times \{v_\perp\} \subseteq E$, and
 - $(v_\perp, v_\perp) \in E$,
- (3) for each $e_1 = (v, v'_1), e_2 = (v, v'_2) \in E$, with $v \in V_0$ we have that $e_1 \neq e_2$ implies $w(e_1) \neq w(e_2)$,
- (4) for each $e \in E \cap (V_0 \times V)$ we have $w(e) \begin{cases} = 0 & \text{if } e \in (V_0 \times \{v_\perp\}), \text{ and} \\ < 0 & \text{otherwise, and} \end{cases}$
- (5) for each $e \in E \cap (V_1 \times V)$ we have $w(e) = 0$.

As discussed above, a countdown game is played in turns. Each turn starts at a vertex v of Player 0, from where Player 0 first picks some outgoing edge e leading to vertex v' of Player 1. That edge has unique weight among the outgoing edges of v due to the third requirement. Moreover, if Player 0 does not opt to end the play by moving to v_\perp , the weight of the edge is negative due to the fourth condition. Subsequently, Player 1 picks a successor of v' and moves to that successor along an edge of weight zero due to the fifth requirement, where the next turn of the play starts.

The countdown condition is defined as

$$\text{Countdown}(w, c_I) = \{\rho = v_0 v_1 v_2 \cdots \in V^\omega \mid \exists j. v_j = v_\perp \text{ and } c_I + w(\rho) = 0\} .$$

Our definition of countdown games differs from the one given by Jurdziński, Laroussinie, and Sproston [18], as we adapted it to fit our framework of games introduced in Section 2. It is, however, easy to see that our definition and the one given by the authors are equivalent.

As countdown games are essentially of finite duration, we obtain that they are determined due to Zermelo [33].

Remark 9.7. Countdown games are determined.

Jurdziński, Laroussinie, and Sproston showed that solving countdown games is EXP-TIME-hard via a reduction from the word problem for alternating Turing machines with polynomially bounded space. In order to concisely encode the exponential number of configurations attainable by the Turing machine during its run on the input word and the transitions between these configurations, this reduction requires the weights along the edges of the countdown game as well as the initial credit c_I to be given in binary encoding.

Proposition 9.8 ([18]). *The following problem is EXP-TIME-hard:*

Let \mathcal{G} be a countdown game and let v be a vertex of \mathcal{G} . Does Player 0 have a winning strategy from v in \mathcal{G} ?

We reduce the problem of solving countdown games to the threshold problem for parity games with weights. To this end, for the remainder of this section, fix some countdown game $\mathcal{G} = (\mathcal{A}, \text{Countdown}(w, c_I))$ where $\mathcal{A} = (V, V_0, V_1, E)$, as well as some initial vertex $v_I \in V$.

Intuitively, we construct the parity game with weights \mathcal{G}' such that at the beginning of the play a single request is opened, which is only answered upon visiting v_\perp . After visiting v_\perp , the play returns to the initial vertex v_I reopening the unique request of \mathcal{G}' along the way.

In a countdown game, only Player 0 may decide to move to v_\perp . Since he should, intuitively, only do so after traversing a play prefix of weight $-c_I$, we equip the edges leading from his vertices to v_\perp with weight $2c_I$. Thus, he can enforce “tallying the score” by moving to v_\perp .

In order to afford Player 1 the same possibility, we add edges that allow her to move from his vertices to v_\perp . Furthermore, in order to incentivize her to only take these edges once she has exceeded the lower bound of $-c_I$, these edges have weight zero. All remaining weights remain unchanged, thus the costs incurred by the unique request in \mathcal{G}' model the remaining credit in the corresponding play in \mathcal{G} .

Formally, let v_\top be some vertex not occurring in V . We define the parity game with weights $\mathcal{G}' = (\mathcal{A}', \text{WeightParity}(\Omega, w'))$, where $\mathcal{A}' = (V', V'_0, V'_1, E')$, with

- $V' = V \cup \{v_\top\}$, $V'_0 = V_0 \cup \{v_\top\}$, $V'_1 = V_1$, and
- $E' = (E \setminus \{(v_\perp, v_\perp)\}) \cup (V_1 \times \{v_\perp\}) \cup \{(v_\perp, v_\top), (v_\top, v_I)\}$.

Since there exists a unique outgoing edge of v_\perp leading back to the initial vertex v_I of the countdown game via v_\top , the play is restarted after visiting the sink vertex.

Furthermore, we define the weight function

$$w'(e) = \begin{cases} w(e) & \text{if } (v, v') \in E \setminus (V_0 \times \{v_\perp\}) \\ 2c_I & \text{if } e \in V_0 \times \{v_\perp\} \\ 0 & \text{otherwise} \end{cases}$$

as well as the coloring

$$\Omega'(v) = \begin{cases} 1 & \text{if } v = v_\top, \\ 2 & \text{if } v = v_\perp, \\ 0 & \text{otherwise} \end{cases}$$

and claim that Player 0 has a strategy σ with $\text{Cost}_{v_\top}(\sigma) \leq c_I$ if and only if he wins \mathcal{G} from v_I . We illustrate this construction in Figure 9.

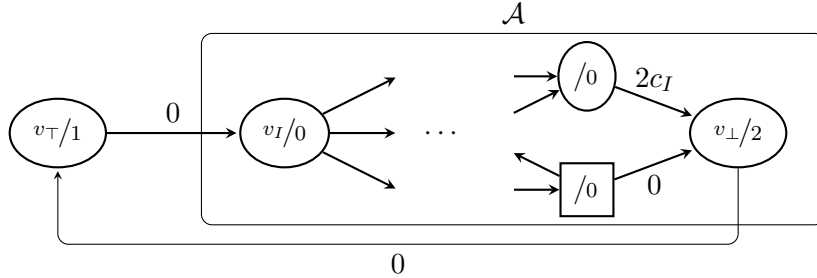


Figure 9: Construction of the parity game with weights \mathcal{G}' from a given countdown game \mathcal{G} . We omit unchanged weights for the sake of readability.

We claim that this construction implements our intuition, i.e., that solving \mathcal{G}' with respect to the bound c_I is indeed equivalent to solving \mathcal{G} .

Lemma 9.9. *Player 0 wins \mathcal{G} from v_I if and only if he has a strategy σ with $\text{Cost}_{v_\top}(\sigma) \leq c_I$ in \mathcal{G}' .*

Proof. We first show the direction from left to right. To this end, let σ be a winning strategy for Player 0 in \mathcal{G} from v_I . Moreover, let $\pi = v_0 \cdots v_j$ be a play prefix in \mathcal{G}' starting in v_\top and let j' be the largest position in π with $v_{j'} = v_\top$. We define the strategy σ' for Player 0 in \mathcal{G}' via $\sigma'(\pi) = \sigma(v_{j'+1} \cdots v_j)$ and claim $\text{Cost}_{v_\top}(\sigma') \leq c_I$.

To prove this claim, let $\rho' = v_0 v_1 v_2 \cdots$ be a play in \mathcal{G}' starting in v_\top that is consistent with σ' . First assume towards a contradiction that ρ' only visits v_\top finitely often. Then,

due to the structure of the arena, ρ' only visits v_\perp finitely often. By construction of σ' , this implies that ρ' contains a suffix that begins in v_I , is consistent with σ , but never visits v_\perp . This contradicts σ being a winning strategy for Player 0 in \mathcal{G} from v_I . Hence, ρ' visits v_\perp infinitely often.

Thus, ρ' is of the form

$$\rho' = v_\top \pi_0 v_\perp \cdot v_\top \pi_1 v_\perp \cdot v_\top \pi_2 v_\perp \cdots ,$$

where each π_j starts in v_I and is consistent with σ . We first argue that, if π_j ends in a vertex of Player 1, then we have $0 \geq w'(\pi_j) \geq -c_I$: All weights in \mathcal{G}' except for those along the edges from $V_0 \times \{v_\perp\}$ are nonpositive. Hence, $0 \geq w'(\pi_j)$ follows directly from the construction of \mathcal{G}' . Moreover, $w'(\pi_j) < -c_I$ would contradict π_j being consistent with the winning strategy σ for Player 0 in \mathcal{G} , since Player 0 would be unable to continue the play prefix π_j such that the resulting play is winning for her. Hence, we have $0 \geq w'(\pi_j) \geq -c_I$. Moreover, since all edges leading from v_\top and all edges leading to v_\perp have weight zero, and since $w'(\pi)$ is decreasing for increasing prefixes π of π_j due to construction of \mathcal{G}' , we obtain $\text{Ampl}(v_\top \pi_j v_\perp) \leq c_I$.

If, however, π_j ends in a vertex of Player 0, then we have $w'(\pi_j) = -c_I$ —and therefore $\text{Ampl}(v_\top \pi_j v_\perp) = c_I$ —as π_j is consistent with the winning strategy σ for Player 0 in \mathcal{G} . In either case, we obtain that the unique request in $v_\top \pi_j v_\perp$ posed by visiting v_\top is answered with cost at most c_I . Hence, ρ' has cost at most c_I , which concludes this direction of the proof.

We show the other direction of the statement via contraposition: Assume Player 0 does not win \mathcal{G} from v_I . Since \mathcal{G} is determined due to Remark 9.7, Player 1 wins \mathcal{G} from v_I , say with strategy τ . We define a strategy τ' for Player 1 in \mathcal{G}' that is winning for her from v_\top via mimicking moves made by τ until the initial credit is used up. At that point, we define τ' to prescribe moving to v_\perp in order to witness exceeding the initial credit and to restart the play.

Formally, let $\pi' = v_0 \cdots v_j$ be a play prefix in \mathcal{G}' that starts in v_\top and ends in some vertex of Player 1. Moreover, let $j' \leq j$ be the largest position such that $v_{j'} = v_\top$. If $w(v_{j'} \cdots v_j) \geq -c_I$, we define $\tau'(\pi') = \tau(v_{j'+1} \cdots v_j)$. Otherwise, we define $\tau'(\pi') = v_\perp$. It remains to show $\text{Cost}_{v_\top}(\tau') > c_I$, which concludes the proof due to Remark 9.1.

To this end, let ρ' be a play in \mathcal{G}' starting in v_\top consistent with τ' . Due to the structure of \mathcal{A}' , every infix of ρ' that visits neither v_\top nor v_\perp traverses edges of weight zero and negative weight in alternation. Moreover, since τ' prescribes moving to v_\perp once the play infix since the last visit to v_\top has incurred weight exceeding $-c_I$, the play ρ' is of the form

$$\rho' = v_\top \pi_0 v_\perp \cdot v_\top \pi_1 v_\perp \cdot v_\top \pi_2 v_\perp \cdots ,$$

where each π_j starts in v_I and is consistent with τ .

We aim to show $\text{Ampl}(v_\top \pi_j v_\perp) > c_I$ for all j , which implies $\limsup_{k \rightarrow \infty} (\rho', k) > c_I$ due to the construction of \mathcal{G}' and thus suffices to show the desired statement. To this end, fix some $j \in \mathbb{N}$ and first consider the case that π_j ends in a vertex of Player 1. Then we obtain $w(\pi_j) < -c_I$ by definition of τ' . This directly implies the desired statement.

Now consider the case that π_j ends in a vertex of Player 0. We first argue that $w'(\pi_j) \neq -c_I$ holds true. Towards a contradiction, assume $w'(\pi_j) = -c_I$ and recall that π_j starts in v_I and is consistent with the winning strategy τ for Player 1 from v_I . Thus the play $\pi_j(v_\perp)^\omega$ in \mathcal{G} is consistent with τ . Hence, $w'(\pi_j) = -c_I$ contradicts τ being a winning strategy for Player 1 from v_I .

It remains to show $\text{Ampl}(v_{\top}\pi_j v_{\perp}) > c_I$ for the case that π_j ends in a vertex of Player 0. If $w'(\pi_j) < -c_I$, we directly obtain $\text{Ampl}(v_{\top}\pi_j v_{\perp}) > c_I$. If, however, $w'(\pi_j) > -c_I$, we have $w'(\pi_j v_{\perp}) > c_I$, since we defined $w'(v, v_{\perp}) = 2c_I$ for each vertex v of Player 0. Thus, we obtain $\text{Ampl}(v_{\top}\pi_j v_{\perp}) > c_I$ for each infix $v_{\top}\pi_j v_{\perp}$ of ρ' . Hence, each request posed by visiting v_{\top} is answered with cost greater than c_I . Since we argued above that ρ' contains infinitely many visits to v_{\top} , we obtain $\limsup_{k \rightarrow \infty} (\rho', k) > c_I$. \square

Due to Lemma 9.9 we obtain a polynomial reduction from the problem of solving countdown games to the threshold problem for parity games with weights. As the former problem is EXPTIME-hard due to Proposition 9.8, this implies EXPTIME-hardness of the latter problem.

Lemma 9.10. *The following decision problem is EXPTIME-hard:*

Given a parity game with weights \mathcal{G} , some vertex v_I of \mathcal{G} , and a bound $b \in \mathbb{N}$, does Player 0 have a strategy σ with $\text{Cost}_{v_I}(\sigma) \leq b$ in \mathcal{G} ?

Proof. We reduce the problem of solving countdown games to the given problem. To this end, let $\mathcal{G} = (\mathcal{A}, \text{Countdown}(w, c_I))$ be a countdown game and let v_I be a vertex of \mathcal{A} . We construct the parity game with weights \mathcal{G}' as described above. Due to Lemma 9.9, Player 0 wins \mathcal{G} from v_I if and only if he has a strategy σ with $\text{Cost}_{v_{\top}}(\sigma) \leq c_I$. As the problem of solving countdown games is known to be EXPTIME-hard due to Proposition 9.8, this implies the desired result. \square

In the following section, we consider the memory requirements of both players when playing optimally.

9.3. Memory Requirements. Recall that, if Player 0 just aims to win a parity game with weights with n vertices, d odd colors, and largest absolute weight W , then a memory structure of size polynomial in n , d , and W suffices to implement a winning strategy due to Theorem 6.1. Dually, Player 1 requires, in general, infinite memory in order to implement a strategy winning for her, again due to Theorem 6.1.

In this section, we show that these bounds change significantly in the context of the threshold problem: In order to implement a strategy that enforces cost of at most b in a parity game with weights with d odd colors, memory of size polynomial in b and exponential in d is both necessary and sufficient for Player 0. Dually, if Player 1 just aims to enforce a cost of the resulting play larger than some threshold b , strategies of size exponential in d suffice for her to do so. Both of these bounds are tight.

We first argue that exponential memory indeed suffices for both players to satisfy or violate a given threshold in a parity game with weights, respectively. To this end, recall that, given a parity game with weights \mathcal{G} and a threshold b , we determined the solution of the threshold problem by solving the b -threshold game \mathcal{G}' of \mathcal{G} . This threshold game is a classical parity game whose arena consists of the arena of \mathcal{G} augmented with request functions and an overflow counter that is bounded from above by n .

Furthermore recall that, in the proof of Lemma 9.3, we showed how to leverage a winning strategy for either player in \mathcal{G}_b in order to construct a winning strategy for them in \mathcal{G} . To this end, we implemented the strategy for Player 0 in \mathcal{G} using the set $\{0, \dots, n\} \times R$ as memory states, where the set $\{0, \dots, n\}$ implements the overflow counter and where R denotes the set of request functions.

The first component of that memory structure is, however, irrelevant for Player 0: If he has a winning strategy whose behavior is dependent not only on the current vertex and the request function, but also on the value of the overflow counter, then he also has one that only depends on the current vertex and the request function.

Lemma 9.11. *Let \mathcal{G} be a parity game with weights containing d odd colors and let $b \in \mathbb{N}$. Moreover, let v^* be a vertex of \mathcal{G} . If Player 0 has a strategy σ in \mathcal{G} with $\text{Cost}_{v^*}(\sigma) = b$, then he also has a strategy σ' with $\text{Cost}_{v^*}(\sigma') \leq b$ and $|\sigma'| = (2b^2 + b + 2)^d$.*

Proof. Recall that we argued previously that if Player 0 has a strategy σ in \mathcal{G} with $\text{Cost}_{v^*}(\sigma) \leq b$, then he also has a winning strategy from $(v^*, \text{init}(v^*))$ in the threshold game \mathcal{G}_b as defined in Section 9.1. Moreover, since \mathcal{G}_b is a parity game, we obtain that if Player 0 wins \mathcal{G}_b from $(v^*, \text{init}(v^*))$, then he also has a positional winning strategy doing so. Thus, let σ_b be a positional winning strategy for Player 0 from $(v^*, \text{init}(v^*))$ in \mathcal{G}_b .

Let \mathcal{R} be the set of vertices reached by plays starting in $(v^*, \text{init}(v^*))$ and consistent with σ_b . Since σ_b is positional, and since the parity condition is prefix-independent, σ_b is winning from all vertices in \mathcal{R} . Furthermore, for each vertex v and each request function r , we define

$$o_{v,r} = \max(\{0\} \cup \{o \mid (v, o, r) \in \mathcal{R}\}) ,$$

i.e., $o_{v,r}$ is the maximal value such that Player 0 wins \mathcal{G}_b from $(v, o_{v,r}, r)$ using σ_b , or zero, if no such value exists.

We now define the strategy σ' for Player 0 in \mathcal{G} such that it has the above properties. To this end, recall that we defined the memory structure $\mathcal{M} = (M, \text{init}, \text{upd})$ for the construction of \mathcal{G}_b , where $M = \{0, \dots, n\} \times R$, and where R is the set of request functions. We define $M' = R$, the update function $\text{upd}'(r, (v, v')) = r'$, if $\text{upd}((o_{v,r}, r), (v, v')) = (o', r')$, as well as the initialization function $\text{init}'(v) = (o_{v,r_v}, r_v)$, if $\text{init}(v) = (0, r_v)$. Finally, we define the next-move function $\text{Nxt}'(v, r) = v'$, where v' is the unique vertex that satisfies $\sigma_b(v, o_{v,r}, r) = (v', o', r')$, and claim that the strategy σ' implemented by $\mathcal{M}' = (M', \text{init}', \text{upd}')$ and Nxt' has $\text{Cost}_{v^*}(\sigma') \leq b$, which suffices to show the desired statement.

To prove this claim, let $\rho = v_0 v_1 v_2 \dots$ be a play starting in v^* and consistent with σ' and let $(v_0, r_0)(v_1, r_1)(v_2, r_2) \dots$ be the unique play defined via $r_0 = \text{init}'(v_0)$ and $r_j = \text{upd}'(r_{j-1}, (v_j, v_{j+1}))$ for all $j > 0$.

A straightforward induction yields $(v_j, o_{v_j, r_j}, r_j) \in \mathcal{R}$ for all $j \in \mathbb{N}$. We first argue that we have $o_{v_j, r_j} \leq o_{v_{j+1}, r_{j+1}}$ for all $j \in \mathbb{N}$. To this end, let $(o, r) = \text{upd}((o_{v_j, r_j}, r_j), (v_j, v_{j+1}))$. By construction of the arena of the threshold game we have $o \geq o_{v_j, r_j}$ and $r = r_{j+1}$. Moreover, since $(v_j, o_{v_j, r_j}, r_j) \in \mathcal{R}$ and due to our definition of σ' , we obtain $(v_{j+1}, o, r) = (v_{j+1}, o, r_{j+1}) \in \mathcal{R}$. Hence, $o \leq o_{v_{j+1}, r_{j+1}}$, which implies $o_{v_{j+1}, r_{j+1}} \geq o_{v_j, r_j}$.

Thus, the o_{v_j, r_j} are monotonically increasing. Furthermore, we easily obtain $o_{v_j, r_j} < n$ due to all (v_j, o_{v_j, r_j}, r_j) being in \mathcal{R} , the definition of \mathcal{R} , and due to σ_b being winning for Player 0 from $(v^*, \text{init}(v^*))$. Hence, the sequence of the o_{v_j, r_j} eventually stabilizes, i.e., there exists a $j \in \mathbb{N}$ such that $o_{v_{j'}, r_{j'}} = o_{v_j, r_j}$ for all $j' \geq j$.

We argue that the play $(v_j, o_{v_j, r_j}, r_j)(v_{j+1}, o_{v_{j+1}, r_{j+1}}, r_{j+1})(v_{j+2}, o_{v_{j+2}, r_{j+2}}, r_{j+2}) \dots$ is consistent with σ_b : Let $j' \geq j$ be such that $(v_{j'}, o_{v_{j'}, r_{j'}}, r_{j'}) \in V'_0$ and let $\sigma_b(v_{j'}, o_{v_{j'}, r_{j'}}, r_{j'}) = (v_{j'+1}, o, r_{j'+1})$. We then clearly obtain $o \leq o_{v_{j'+1}, r_{j'+1}}$ by definition of the latter. Furthermore, we have $o_{v_{j'}, r_{j'}} \leq o$ due to the construction of the arena of the threshold game, which yields $o = o_{v_{j'+1}, r_{j'+1}}$ due to our assumption $o_{v_{j'}, r_{j'}} = o_{v_{j'+1}, r_{j'+1}}$.

Thus, the play $(v_j, o_{v_j, r_j}, r_j)(v_{j+1}, o_{v_{j+1}, r_{j+1}}, r_{j+1})(v_{j+2}, o_{v_{j+2}, r_{j+2}}, r_{j+2}) \cdots$ starts in a vertex from \mathcal{R} , is consistent with σ_b , and shares a color sequence with a suffix of ρ due to $o_j \leq o_{v_j, r_j} < n$. The strategy σ_b being winning for Player 0 from $(v^*, \text{init}(v^*))$, the construction of \mathcal{G}_b and prefix-independence of the parity condition with weights then yield $w(\rho) \leq b$. \square

For Player 1, in contrast, it is open whether one can omit the overflow counter when implementing a strategy with cost at least b . Hence, we have to include it in the resulting memory structure. Recall, however, that we have argued in Section 9.1 that we are able to omit those memory states modeling a saturated overflow counter, thus slightly reducing the size of the resulting strategy in comparison to a naive implementation. The following upper bound thus results directly from the results of Section 9.1.

Corollary 9.12. *Let \mathcal{G} be a parity game with weights with n vertices and d odd colors and let $b \in \mathbb{N}$. Moreover, let v^* be a vertex of \mathcal{G} . If Player 1 has a strategy τ in \mathcal{G} with $\text{Cost}_{v^*}(\tau) = b$, then she also has a strategy τ' with $\text{Cost}_{v^*}(\tau') \geq b$ and $|\tau'| = n(2b^2 + 3b + 2)^d$.*

Having argued that exponential memory suffices for both players to implement optimal strategies, we now turn our attention to providing matching lower bounds. These exponential lower bounds are inherited from the special case of finitary parity games, for which Weinert and Zimmermann [32] showed that both players require exponential memory in order to implement strategies that ensure or violate a given threshold. We reprint these results here for the sake of completeness.

Proposition 9.13 ([32]).

- (1) *For every $d \geq 1$ there exists a finitary parity game \mathcal{G}_d with a vertex v^* such that*
 - \mathcal{G}_d has d odd colors and $|\mathcal{G}_d| \in \mathcal{O}(d^2)$,
 - Player 0 has a strategy σ in \mathcal{G}_d with $\text{Cost}_{v^*}(\sigma) = d^2 + 2d$,
 - there exists no strategy σ' for Player 0 with $\text{Cost}_{v^*}(\sigma') < d^2 + 2d$, and
 - for every strategy σ for Player 0 in \mathcal{G}_d , $\text{Cost}_{v^*}(\sigma) = d^2 + 2d$ implies $|\sigma| \geq 2^{d-1}$.
- (2) *For every $d \geq 1$ there exists a finitary parity game \mathcal{G}_d with a vertex v^* such that*
 - \mathcal{G}_d has $\mathcal{O}(d)$ many vertices and $2d$ odd colors,
 - Player 1 has a strategy τ in \mathcal{G}_d with $\text{Cost}_{v^*}(\tau) = 5(d-1) + 7$,
 - there exists no strategy τ' for Player 1 with $\text{Cost}_{v^*}(\tau') > 5(d-1) + 7$, and
 - every strategy τ for Player 0 in \mathcal{G}_d with $\text{Cost}_{v^*}(\tau) = 5(d-1) + 7$ has size at least 2^d .

10. 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 $\text{NP} \cap \text{co-NP}$, they are not yet considered a class in their own right. It is also interesting because their 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 2.

As future work, we are looking into the natural extensions of parity games with weights to Streett games with weights [7, 15], and at the complexity of determining optimal bounds

	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 [15, 25]	quasi-poly.	pos./inf.	$\mathcal{O}(nW)$
Parity Games with Weights	pseudo-quasi-poly.	$\mathcal{O}(nd^2W)$ /inf.	$\mathcal{O}((ndW)^2)$

Table 2: Characteristic properties of variants of parity games.

and strategies that obtain them [32]. 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, using the definition $\text{Ampl}(\pi) = \sup_{j \leq k < |\pi|} |w(v_j \cdots v_k)| \in \mathbb{N}_\infty$.

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: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. Khossainov, 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 Simona Ronchi Della Rocca, editor, *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 ω -regular games. In Holger Hermanns and Jens Palsberg, editors, *TACAS 2006*, volume 3920 of *LNCS*, pages 257–271. Springer, 2006. doi:10.1007/11691372_17.
- [7] Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. Finitary winning in ω -regular games. *Trans. Comput. Log.*, 11(1):1–1:27, 2009. doi:10.1145/1614431.1614432.
- [8] Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdziński. 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 mean-payoff parity games. In Dawar and Grädel [11], pages 325–334. doi:10.1145/3209108.3209162.
- [11] Anuj Dawar and Erich Grädel, editors. *LICS 2018*. ACM, 2018. doi:10.1145/3209108.
- [12] E. Allen Emerson and Charanjit S. Jutla. Tree automata, μ -calculus and determinacy. In *FOCS 1991*, pages 368–377. IEEE Computer Society, 1991. doi:10.1109/SFCS.1991.185392.
- [13] 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.
- [14] 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.
- [15] Nathanaël Fijalkow and Martin Zimmermann. Parity and Streett games with costs. *LMCS*, 10(2), 2014. doi:10.2168/LMCS-10(2:14)2014.
- [16] Marcin Jurdziński. Deciding the winner in parity games is in $UP \cap co-UP$. *Inf. Proc. Lett.*, 68(3):119–124, November 1998. doi:10.1016/S0020-0190(98)00150-1.
- [17] Marcin Jurdziński. Small progress measures for solving parity games. In *STACS 2000*, volume 1770 of *LNCS*, pages 290–301. Springer, 2000. doi:10.1007/3-540-46541-3_24.
- [18] Marcin Jurdziński, François Laroussinie, and Jeremy Sproston. Model checking probabilistic timed automata with one or two clocks. *LMCS*, 4(3), 2008. doi:10.2168/LMCS-4(3:12)2008.
- [19] 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.
- [20] 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.
- [21] Dexter Kozen. Results on the propositional μ -calculus. *Theo. Comp. Sci.*, 27:333–354, 1983. doi:10.1016/0304-3975(82)90125-6.
- [22] Karoliina Lehtinen. A modal μ perspective on solving parity games in quasi-polynomial time. In Dawar and Grädel [11], pages 639–648. doi:10.1145/3209108.3209115.
- [23] Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975. URL: <http://www.jstor.org/stable/1971035>.
- [24] 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.
- [25] Fabio Mogavero, Aniello Murano, and Loredana Sorrentino. On promptness in parity games. *Fundam. Inform.*, 139(3):277–305, 2015. doi:10.3233/FI-2015-1235.
- [26] 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.
- [27] Anuj Puri. *Theory of hybrid systems and discrete event systems*. PhD thesis, Computer Science Department, University of California, Berkeley, 1995.
- [28] Sven Schewe. An optimal strategy improvement algorithm for solving parity and payoff games. In *CSL 2008*, volume 5213 of *LNCS*, pages 368–383. Springer, 2008. doi:10.1007/978-3-540-87531-4_27.
- [29] 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.
- [30] Sven Schewe, Ashutosh Trivedi, and Thomas Varghese. Symmetric strategy improvement. In *ICALP 2015*, volume 9135 of *LNCS*, pages 388–400. Springer, 2015. doi:10.1007/978-3-662-47666-6_31.
- [31] Jens Vöge and Marcin Jurdzinski. A discrete strategy improvement algorithm for solving parity games. In E. Allen Emerson and A. Prasad Sistla, editors, *CAV 2000*, volume 1855 of *LNCS*, pages 202–215. Springer, 2000. doi:10.1007/10722167_18.
- [32] 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.
- [33] Ernst Zermelo. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In *Proc. Fifth Congress of Mathematicians, Vol. 2*, pages 501–504. Cambridge Press, 1913.
- [34] Wiesław Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theo. Comp. Sci.*, 200(1-2):135–183, 1998. doi:10.1016/S0304-3975(98)00009-7.
- [35] 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.