Cost-Parity and Cost-Streett Games[*]

Nathanaël Fijalkow[1,2] and Martin Zimmermann[2]

[1] LIAFA, Université Paris 7. nath@liafa.univ-paris-diderot.fr
[2] Institute of Informatics, University of Warsaw. zimmermann@mimuw.edu.pl

Abstract. We consider two-player games played on graphs equipped with costs on edges and introduce two winning conditions, cost-parity and cost-Streett, which require bounds on the cost between requests and their responses. Both conditions generalize the corresponding classical ω-regular conditions as well as the corresponding finitary conditions.
For cost-parity games we show that the first player has positional winning strategies and that determining the winner lies in **NP** ∩ **coNP**. For cost-Streett games we show that the first player has finite-state winning strategies and that determining the winner is **EXPTIME**-complete. This unifies the complexity results for the classical and finitary variants of these games. Both types of cost-games can be solved by solving linearly many instances of their classical variants.

1 Introduction

In recent years, boundedness problems arose in topics pertaining to automata and logics leading to the development of novel models and techniques to tackle these problems. Although in general undecidable, many boundedness problems for automata turn out to be decidable if the acceptance condition can refer to boundedness properties of variables, but the transitions cannot access variable values. A great achievement was made by Hashiguchi [12] who proved decidability of the star-height problem by reducing it to a boundedness problem for automata. This led the path to recent developments towards a general theory of bounds in automata and logics, comprising automata and logics with bounds [1, 3] and satisfiability algorithms for these logics [2, 4], and regular cost-functions [9, 17].

In this work, we consider boundedness problems in turn-based two-player graph games of infinite duration. Using the acceptance condition of the automata models of [3] (namely ωB- and ωS automata) yields games that are equivalent to ω-regular games. Hence, we take a different route and introduce cost-parity and cost-Streett conditions which generalize the (classical) ω-regular parity- respectively Streett condition, as well as the finitary parity- respectively finitary Streett condition [8]. While both finitary variants strengthen the classical conditions by adding bounds, the complexity of solving these games diverges: (in the state of the art) finitary parity games are simpler than parity games, while finitary Streett games are harder than Streett games. Indeed solving finitary parity games can

---

be carried out in polynomial time, while no polynomial-time algorithm for parity games is yet known, and the decision problem is in $\mathbf{NP} \cap \mathbf{coNP}$. The situation is reversed for Streett games, since solving Streett games is $\mathbf{coNP}$-complete while solving finitary Streett games is $\mathbf{EXPTIME}$-complete.

Cost-parity and cost-Streett conditions can be expressed using the acceptance conditions investigated in [17][3]. The languages accepted by automata with these conditions strictly subsume the ω-regular languages, just like the winning conditions studied here do. Indeed, cost-parity and cost-Streett games involve behaviors not exhibited by ω-regular games, e.g., Player 1 needs infinite memory to win such a game.

A cost-parity game is played on an arena whose vertices are colored by natural numbers, and where traversing an edge incurs a non-negative cost. Player 0 wins a play if there is a bound b such that almost all odd colors (which we think of as requests) are followed by a larger even color (which we think of as responses) that is reached with cost at most b. The cost of traversing an edge can be used to model the consumption of a resource. Thus, if Player 0 wins a play she can achieve her goal along an infinite run with bounded resources. On the other hand, Player 1's objective is to exhaust the resource, no matter how big the capacity is. The definition of (cost-) Streett games goes along the same lines, but the requests and responses are not hierarchically ordered, but independent.

We show that cost-parity games enjoy two nice properties of parity and finitary parity games: Player 0 has memoryless winning strategies, and determining the winner lies in $\mathbf{NP} \cap \mathbf{coNP}$. Furthermore, we show that solving cost-parity games can be algorithmically reduced to solving parity games, which allows to solve these games almost as efficiently as parity games. We then consider cost-Streett games and prove that Player 0 has finite-state winning strategies, and that determining the winner is $\mathbf{EXPTIME}$-complete. Our complexity results unify the previous results about finitary parity and Streett games [8] and the results about their classical variants.

To obtain our results, we present an algorithm to solve cost-parity games that iteratively computes the winning region of Player 0 employing an algorithm to solve parity games. As a by-product, we obtain finite-state winning strategies for Player 0. We improve this by showing how to transform a finite-state winning strategy into a positional winning strategy. This construction relies on so-called scoring functions (which are reminiscent of progress measures [13] and signatures [18] for parity games) and presents a general framework to turn finite-state strategies into positional ones, which we believe to be applicable in other situations as well. For cost-Streett games, we present a similar algorithm, this time employing an algorithm to solve Streett games.

Adding quantitative requirements to qualitative winning conditions has been an active field of research during the last decade. Just recently, there has been a lot of interest in so-called energy-parity games [7], whose winning conditions are conjunctions of a parity condition and a boundedness requirement on the resource consumption (the resource is consumed and refilled along edges). The

---

[3] For automata, the conditions in [17] are equivalent to ωB- respectively ωS conditions

winner in such a game can be determined in **NP** ∩ **coNP** and Player 1 has positional winning strategies. Going beyond a single resource often leads to undecidability [10] (see also [5]). Although the first two results are similar to ours on cost-parity games, the energy parity condition does not relate the energy consumption to the parity condition. In contrast, the costs in cost-parity games give a qualitative measure of the satisfaction of the parity condition.

Finally, our results on half-positional determinacy for cost-parity games and finite-state determinacy for cost-Streett games shows that these two conditions can be expressed in parametric linear temporal logic [19], which extends linear temporal logic by parameterized operators. However, solving these games is complete for doubly-exponential time, and thus worse than our results here.

2 Definitions

Infinite Games. A (cost) arena $\mathcal{A} = (V, V_0, V_1, E, \mathrm{cst})$ consists of a finite, directed graph (V, E), a partition $\{V_0, V_1\}$ of V, and an edge-labeling $\mathrm{cst} \colon E \to \{\varepsilon, i\}$. An edge with label i is called increment-edge, edges labeled by ε are called accordingly ε-edges. We extend the edge-labeling to a cost function over finite and infinite paths obtained by counting the number of increment-edges traversed along the path. A play in $\mathcal{A}$ starting in $v \in V$ is an infinite path $\rho = \rho_0 \rho_1 \rho_2 \cdots$ such that $\rho_0 = v$. To avoid the nuisance of dealing with finite plays, we assume every vertex to have an outgoing edge.

A game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ consists of an arena $\mathcal{A}$ and a set $\mathrm{Win} \subseteq V^\omega$ of winning plays for Player 0. The set of winning plays for Player 1 is $V^\omega \setminus \mathrm{Win}$. A strategy for Player i is a mapping $\sigma \colon V^* V_i \to V$ such that $(v, \sigma(wv)) \in E$ for all $wv \in V^* V_i$. We say that σ is positional if $\sigma(wv) = \sigma(v)$ for every $wv \in V^* V_i$. A play $\rho_0 \rho_1 \rho_2 \ldots$ is consistent with σ if $\rho_{n+1} = \sigma(\rho_0 \cdots \rho_n)$ for every n with $\rho_n \in V_i$. A strategy σ for Player i is a winning strategy from a set of vertices $W \subseteq V$ if every play that starts in some $v \in W$ and is consistent with σ is won by Player i. The winning region $W_i(\mathcal{G})$ of Player i in $\mathcal{G}$ is the set of vertices from which Player i has a winning strategy. We say that a strategy is uniform, if it is winning from all $v \in W_i(\mathcal{G})$. We always have $W_0(\mathcal{G}) \cap W_1(\mathcal{G}) = \emptyset$. On the other hand, if $W_0(\mathcal{G}) \cup W_1(\mathcal{G}) = V$, then we say that $\mathcal{G}$ is determined. All games we consider in this work are determined. Solving a game amounts to determining its winning regions.

A memory structure $\mathcal{M} = (M, \mathrm{init}, \mathrm{upd})$ for an arena $(V, V_0, V_1, E, \mathrm{cst})$ consists of a finite set M of memory states, an initialization function $\mathrm{init} \colon V \to M$, and an update function $\mathrm{upd} \colon M \times V \to M$. The update function can be extended to $\mathrm{upd}^* \colon V^+ \to M$ in the usual way: $\mathrm{upd}^*(\rho_0) = \mathrm{init}(\rho_0)$ and $\mathrm{upd}^*(\rho_0 \ldots \rho_n \rho_{n+1}) = \mathrm{upd}(\mathrm{upd}^*(\rho_0 \ldots \rho_n), \rho_{n+1})$. A next-move function (for Player i) $\mathrm{nxt} \colon V_i \times M \to V$ has to satisfy $(v, \mathrm{nxt}(v, m)) \in E$ for all $v \in V_i$ and all $m \in M$. It induces a strategy σ for Player i with memory $\mathcal{M}$ via $\sigma(\rho_0 \ldots \rho_n) = \mathrm{nxt}(\rho_n, \mathrm{upd}^*(\rho_0 \ldots \rho_n))$. A strategy is called finite-state if it can be implemented by a memory structure.

An arena $\mathcal{A} = (V, V_0, V_1, E, \mathrm{cst})$ and a memory structure $\mathcal{M} = (M, \mathrm{init}, \mathrm{upd})$ for $\mathcal{A}$ induce the expanded arena $\mathcal{A} \times \mathcal{M} = (V \times M, V_0 \times M, V_1 \times M, E', \mathrm{cst}')$ where $((v, m), (v', m')) \in E'$ if and only if $(v, v') \in E$ and $\mathrm{upd}(m, v') = m'$, and $\mathrm{cst}'((v, m)(v', m')) = \mathrm{cst}(v, v')$. Every play ρ in $\mathcal{A}$ has a unique extended play $\rho' = (\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2) \ldots$ in $\mathcal{A} \times \mathcal{M}$ defined by $m_0 = \mathrm{init}(\rho_0)$ and $m_{n+1} = \mathrm{upd}(m_n, \rho_{n+1})$, i.e., $m_n = \mathrm{upd}^*(\rho_0 \cdots \rho_n)$. Note that every infix of ρ and the corresponding infix of ρ' have the same cost.

A game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ is reducible to $\mathcal{G}' = (\mathcal{A}', \mathrm{Win}')$ via $\mathcal{M}$, written $\mathcal{G} \leq_{\mathcal{M}} \mathcal{G}'$, if $\mathcal{A}' = \mathcal{A} \times \mathcal{M}$ and every play ρ in $\mathcal{G}$ is won by the player who wins the extended play ρ' in $\mathcal{G}'$, i.e., $\rho \in \mathrm{Win}$ if and only if $\rho' \in \mathrm{Win}'$.

Lemma 1. *Let $\mathcal{G}$ be a game with vertex set V and $W \subseteq V$. If $\mathcal{G} \leq_{\mathcal{M}} \mathcal{G}'$ and Player i has a finite-state winning strategy for $\mathcal{G}'$ from $\{(v, \mathrm{init}(v)) \mid v \in W\}$, then she also has a finite-state winning strategy for $\mathcal{G}$ from W.*

Let $\mathcal{A} = (V, V_0, V_1, E, \mathrm{cst})$ and let $i \in \{0, 1\}$. The i-attractor of $F \subseteq V$ in $\mathcal{A}$, denoted by $\mathrm{Attr}_i^{\mathcal{A}}(F)$, is defined by $\mathrm{Attr}_i^{\mathcal{A}}(F) = \bigcup_{j=0}^{|V|} A_j$, where $A_0 = F$ and

$$A_{j+1} = A_j \cup \{v \in V_i \mid \exists v' \in A_j \text{ such that } (v, v') \in E\}$$
$$\cup \{v \in V_{1-i} \mid \forall v' \text{ with } (v, v') \in E : v' \in A_j\} \ .$$

Player i has a positional strategy on $\mathrm{Attr}_i^{\mathcal{A}}(F) \setminus F$ such that every play that starts in this set and is consistent with the strategy visits F. Such strategies are called attractor strategies.

Cost-Parity Games. Let $\mathcal{A} = (V, V_0, V_1, E, \mathrm{cst})$ be an arena and let $\Omega \colon V \to \mathbb{N}$ be a coloring of its vertices. In all games we are about to define, we interpret the occurrence of a color as request, which has to be answered by visiting a vertex of larger even color at a later position. By imposing conditions on the responses, we obtain three different types of winning conditions. To simplify our notation, let $\mathrm{Ans}(c) = \{c' \in \mathbb{N} \mid c' \geq c \text{ and } c' \text{ even}\}$ be the set of colors that answer a request of color c. Note that $\mathrm{Ans}(c) \subseteq \mathrm{Ans}(c')$ for $c \geq c'$ and $c \in \mathrm{Ans}(c)$ if c is even.

The parity winning condition, denoted by $\mathrm{Parity}(\Omega)$, requires that almost all requests are eventually answered. Equivalently, $\rho \in \mathrm{Parity}(\Omega)$ if and only if the maximal color that occurs infinitely often in ρ is even. The finitary parity condition [8] is a strengthening of the parity condition; it requires the existence of a bound b such that almost all requests are answered within b steps. Formally, given a play $\rho = \rho_0 \rho_1 \rho_2 \cdots$ and $k \in \mathbb{N}$, we define

$$\mathrm{dist}(\rho, k) = \inf\{k' - k \mid k' \geq k \text{ and } \Omega(\rho_{k'}) \in \mathrm{Ans}(\Omega(\rho_k))\}$$

where we use $\inf \emptyset = \infty$. Hence, $\mathrm{dist}(\rho, k)$ is the number of steps between the request at position k and its earliest response. The finitary parity winning condition, denoted by $\mathrm{FinParity}(\Omega)$, is

$$\mathrm{FinParity}(\Omega) = \{\rho \in V^{\omega} \mid \limsup_{k \to \infty} \mathrm{dist}(\rho, k) < \infty\} \ .$$

Note that both the parity and the finitary parity condition do not depend on the cost function. Finally, by not bounding the number of steps between the requests and their responses, but by bounding the *cost* between requests and responses, we obtain the cost-parity condition. Given a play $\rho = \rho_0 \rho_1 \rho_2 \cdots$ and $k \in \mathbb{N}$, we define the cost-of-response by

$$\mathrm{cor}(\rho, k) = \inf\{\mathrm{cst}(\rho_k \cdots \rho_{k'}) \mid k' \geq k \text{ and } \Omega(\rho_{k'}) \in \mathrm{Ans}(\Omega(\rho_k))\} \ .$$

The cost-parity winning condition, denoted by $\mathrm{CostParity}(\Omega)$, is

$$\mathrm{CostParity}(\Omega) = \{\rho \in V^\omega \mid \limsup_{k \to \infty} \mathrm{cor}(\rho, k) < \infty\} \ .$$

Remark 2. $\mathrm{FinParity}(\Omega) \subseteq \mathrm{CostParity}(\Omega) \subseteq \mathrm{Parity}(\Omega)$.

A game $\mathcal{G} = (\mathcal{A}, \mathrm{CostParity}(\Omega))$ is called cost-parity game. In a similar way, we define parity and finitary parity games. Note that if $\mathcal{A}$ contains no increment-edges, then we have $\mathrm{CostParity}(\Omega) = \mathrm{Parity}(\Omega)$, and if $\mathcal{A}$ contains no ε-edges, then $\mathrm{CostParity}(\Omega) = \mathrm{FinParity}(\Omega)$. Hence, cost-parity games generalize both parity and finitary parity games.

Since cost-parity conditions can be shown to be on the third level of the Borel-hierarchy, we obtain the following result as a consequence of the Borel determinacy theorem [15].

Remark 3. Cost-parity games are determined.

Fix a play ρ. We say that a request at position k is answered with cost c, if $\mathrm{cor}(\rho, k) = c$. Note that a request at a position k with an even color is answered with cost 0. Furthermore, we say that a request at position k is unanswered with cost c, if there is no position $k' \geq k$ such that $\Omega(\rho_{k'}) \in \mathrm{Ans}(\Omega(\rho_k))$, but we have $\mathrm{cst}(\rho_k \rho_{k+1} \cdots) = c$, i.e., there are exactly c increment-edges after position k. Finally, we say that a request at position k is unanswered with cost ∞, if there is no position $k' \geq k$ such that $\Omega(\rho_{k'}) \in \mathrm{Ans}(\Omega(\rho_k))$ and we have $\mathrm{cst}(\rho_k \rho_{k+1} \cdots) = \infty$, i.e., there are infinitely many increment-edges after position k. We say that a request is unanswered, if it is unanswered with cost in $\mathbb{N} \cup \{\infty\}$. We often use the following equivalence.

Remark 4. A play ρ has only finitely many unanswered requests if and only if $\rho \in \mathrm{Parity}(\Omega)$.

Example 5. Consider the cost-parity game depicted in Figure 1 where all vertices belong to V_1, and the label of a vertex denotes its name (in the upper part) and its color (in the lower part).

Player 1 wins from $\{a, b, c\}$ by requesting color 1 at vertex a infinitely often and staying at vertex b longer and longer, but also visiting c infinitely often (and thereby answering the request). Note that this strategy is not finite-state. Indeed, one can easily prove that Player 1 does not have a finite-state winning strategy for this game.

On the other hand, Player 0 wins from every other vertex, since Player 1 can raise only finitely many requests from these vertices, albeit these requests are unanswered with cost ∞.

Finally, by turning the self-loop at vertex b into an ε-edge, we obtain a cost-parity game in which Player 0 wins from everywhere.
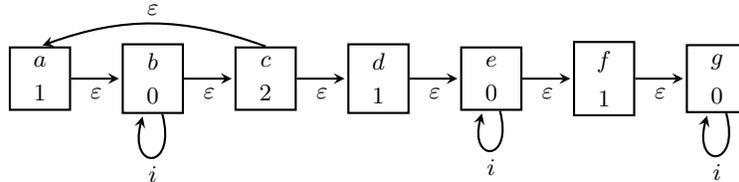
Fig. 1. A cost-parity game.

3 Solving Cost-Parity Games

In this section, we show how to determine the winning regions in a cost-parity game. Our algorithm proceeds in two steps: in Subsection 3.1, we show that it suffices to solve games whose winning condition is a strengthening of the cost-parity condition, the so-called bounded cost-parity condition. Then, in Subsection 3.2, we show how to solve bounded cost-parity games by solving ω-regular games. The main result of this section is the following theorem.

Theorem 6. *To solve a cost-parity game with n vertices, it suffices to solve n parity games which are at most polynomially larger.*

3.1 From Cost-Parity Games to Bounded Cost-Parity Games

The cost-parity condition can be rephrased as follows: $\rho \in \text{CostParity}(\Omega)$ if and only if

> there exists a $b \in \mathbb{N}$ such that all but finitely many requests are answered or unanswered with cost less or equal than b and there are only finitely many unanswered requests.

Note that this allows a finite number of unanswered requests with cost ∞. By disallowing this, we obtain a strengthening of the cost-parity condition and show how solving games with the stronger winning condition solves cost-parity games. Formally, we define the bounded cost-parity winning condition, denoted by $\text{BndCostParity}(\Omega)$, to be the set of plays ρ satisfying the following property:

> there exists a $b \in \mathbb{N}$ such that every request is answered or unanswered with cost less or equal than b and there are only finitely many unanswered requests.

Note that this condition does not allow an unanswered request with cost ∞. A game $(\mathcal{A}, \mathrm{BndCostParity}(\Omega))$ is called bounded cost-parity game.

Example 7. Consider the game in Figure 1, this time with the bounded cost-parity condition: Player 1 wins from every vertex but g by moving to g and then staying there ad infinitum. Every such play contains a request of color 1 that is unanswered with cost ∞.

Bounded cost-parity conditions are on the third level of the Borel-hierarchy. Hence, bounded cost-parity games are also determined. Furthermore, the bounded cost-parity condition is indeed a strengthening of the cost-parity condition.

Remark 8. $\mathrm{BndCostParity}(\Omega) \subseteq \mathrm{CostParity}(\Omega)$.

The following lemma is used below to show that being able to solve bounded cost-parity games suffices to solve cost-parity games.

Lemma 9. *Let* $\mathcal{G} = (\mathcal{A}, \mathrm{CostParity}(\Omega))$ *and let* $\mathcal{G}' = (\mathcal{A}, \mathrm{BndCostParity}(\Omega))$.

1. $W_0(\mathcal{G}') \subseteq W_0(\mathcal{G})$.
2. *If* $W_0(\mathcal{G}') = \emptyset$, *then* $W_0(\mathcal{G}) = \emptyset$.

Proof. 1. This follows directly from Remark 8: a winning strategy for Player 0 in $\mathcal{G}'$ from v is also a winning strategy for her in $\mathcal{G}$ from v.

2. Due to determinacy, if $W_0(\mathcal{G}') = \emptyset$, then we have $W_1(\mathcal{G}') = V$, i.e., from every vertex v, Player 1 has a winning strategy τ_v. Consider a play consistent with τ_v starting in v: either, for every b, there is a request that is not answered with cost at most b, or the maximal color seen infinitely often is odd.

We define a strategy τ for Player 1 as follows: it is guided by a vertex v_{cur} and a counter b_{cur}. Assume a play starts in vertex v. Then, we initialize v_{cur} by v and b_{cur} by 1. The strategy plays the strategy $\tau_{v_{\mathrm{cur}}}$ until a request is not answered with cost b_{cur}. If this is the case, then v_{cur} is set to the current vertex, b_{cur} is incremented, and τ plays according to $\tau_{v_{\mathrm{cur}}}$ (forgetting the history of the play constructed so far).

We show that τ is winning (in the cost-parity game) from every vertex, which implies $W_0(\mathcal{G}) = \emptyset$. Let ρ be a play that is consistent with τ and distinguish two cases: if the counter is incremented infinitely often, then for every bound b, there exists a request that is not answered with cost b. Such a play is winning for Player 1 in the cost-parity game. On the other hand, if b_{cur} incremented only finitely often (say to value b), then there is a suffix ρ' of ρ with some first vertex v that is consistent with the strategy τ_v. Since the counter is not incremented during ρ', every request in ρ' is either answered or unanswered with cost at most b. As ρ' is nevertheless winning for Player 1, the maximal color seen infinitely often during ρ' is odd. As ρ' and ρ only differ by a finite prefix, the maximal color seen infinitely often during ρ is the same as in ρ' and therefore odd. Hence, ρ is winning for Player 1 in the cost-parity game. $\square$

To conclude this subsection, we show how Lemma 9 can be used to solve cost-parity games, provided we are able to solve bounded cost-parity games (which is the subject of the next subsection). Let $\mathcal{A}$ be an arena and Ω a coloring of its vertices, and let $\mathcal{G} = (\mathcal{A}, \mathrm{CostParity}(\Omega))$. The algorithm proceeds by iteratively removing parts of $\mathcal{A}$ that are included in the winning region of Player 0 in $\mathcal{G}$. In each step, one computes the winning region of the current arena w.r.t. the bounded cost-parity condition. Due to Lemma 9.1, this is included in the winning region of Player 0. Furthermore, the attractor of this region also belongs to the winning region of the cost-parity game, since the cost-parity condition is prefix-independent. This continues until Player 0's winning region of the current arena w.r.t. the bounded cost-parity condition is empty. In this situation, Player 1 wins everywhere w.r.t. the cost-parity condition in the current arena due to Lemma 9.2. Since Player 0 cannot leave this region (in the original arena $\mathcal{A}$), it is winning for Player 1 w.r.t. the cost-parity condition in $\mathcal{A}$.

We formalize this intuition in Algorithm 1. Note that removing the attractor does not introduce terminal vertices.

Algorithm 1 A fixed-point algorithm for cost-parity games.

$j \leftarrow 0;\ W_j \leftarrow \emptyset;\ \mathcal{A}_j \leftarrow \mathcal{A}$
repeat
    $j \leftarrow j + 1$
    $X_j \leftarrow W_0(\mathcal{A}_{j-1}, \mathrm{BndCostParity}(\Omega))$
    $W_j \leftarrow W_{j-1} \cup \mathrm{Attr}_0^{\mathcal{A}_{j-1}}(X_j)$
    $\mathcal{A}_j \leftarrow \mathcal{A}_{j-1} \setminus \mathrm{Attr}_0^{\mathcal{A}_{j-1}}(X_j)$
until $X_j = \emptyset$
return W_j

Algorithm 1 terminates after at most $|V|$ iterations, say after n. We claim that the set W_n returned by it is the winning region of Player 0 in $\mathcal{G} = (\mathcal{A}, \mathrm{CostParity}(\Omega))$. To this end, we show $W_n \subseteq W_0(\mathcal{G})$ and $V \setminus W_n \subseteq W_1(\mathcal{G})$, which implies $W_n = W_0(\mathcal{G})$ (and $V \setminus W_n = W_1(\mathcal{G})$).

We start by considering Player 1: we have $W_0(\mathcal{A}_{n-1}, \mathrm{BndCostParity}(\Omega)) = \emptyset$, which implies $W_1(\mathcal{A}_{n-1}, \mathrm{BndCostParity}(\Omega)) = V \setminus W_n$ (as the set of vertices of $\mathcal{A}_{j-1}$ is $V \setminus W_n$). Hence, due to Lemma 9.2, we have $W_1(\mathcal{A}_{n-1}, \mathrm{CostParity}(\Omega)) = V \setminus W_n$, i.e., Player 1 has a winning strategy for the cost-parity game in the arena $\mathcal{A}_{n-1}$ from every vertex. The winning strategies are also winning for Player 1 in the cost-parity game in the arena $\mathcal{A}$, since $V \setminus W_n$ is a trap for Player 0, i.e., all her outgoing edges from $V \setminus W_n$ lead to $V \setminus W_n$. Thus, we indeed have $V \setminus W_n \subseteq W_1(\mathcal{A}, \mathrm{CostParity}(\Omega))$.

Now, consider the winning region of Player 0: the algorithm computes $W_n \supsetneq W_{n-1} \supsetneq \cdots \supsetneq W_0$, where every difference $W_j \setminus W_{j-1}$ is equal to $\mathrm{Attr}_0^{\mathcal{A}_{j-1}}(X_j)$ and $X_j = W_0(\mathcal{A}_{j-1}, \mathrm{BndCostParity}(\Omega))$. We will show in Lemma 15 that Player 0 has a uniform positional winning strategy in every bounded cost-parity game. Here, we combine such strategies σ_j for the winning regions X_j and attractor

strategies $\sigma_j^{\mathcal{A}}$ for $\mathrm{Attr}_0^{\mathcal{A}_{j-1}}(X_j) \setminus X_j$ to a uniform positional winning strategy σ for Player 0 from W_n in $(\mathcal{A}, \mathrm{CostParity}(\Omega))$ via

$$\sigma(wv) = \begin{cases} \sigma_j(wv) & \text{if } v \in X_j, \\ \sigma_j^{\mathcal{A}}(wv) & \text{if } v \in \mathrm{Attr}_0^{\mathcal{A}_{j-1}}(X_j) \setminus X_j. \end{cases}$$

Note that we always have $\sigma(wv) \in X_j$, if $v \in X_j$. This implies the following fact for every play ρ that is consistent with σ and starts in W_n: if ρ visits some X_j, then it either remains in X_j forever or visits W_{j-1}. On the other hand, if it visits some $\mathrm{Attr}_0^{\mathcal{A}_{j-1}}(X_j) \setminus X_j$, then it will visit X_j or W_{j-1}. Thus, ρ has a suffix ρ' that starts in some X_j, stays in X_j forever, and is consistent with σ_j. Hence, $\rho' \in \mathrm{BndCostParity}(\Omega)$, which implies $\rho \in \mathrm{CostParity}(\Omega)$. Thus, $W_n \subseteq W_0(\mathcal{A}, \mathrm{CostParity}(\Omega))$.

Example 10. Running on the cost-parity game of Figure 1, Algorithm 1 computes $X_1 = \{g\}$ and $W_1 = \{f, g\}$, $X_2 = \{e\}$ and $W_2 = \{d, e, f, g\}$, and $X_3 = \emptyset$.

3.2 From Bounded Cost-Parity Games to ω-regular Games

Next, we show how to solve bounded cost-parity games. Let $\mathcal{A}$ be an arena. In this subsection, we assume that no vertex of $\mathcal{A}$ has both incoming increment- and ε-edges. This can be achieved by subdividing every increment-edge $e = (v, v')$: we add a new vertex $\mathrm{sub}(e)$ and replace e by $(v, \mathrm{sub}(e))$ (which is an increment-edge) and by $(\mathrm{sub}(e), v')$ (which is an ε-edge). Now, only the newly added vertices have incoming increment-edges, but they do not have incoming ε-edges. Furthermore, it is easy to see that Player i wins from v in the original game if and only if she wins from v in the modified game[4].

Assuming this convention, we say that a vertex is an increment-vertex, if it has an incoming increment-edge (which implies that all incoming edges have an increment). Let F be the set of increment-vertices and denote the set of plays with finite cost by $\mathrm{coB\ddot{u}chi}(F)$. Furthermore, let $\mathrm{RR}(\Omega)$ denote the set of plays ρ in which every request is answered. Using these definitions, let

$$\mathrm{PCRR}(\Omega) = \big(\mathrm{Parity}(\Omega) \cap \mathrm{coB\ddot{u}chi}(F)\big) \cup \mathrm{RR}(\Omega) .$$

As $\mathrm{PCRR}(\Omega)$ is ω-regular, a game with winning condition $\mathrm{PCRR}(\Omega)$ is determined and both players have uniform finite-state winning strategies [6]. Furthermore, $\mathrm{PCRR}(\Omega)$ is weaker than $\mathrm{BndCostParity}(\Omega)$.

Remark 11. $V^\omega \setminus \mathrm{PCRR}(\Omega) \subseteq V^\omega \setminus \mathrm{BndCostParity}(\Omega)$.

Note that the converse implication is false, as the request-response condition does not bound the cost between requests and their responses. However, every finite-state winning strategy bounds the distance between requests and their responses, and thereby also the cost.

---

[4] Where we color $\mathrm{sub}(e)$ by $\Omega(v')$.

Lemma 12. *Let $\mathcal{G} = (\mathcal{A}, \mathrm{BndCostParity}(\Omega))$, and $\mathcal{G}' = (\mathcal{A}, \mathrm{PCRR}(\Omega))$. Then, $W_i(\mathcal{G}) = W_i(\mathcal{G}')$ for $i \in \{0, 1\}$.*

Proof. We have $W_1(\mathcal{G}') \subseteq W_1(\mathcal{G})$ due to Remark 11. Thus, it suffices to show $W_0(\mathcal{G}') \subseteq W_0(\mathcal{G})$. So, let σ be a uniform finite-state winning strategy for Player 0 for $\mathcal{G}'$ (which exists, since $\mathrm{PCRR}(\Omega)$ is ω-regular). We argue that σ is also a uniform winning strategy for Player 0 for $\mathcal{G}$: let ρ be consistent with σ, which implies $\rho \in \mathrm{PCRR}(\Omega)$.

If ρ satisfies $\mathrm{Parity}(\Omega)$ and has only finitely many increments, then every request is answered or unanswered with bounded cost and there are only finitely many unanswered requests, i.e., ρ is winning for Player 0 in $\mathcal{G}$.

To conclude, we consider the case $\rho \in \mathrm{RR}(\Omega)$. Since σ is a finite-state winning strategy for this condition, there is a bound b (which only depends on the size of σ and the size of the arena) such that every request is answered within b steps, and therefore also with cost at most b. Hence, ρ is winning for Player 0 in $\mathcal{G}$. □

So, to determine the winning regions of a bounded cost-parity game it suffices to solve an ω-regular game with winning condition $\mathrm{PCRR}(\Omega)$. This condition can be reduced to a parity condition using a *small* memory structure that keeps track of the largest unanswered request and goes to a special state $\bot$ if this request is answered. Using this memory structure, the $\mathrm{RR}(\Omega)$-condition reduces to a Büchi condition requiring the state $\bot$ to be visited infinitely often. We are now left with the union of a Büchi condition and an intersection of a parity and a co-Büchi condition. First, by coloring the increment-vertices of $\mathcal{A}$ by an odd color that is larger than $\max \Omega(V)$ (and leaving the colors of all other vertices unchanged), we obtain a new parity condition that is equivalent to the intersection of $\mathrm{Parity}(\Omega)$ and $\mathrm{coBüchi}(F)$. Using a similar construction, we can turn the union of the new parity condition and of the Büchi condition into an equivalent parity condition. Thus, to determine the winning regions of a game with winning condition $\mathrm{PCRR}(\Omega)$, if suffices to solve a linearly larger parity game. This also concludes the proof of Theorem 6: Algorithm 1 terminates after at most n iterations and solves a *small* parity game in each iteration.

Furthermore, from the previous observations we obtain an upper bound on the memory requirements for both players, which we improve in the next section.

Remark 13. In every bounded cost-parity game, both players have uniform finite-state winning strategies with $\ell + 1$ memory states, where ℓ is the number of odd colors in $\Omega(V)$.

3.3 Computational Complexity

Using the reduction from bounded cost-parity to parity games, we can determine the computational complexity of solving such games. Furthermore, since we can solve cost-parity games by solving bounded cost-parity games, we obtain the same result for cost-parity games.

Theorem 14. *The following problems are in $\mathbf{NP} \cap \mathbf{coNP}$:*

1. *Given a bounded cost-parity game $\mathcal{G}$ and a vertex v, is $v \in W_0(\mathcal{G})$?*
2. *Given a cost-parity game $\mathcal{G}$ and a vertex v, is $v \in W_0(\mathcal{G})$?*

Proof. 1. Since bounded cost-parity games are reducible to polynomially larger parity games, which can be solved in $\mathbf{NP} \cap \mathbf{coNP}$, the result follows.

2. We rely on Algorithm 1, which has to determine the winning regions of at most $|V|$ bounded cost-parity games, which we just have shown to be in $\mathbf{NP} \cap \mathbf{coNP}$. □

4 Half-Positional Determinacy of (Bounded) Cost-Parity Games

In this section, we show that the finite-state strategy for Player 0 in a bounded cost-parity game obtained by the equivalence with the ω-regular PCRR-condition is not optimal in terms of memory size: Player 0 needs no memory at all to win a bounded cost-parity game.

Lemma 15. *In bounded cost-parity games, Player 0 has uniform positional winning strategies.*

As we have already explained while proving the correctness of Algorithm 1, the previous lemma implies a similar result for cost-parity games.

Theorem 16. *In cost-parity games, Player 0 has uniform positional winning strategies.*

For bounded cost-parity games we have already proved the existence of uniform finite-state winning strategies (see Remark 13). Hence, it remains to show how to eliminate the memory. To this end, we define a so-called scoring function for bounded cost-parity games that measures the quality of a play prefix (from Player 0's vantage point) by keeping track of the largest unanswered request, the number of increment-edges traversed since it was raised, and how often each odd color was seen since the last increment-edge.

In the following, fix an arena $(V, V_0, V_1, E, \mathrm{cst})$ and a coloring function $\Omega \colon V \to \mathbb{N}$. Let $\Omega(V) \subseteq \{0, \ldots, d\}$, where we assume d to be odd, and define $\ell = \frac{d+1}{2}$ to denote the number of odd colors in $\{0, \ldots, d\}$. A (score-) sheet is a vector $(c, n, s_d, s_{d-2}, \ldots, s_1) \in \mathbb{N}^{2+\ell}$ containing the following information about a play prefix w:

- c denotes the largest unanswered request in w, i.e., the largest odd color in w that is not followed by a color in $\mathrm{Ans}(c)$.
- n denotes the cost of the suffix starting with the *first* unanswered occurrence of c in w.
- $s_{c'}$ (here, c' is odd) denotes the number of times c' occurred in w since the first unanswered occurrence of c, since the last increment-edge was traversed, or since the last time a color larger than c' occurred in w, depending on which happened last.

Finally, we use the empty sheet $\bot$ for play prefixes without unanswered requests. To define the sheet of a play prefix inductively, we define the initial sheet $i(v)$ of a vertex v by

$$i(v) = \begin{cases} \bot & \text{if } \Omega(v) \text{ is even,} \\ (\Omega(v), 0, 0, \ldots, 0) & \text{if } \Omega(v) \text{ is odd,} \end{cases}$$

and the effect of traversing the edge (v, v') on a sheet: we define $\bot \oplus (v, v') = i(v')$ and $(c, n, s_d, \ldots, s_1) \oplus (v, v')$ is given by the following case distinction, where we denote $\Omega(v')$ by c'

$$\begin{cases} \bot & \text{if } c' \in \text{Ans}(c), \\ (c, n+1, 0, \ldots, 0) & \text{if } c' < c \text{ and even, and } \text{cst}(v, v') = i, \\ (c, n, s_d, \ldots, s_{c'+1}, 0, \ldots, 0) & \text{if } c' < c \text{ and even, and } \text{cst}(v, v') = \varepsilon, \\ (c', 0, 0, \ldots, 0) & \text{if } c' > c \text{ and odd,} \\ (c, n+1, 0, \ldots, 0) & \text{if } c' \leq c \text{ and odd, and } \text{cst}(v, v') = i, \\ (c, n, s_d, \ldots, s_{c'+2}, s_{c'}+1, 0, \ldots, 0) & \text{if } c' \leq c \text{ and odd, and } \text{cst}(v, v') = \varepsilon. \end{cases}$$

Now, we define $\text{Sh}(v) = i(v)$ for every $v \in V$, and $\text{Sh}(wvv') = \text{Sh}(wv) \oplus (v, v')$ for every $w \in V^*$ and every $(v, v') \in E$.

The reversed ordering of the score values $s_d, \ldots, s_1$ in the sheets is due to the max-parity condition, in which larger colors are more important than smaller ones. This is reflected by the fact that we compare sheets in the lexicographical order induced by $<$ on its components and add $\bot$ as minimal element. For example, $(3, 3, 0, 1, 1) < (3, 3, 1, 0, 7)$ and $\bot < s$ for every sheet $s \neq \bot$. As usual, we write $s \leq s'$, if $s = s'$ or $s < s'$. We say that a sheet $(c, n, s_d, \ldots, s_1)$ is bounded by $b \in \mathbb{N}$, if we have $n \leq b$ and $s_c \leq b$ for every c. Also, $\bot$ is bounded by every b. The following lemma shows that Sh is a congruence w.r.t. $\leq$. Here $\text{lst}(x)$ denotes the last vertex of the non-empty finite play x.

Lemma 17. *If* $\text{lst}(x) = \text{lst}(y)$ *and* $\text{Sh}(x) \leq \text{Sh}(y)$, *then* $\text{Sh}(xv) \leq \text{Sh}(yv)$ *for every* $v \in V$.

Let $\text{Sh}(w) = (c, n, s_d, \ldots, s_1)$. To simplify our notation in the following proofs, we define $\text{Req}(w) = c$, $\text{ReqCst}(w) = n$, and $\text{Sc}_{c'}(w) = s_{c'}$. For w with $\text{Sh}(w) = \bot$ we leave these functions undefined.

Before we begin the proof, we state the following useful facts.

Remark 18.

1. If $\text{Sc}_c(w) > 0$, then $c \leq \text{Req}(w)$.
2. If $\text{Sh}(w) = \bot$, then $\text{Sh}(wv) = \text{Sh}(v)$.
3. If $\text{Req}(wv) \neq \text{Req}(w)$, then $\text{Sh}(wv) = \text{Sh}(v)$.
4. Let $\Omega(v)$ be odd. Then, $\text{Sh}(wv) \geq \text{Sh}(v)$.

Proof (Proof of Lemma 17). If $\text{Sh}(x) = \text{Sh}(y)$, then $\text{Sh}(xv) = \text{Sh}(yv)$, since the sheets of xv and yv only depend on the sheets of x and y (which are equal) and the last edges of xv and yv (which are also equal).

So, consider the case $\text{Sh}(x) < \text{Sh}(y)$. First, assume we have $\text{Sh}(x) = \bot$, which implies $\text{Sh}(xv) = \text{Sh}(v)$ due to Remark 18.2. If $\Omega(v)$ is even, then $\text{Sh}(v) = \bot$ and we are done, since $\bot$ is the minimal element. Otherwise, applying Remark 18.4 to yv yields the desired result.

So, assume we have $\text{Sh}(x) \neq \bot$, which implies $\text{Sh}(y) \neq \bot$. We proceed by case distinction over the first position where the sheets differ:

1. if $\text{Req}(x) < \text{Req}(y)$, we have to consider three subcases:
   (a) If $\Omega(v) \in \text{Ans}(\text{Req}(y))$, then $\text{Sh}(yv) = \text{Sh}(xv) = \bot$, since $\Omega(v) \in \text{Ans}(\text{Req}(y)) \subseteq \text{Ans}(\text{Req}(x))$.
   (b) If $\Omega(v) \notin \text{Ans}(\text{Req}(y))$ and $\Omega(v)$ is even, then we have $\text{Req}(y) = \text{Req}(yv)$ and $\text{Sh}(xv) = \bot$ (if $\Omega(v) \in \text{Ans}(\text{Req}(x))$) or $\text{Req}(xv) = \text{Req}(x)$ (if $\Omega(v) \notin \text{Ans}(\text{Req}(x))$). In both cases, we have $\text{Sh}(xv) < \text{Sh}(yv)$.
   (c) If $\Omega(v) \notin \text{Ans}(\text{Req}(y))$ and $\Omega(v)$ is odd, then we again distinguish three subcases:
      i. If $\Omega(v) > \text{Req}(y)$, then we have $\text{Sh}(yv) = \text{Sh}(xv) = (\Omega(v), 0, 0, \ldots, 0)$, since $\Omega(v)$ is larger than $\text{Req}(y)$ and $\text{Req}(x)$.
      ii. If $\Omega(v) = \text{Req}(y)$, then we have $\text{Req}(xv) = \Omega(v)$. Thus, we have $\text{Req}(xv) \neq \text{Req}(x)$ and an application of Remark 18.3 and 18.4 to yv concludes this case.
      iii. If $\Omega(v) < \text{Req}(y)$, then we have

      $$\text{Req}(xv) = \max\{\text{Req}(x), \Omega(v)\} < \text{Req}(y) = \text{Req}(yv) \ ,$$

      and therefore $\text{Sh}(xv) < \text{Sh}(yv)$.

2. So, assume we have $\text{Req}(x) = \text{Req}(y)$ and $\text{ReqCst}(x) < \text{ReqCst}(y)$. If $\text{Sh}(yv) = \bot$, then also $\text{Sh}(xv) = \bot$, since their Req-values are equal. Hence, we may assume $\text{Sh}(yv) \neq \bot$, which implies $\text{Sh}(xv) \neq \bot$ and $\text{Req}(yv) = \text{Req}(xv)$. We consider two subcases:
   (a) $\text{Req}(yv) \neq \text{Req}(y)$, which implies $\text{Req}(xv) \neq \text{Req}(x)$ due to $\text{Req}(x) = \text{Req}(y)$, we have $\text{Sh}(xv) = \text{Sh}(yv) = \text{Sh}(v)$ due to Remark 18.3.
   (b) If $\text{Req}(yv) = \text{Req}(y)$, which implies $\text{Req}(xv) = \text{Req}(x)$, then we have

   $$\begin{aligned} \text{ReqCst}(xv) =\ & \text{ReqCst}(x) + \text{cst}(\text{lst}(x), v) \\ < & \text{ReqCst}(y) + \text{cst}(\text{lst}(y), v) = \text{ReqCst}(yv) \ , \end{aligned}$$

   due to $\text{lst}(x) = \text{lst}(y)$.

3. Finally, we consider the case where $\text{Req}(x) = \text{Req}(y)$ and $\text{ReqCst}(x) = \text{ReqCst}(y)$, which implies $\text{Req}(xv) = \text{Req}(yv)$ and $\text{ReqCst}(xv) = \text{ReqCst}(yv)$. Then, there exists an odd c in the range $\{1, \ldots, d\}$ such that $\text{Sc}_{c'}(x) = \text{Sc}_{c'}(y)$ for every $c' > c$ and $\text{Sc}_c(x) < \text{Sc}_c(y)$.
   If $\text{Req}(yv) \neq \text{Req}(y)$, then we again have $\text{Sh}(xv) = \text{Sh}(yv)$ due to Remark 18.3. Furthermore, if the Req-values do not change, but $\text{ReqCst}(yv) \neq \text{ReqCst}(y)$, then all scores of yv and xv are set to zero. Thus, we have $\text{Sh}(yv) = \text{Sh}(xv)$ in this case as well.
   So, assume both the Req-values and the ReqCst-values are unchanged. There are again three subcases:

(a) If $\Omega(v) \in \mathrm{Ans}(c)$, then we have $\mathrm{Sc}_{c'}(xv) = \mathrm{Sc}_{c'}(yv) = 0$ for every $c' < \Omega(v)$ and

$$\mathrm{Sc}_{c'}(xv) = \mathrm{Sc}_{c'}(x) = \mathrm{Sc}_{c'}(y) = \mathrm{Sc}_{c'}(yv)$$

for every $c' > \Omega(v) > c$, i.e., the scores of xv and yv are equal, and therefore $\mathrm{Sh}(xv) = \mathrm{Sh}(yv)$.

(b) If $\Omega(v) = c$, then we have

$$\mathrm{Sc}_c(xv) = \mathrm{Sc}_c(x) + 1 < \mathrm{Sc}_c(y) + 1 = \mathrm{Sc}_c(yv)$$

and

$$\mathrm{Sc}_{c'}(xv) = \mathrm{Sc}_c(x) = \mathrm{Sc}_{c'}(y) = \mathrm{Sc}_{c'}(yv)$$

for every $c' > c$. Thus, Sc_c again witnesses $\mathrm{Sh}(xv) < \mathrm{Sh}(yv)$.

(c) If $\Omega(v) \notin \mathrm{Ans}(c)$ and $\Omega(v) \neq c$, then we have $\mathrm{Sc}_{c'}(xv) = \mathrm{Sc}_{c'}(yv)$ for every $c' > c$ and $\mathrm{Sc}_c(xv) < \mathrm{Sc}_c(yv)$. Thus, Sc_c again witnesses $\mathrm{Sh}(xv) < \mathrm{Sh}(yv)$. □

We begin the proof of Lemma 15 by showing that the sheets of a play ρ being bounded is a sufficient condition for ρ satisfying bounded cost-parity.

Lemma 19. *If there exists a bound b such that the sheets of all prefixes of a play ρ are bounded by b, then $\rho \in \mathrm{BndCostParity}(\Omega)$.*

Proof. Let all sheets be bounded by b. Then, every request is answered or unanswered with cost at most $b \cdot (\ell - 1)$ (recall that ℓ denotes the number of odd colors), since the ReqCst-value is updated along every increment-edge that is traversed. Hence, after b increments, the Req-value has to be increased, which can only happen $\ell - 1$ times. Thus, if there are infinitely many increment-edges in ρ, then every request is answered with cost at most $b \cdot (\ell - 1)$, i.e, the bounded cost-parity condition is satisfied. If there are only finitely many increment edges, then we have to show that the parity condition is satisfied by ρ. Assume the maximal color that appears infinitely often in ρ, call it c, is odd. Then, after the last increment-edge is traversed and after the vertices that appear only finitely often do not appear any more, Sc_c is incremented with each occurrence of the color c, but never reset to 0. This contradicts the boundedness of the sheets. Thus, ρ satisfies the parity condition. □

Next, we show that finite-state winning strategies uniformly bound the sheets.

Lemma 20. *Let σ be a uniform finite-state winning strategy σ (of size m) for Player 0 in a bounded cost-parity game $\mathcal{G} = (\mathcal{A}, \mathrm{BndCostParity}(\Omega))$. Furthermore, let ρ be starting in $W_0(\mathcal{G})$ and be consistent with σ. Then, the sheets of all prefixes of ρ are bounded by $m \cdot |V|$.*

Proof. Assume the sheets are not bounded by $b = m \cdot |V|$. First, we assume there is a prefix w such that $\mathrm{ReqCst}(w) \geq b + 1$. Then, there is a vertex with odd color c in w that is followed by $b + 1$ increment-edges, but no vertex of larger even

color before the end of the $(b+1)$-th increment-edge after the request of c. Hence, there are two positions in this interval that have the same vertex, the memory structure implementing σ assumes the same state after both positions, and there is at least one increment-edge between these positions. Hence, there is also a play consistent with σ and starting in $W_0(\mathcal{G})$ that contains an unanswered request with cost ∞. However, this contradicts the fact that σ is a winning strategy.

Now, assume we have a prefix w with $\mathrm{Sc}_c(w) \geq b+1$. In this case, there is an infix between two positions of w that have the same vertex, the memory structure implementing σ assumes the same state after both positions, and the maximal color between these positions is c. This implies the existence of a play consistent with σ and starting in $W_0(\mathcal{G})$ whose maximal color seen infinitely often is odd. Such a play has infinitely many unanswered requests, which again contradicts the fact that σ is a winning strategy. $\qquad\square$

Now we are able to prove our main technical result of this section: Player 0 has a uniform positional winning strategy in every bounded cost-parity game (and therefore also in every cost-parity game).

Proof (Proof of Lemma 15). Fix some uniform finite-state winning strategy σ' for Player 0 in a bounded cost-parity game $\mathcal{G}$. For every $v \in W_0(\mathcal{G})$, let P_v denote the set of play prefixes that begin in $W_0(\mathcal{G})$, are consistent with σ', and end in v. Due to Lemma 20, the sheets of the prefixes in P_v are bounded by some b. Thus, for every v the set $\{\mathrm{Sh}(w) \mid w \in P_v\}$ is finite. Hence, there exists a play prefix $\max_v \in P_v$ such that $\mathrm{Sh}(w) \leq \mathrm{Sh}(\max_v)$ for every $w \in P_v$.

We define a uniform positional strategy σ by $\sigma(wv) = \sigma'(\max_v)$ and claim that it is a uniform winning strategy for $\mathcal{G}$. An inductive application of Lemma 17 shows that we have $\mathrm{Sh}(\rho_0 \cdots \rho_n) \leq \mathrm{Sh}(\max_{\rho_n})$ for every n and every play ρ that is consistent with σ. Hence, the sheets of ρ are bounded by b, which implies $\rho \in \mathrm{BndCostParity}(\Omega)$ due to Lemma 19. $\qquad\square$

Note that the previous proof only relies on the following properties:

1. The score-sheets are totally ordered.
2. The score-sheet function is a congruence w.r.t. this order.
3. If the score-sheets of a play are bounded, then it is winning for Player 0.
4. A finite-state winning strategy uniformly bounds the score-sheets of all plays.

It follows that for every winning condition for which one can define a scoring function meeting these assumptions, one can turn a finite-state winning strategy into a positional one.

5 Cost-Streett Games

In this section, we introduce cost-Streett games which generalize both Streett games and finitary Streett games [8]. We first present an algorithm to solve these games following the same ideas as in the previous sections, and prove **EXPTIME**-completeness of the corresponding decision problem. From our algorithm, we

deduce finite-state determinacy for Player 0, while Player 1 needs infinite memory in general.

To simplify our notation, let $[d] = \{0, \ldots, d-1\}$. A d-dimensional cost-arena has the form $\mathcal{A} = (V, V_0, V_1, E, (\text{cst}_\ell)_{\ell \in [d]})$ where the first four components are as usual and where each cst_ℓ is a mapping from E to $\{\varepsilon, i\}$. Again, each of these functions induces a cost-function on (infixes of) plays, denoted by cst_ℓ as well.

Let $\Gamma = (Q_\ell, P_\ell)_{\ell \in [d]}$ be a collection of (Streett) pairs of subsets of V, i.e., $Q_\ell, P_\ell \subseteq V$. We define

$$
\text{StDist}_\ell(\rho, k) = \begin{cases} 0 & \text{if } \rho_k \notin Q_\ell, \\ \inf\{\text{cst}_\ell(\rho_k \cdots \rho_{k'}) \mid \rho_{k'} \geq \rho_k \text{ and } \rho_{k'} \in P_\ell\} & \text{if } \rho_k \in Q_\ell, \end{cases}
$$

where we use $\inf \emptyset = \infty$, and $\text{StDist}(\rho, k) = \max\{\text{StDist}_\ell(\rho, k) \mid \ell \in [d]\}$. Using this, we define the following three winning conditions:

– the (classical) Streett condition $\text{Streett}(\Gamma)$ requires for every ℓ that P_ℓ is visited infinitely often if Q_ℓ is.
– the request-response condition $\text{RR}(\Gamma)$ requires for every ℓ that every visit to Q_ℓ is answered by a later visit to P_ℓ.
– the cost-Streett condition $\text{CostStreett}(\Gamma) = \{\rho \mid \limsup_{k \to \infty} \text{StDist}(\rho, k) < \infty\}$.

A cost-Streett game $(\mathcal{A}, \Gamma)$ consists of a d-dimensional arena and a collection Γ of d Streett pairs. Streett and request-response games are defined accordingly. As for cost-parity games, a cost-Streett game in which every edge is an increment-edge (w.r.t. all cst_ℓ) is a finitary Streett game and a cost-Streett game in which every edge is an epsilon-edge (w.r.t. all cst_ℓ) is a Streett game. Furthermore, just as classical Streett games subsume parity games, cost-Streett games subsume cost-parity games.

Remark 21. Cost-Streett games are determined.

Our main theorem is proved along the same lines as Theorem 6.

Theorem 22. *To solve a cost-Streett game with n vertices, it suffices to solve n Streett games which are exponentially larger (in the number of Streett pairs).*

The requests at position k are answered with cost c, if $\text{StDist}(\rho, k) = c$; that the requests are unanswered with cost c, if $\text{StDist}(\rho, k) = \infty$, but there are at most c many increment-edges after position k (w.r.t. all cst_ℓ such that $\rho_k \in Q_\ell$); and that the requests are unanswered with cost ∞, if $\text{StDist}(\rho, k) = \infty$ and there are infinitely many increment-edges after position k (w.r.t. some cst_ℓ such that $\rho_k \in Q_\ell$).

As in the case of cost-parity games, we begin by introducing a strengthening of the cost-Streett condition. The bounded cost-Streett condition, denoted by $\text{BndCostStreett}(\Gamma)$, is the set of plays ρ that satisfy the following condition:

there exists a b such that all requests are answered or unanswered with cost at most b, and there are only finitely many unanswered requests.

Similarly to the bounded cost-parity condition, having only finitely many unanswered requests is equivalent to the Streett condition being satisfied. We have $\mathrm{BndCostStreett}(\Gamma) \subseteq \mathrm{CostStreett}(\Gamma)$ and cost-Streett and bounded cost-Streett games have the same relationship as cost-parity and bounded cost-parity games.

Lemma 23. *Let* $\mathcal{G} = (\mathcal{A}, \mathrm{CostStreett}(\Gamma))$, *and let* $\mathcal{G}' = (\mathcal{A}, \mathrm{BndCostStreett}(\Gamma))$.

1. $W_0(\mathcal{G}') \subseteq W_0(\mathcal{G})$.
2. *If* $W_0(\mathcal{G}') = \emptyset$, *then* $W_0(\mathcal{G}) = \emptyset$.

The proof is analogously to the one for Lemma 9. Also, Algorithm 1 (where X_j is now Player 0's winning region in the bounded cost-Streett game) works for this pair of winning conditions as well. Hence, we need to solve bounded cost-Streett games. To this end, we again assume for every ℓ that no vertex has both an incoming increment-edge (w.r.t. cst_ℓ) and an incoming epsilon-edge (again, w.r.t. cst_ℓ). Having different types of incoming edges with respect to different cost-functions is allowed. This property can again be established by subdividing edges. Assuming this, let F_ℓ denote the vertices with incoming increment-edges w.r.t. cst_ℓ. Then, $\mathrm{coB\ddot{u}chi}(F_\ell) = \{\rho \mid \mathrm{cst}_\ell(\rho) < \infty\}$ is the set of plays with finitely many increment-edges w.r.t. cst_ℓ. Finally, we define the ω-regular condition

$$\mathrm{SCRR}(\Gamma) = \bigcap_{\ell \in [d]} \big(\mathrm{Streett}(Q_\ell, P_\ell) \cap \mathrm{coB\ddot{u}chi}(F_\ell)\big) \ \cup \ \mathrm{RR}(Q_\ell, P_\ell) \ .$$

Lemma 24. *Let* $\mathcal{G} = (\mathcal{A}, \mathrm{BndCostStreett}(\Gamma))$, *and* $\mathcal{G}' = (\mathcal{A}, \mathrm{SCRR}(\Gamma))$. *Then,* $W_i(\mathcal{G}) = W_i(\mathcal{G}')$ *for* $i \in \{0, 1\}$.

The proof is again analogously to the one for Lemma 12 and relies on finite-state determinacy of ω-regular games. To solve $(\mathcal{A}, \mathrm{SCRR}(\Gamma))$ we add a memory structure of size 2^d that keeps track of the open requests during a play, which allows to turn the request-response conditions into Büchi conditions. Let A_ℓ denote the vertices in which request ℓ is not pending. Then, the condition obtained in the reduction is equivalent to

$$\bigcap_{\ell \in [d]} \mathrm{Streett}(Q_\ell, P_\ell \cup A_\ell) \ \cap \ \mathrm{Streett}(F_\ell, A_\ell) \ ,$$

which is a Streett condition with $2d$ pairs in an arena of size $|\mathcal{A}| \cdot 2^d$. Hence, using the algorithm for Streett games from [16], we can solve the resulting game in exponential time (in the size of $(\mathcal{A}, \mathrm{SCRR}(\Gamma))$), even though the arena is of exponential size (but only in d). Together with the **EXPTIME**-hardness of solving finitary Streett games [8], which are a special case of cost-Streett games, we obtain the following result.

Theorem 25. *The problem "Given a cost-Streett game* $\mathcal{G}$ *and a vertex* v, *is* $v \in W_0(\mathcal{G})$?" *is* **EXPTIME**-*complete.*

Furthermore, the reduction described above (and the memory requirements for Streett games [11]) yields upper bounds on the memory requirements in bounded cost-Streett and cost-Streett games. Player 1 needs in general infinite memory in cost-Streett games.

Corollary 26.

- *Player 0 has finite-state winning strategies of size $2^d \cdot (2d)!$ in bounded cost-Streett games and cost-Streett games.*
- *Player 1 has finite-state winning strategies of size 2^d in bounded cost-Streett games.*

6   Conclusion

We introduced infinite games with cost conditions, generalizing both classical conditions and finitary conditions. For cost-parity games, we proved half-positional determinacy and that solving these games is not harder than solving parity games. Furthermore, the corresponding decision problem is in $\mathbf{NP} \cap \mathbf{coNP}$. For cost-Streett games, we showed that Player 0 has finite-state winning strategies and that solving these games is not harder than solving finitary Streett games and can be done by solving linearly many (classical) Streett games of exponential size (in the number of Streett pairs). Our results unify the previous results on both classical and finitary variants. Table 1 sums up all these results.

winning condition	computational complexity	memory Player 0	memory Player 1
parity	$\mathbf{NP} \cap \mathbf{coNP}$	positional	positional
finitary parity	$\mathbf{PTIME}$	positional	infinite
cost-parity	$\mathbf{NP} \cap \mathbf{coNP}$	positional	infinite
Streett	$\mathbf{coNP}$-complete	finite	positional
finitary Streett	$\mathbf{EXPTIME}$-complete	finite	infinite
cost-Streett	$\mathbf{EXPTIME}$-complete	finite	infinite

Table 1. Overview of results

There are at least two directions to extend these results: on the one hand, one could consider infinite arenas, e.g., configuration graphs of pushdown systems. This is already open for finitary games and requires new ideas, since our techniques rely heavily on the finiteness of the arena.

On the other hand, one could extend the cost-functions by decrement-edges. Here, different definitions are possible, depending on whether one allows negative costs (i.e., is $0-1$ equal to 0 or to -1?) and whether the cost-of-response measures the cost between a request and the first answer or whether it measures the cost between a request and the cheapest subsequent answer (which is not necessarily the first, if we allow decrements).

References

1. Mikołaj Bojańczyk. A bounding quantifier. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *CSL*, volume 3210 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.
2. Mikołaj Bojańczyk. Weak MSO with the unbounding quantifier. *Theory Comput. Syst.*, 48(3):554–576, 2011.
3. Mikołaj Bojańczyk and Thomas Colcombet. Bounds in ω-regularity. In LICS [14], pages 285–296.
4. Mikołaj Bojańczyk and Szymon Toruńczyk. Weak MSO+U over infinite trees. In Christoph Dürr and Thomas Wilke, editors, *STACS*, volume 14 of *LIPIcs*, pages 648–660. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.
5. Tomás Brázdil, Krishnendu Chatterjee, Antonín Kučera, and Petr Novotný. Efficient controller synthesis for consumption games with multiple resource types. *CoRR*, abs/1202.0796, 2012.
6. J. Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:pp. 295–311, 1969.
7. Krishnendu Chatterjee and Laurent Doyen. Energy parity games. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *ICALP (2)*, volume 6199 of *Lecture Notes in Computer Science*, pages 599–610. Springer, 2010.
8. Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. Finitary winning in ω-regular games. *ACM Trans. Comput. Log.*, 11(1), 2009.
9. Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *ICALP (2)*, volume 5556 of *Lecture Notes in Computer Science*, pages 139–150. Springer, 2009.
10. Uli Fahrenberg, Line Juhl, Kim G. Larsen, and Jirí Srba. Energy games in multi-weighted automata. In Antonio Cerone and Pekka Pihlajasaari, editors, *ICTAC*, volume 6916 of *Lecture Notes in Computer Science*, pages 95–115. Springer, 2011.
11. Yuri Gurevich and Leo Harrington. Trees, automata, and games. In *STOC*, pages 60–65. ACM, 1982.
12. Kosaburo Hashiguchi. Limitedness theorem on finite automata with distance functions. *J. Comput. Syst. Sci.*, 24(2):233–244, 1982.
13. Marcin Jurdziński. Small progress measures for solving parity games. In Horst Reichel and Sophie Tison, editors, *STACS*, volume 1770 of *LNCS*, pages 290–301. Springer, 2000.
14. *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*. IEEE Computer Society, 2006.
15. Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.
16. Nir Piterman and Amir Pnueli. Faster solutions of Rabin and Streett games. In LICS [14], pages 275–284.
17. Michael Vanden Boom. Weak cost monadic logic over infinite trees. In Filip Murlak and Piotr Sankowski, editors, *MFCS*, volume 6907 of *Lecture Notes in Computer Science*, pages 580–591. Springer, 2011.
18. Igor Walukiewicz. Pushdown processes: Games and model-checking. *Inf. Comput.*, 164(2):234–263, 2001.
19. Martin Zimmermann. Optimal bounds in parametric LTL games. In Giovanna D'Agostino and Salvatore La Torre, editors, *GandALF*, volume 54 of *EPTCS*, pages 146–161, 2011.