Window Parity Games: An Alternative Approach Toward Parity Games with Time Bounds*

Véronique Bruyère Quentin Hautem
Computer Science Department, Université de Mons (UMONS), Belgium
Mickael Randour
Computer Science Department, Université libre de Bruxelles (ULB), Belgium
Computer Science Department, Université de Mons (UMONS), Belgium

Classical objectives in two-player zero-sum games played on graphs often deal with limit behaviors of infinite plays: e.g., *mean-payoff* and *total-payoff* in the quantitative setting, or *parity* in the qualitative one (a canonical way to encode ω-regular properties). Those objectives offer powerful abstraction mechanisms and often yield nice properties such as memoryless determinacy. However, their very nature provides no guarantee on time bounds within which something good can be witnessed. In this work, we consider two approaches toward inclusion of *time bounds* in parity games. The first one, *parity-response* games, is based on the notion of finitary parity games [8] and parity games with costs [16, 29]. The second one, *window parity* games, is inspired by window mean-payoff games [5]. We compare the two approaches and show that while they prove to be equivalent in some contexts, window parity games offer a more tractable alternative when the time bound is given as a parameter (P-c. vs. PSPACE-c.). In particular, it provides a conservative approximation of parity games computable in polynomial time. Furthermore, we extend both approaches to the multi-dimension setting. We give the full picture for both types of games with regard to complexity and memory bounds.

1 Introduction

Games on graphs. Two-player games played on directed graphs constitute an important framework for the synthesis of a suitable controller for a reactive system faced to an uncontrollable environment [25]. In this setting, *vertices* of the graph represent states of the system and edges represent transitions between those states. We consider *turn-based two-player* games: each vertex either belongs to the system (the first player, denoted by $\mathscr{P}_1$) or the environment (the second player, denoted by $\mathscr{P}_2$). A game is played by moving an imaginary pebble from vertex to vertex according to existing transitions: the owner of a vertex decides where to move the pebble. The outcome of the game is an infinite sequence of vertices called *play*. The choices of both players depend on their respective *strategy* which can use an arbitrary amount of memory in full generality. In the classical setting, $\mathscr{P}_1$ tries to achieve an *objective* (describing a set of *winning plays*) while $\mathscr{P}_2$ tries to prevent him from succeeding: hence, our games are *zero-sum*. As all the objectives considered in this paper define Borel sets, Martin's theorem [24] guarantees determinacy.

Parity games. Two-player games with ω-regular objectives have been studied extensively in the literature. See for example [27, 17] for an introduction. A canonical way to represent games with ω-regular conditions is the class of *parity games*: vertices are assigned a non-negative integer *priority* (or *color*), and the objective asks that among the vertices that are seen infinitely often along a play, the minimal priority be even. Parity games have been under close scrutiny for a long time both due to their importance

*Q. Hautem is supported by a FRIA fellowship, M. Randour is an F.R.S.-FNRS Postdoctoral researcher.

D. Cantone and G. Delzanno (Eds.): Seventh Symposium on Games, Automata, Logics and Formal Verification (GandALF'16)
EPTCS 226, 2016, pp. 135–148, doi:10.4204/EPTCS.226.10

(e.g., they subsume modal μ-calculus model checking [14]) and their intriguing complexity: they belong to the class of problems in $\mathsf{UP} \cap \mathsf{coUP}$ [20] and despite many efforts (e.g., [30, 21, 22, 26]), whether they belong to P is still an open question. Furthermore, parity games enjoy memoryless determinacy [12, 30]. Multi-dimension parity games were studied in [9]: in such games, n-dimension vectors of priorities are associated to each vertex, and the objective is to satisfy the conjunction of all the one-dimension parity objectives. The complexity of solving those games is higher: deciding if $\mathscr{P}_1$ (resp. $\mathscr{P}_2$) has a winning strategy is coNP-complete (resp. NP-complete) and exponential memory is needed for $\mathscr{P}_1$ whereas $\mathscr{P}_2$ remains memoryless [11, 18].

Time bounds. In its classical formulation, the parity objective essentially requires that *for each odd priority seen infinitely often, a smaller even priority should also be seen infinitely often*. An odd priority can be seen as a stimulus that must be answered by seeing a smaller even priority. The parity objective has fundamental qualities. The simplicity of its definition abstracts timing issues like "how much time has elapsed between a stimulus and its answer" and is key to memoryless determinacy. It makes it robust to slight changes in the model which could impact more precise formulations (e.g., counting the number of steps between a stimulus and its answer critically depends on the granularity of the game graph).

Nonetheless, it has been recently argued that in a large number of practical applications, *timing does matter* (e.g., [8, 23, 5]). Indeed, in general it does not suffice to know that a "good behavior" will *eventually* happen, and one wants to ensure that it can actually be witnessed *within a time frame which is acceptable* with regard to the modeled reactive system. For example, consider a computer server having to grant requests to clients. A classical parity objective can encode that requests should eventually be granted. However, it is clear that in a desired controller, requests should not be placed on hold for an arbitrarily long time. In order to accomodate such requirements, various attempts to associate classical game objectives with time bounds have been recently studied. For example, *window mean-payoff* and *window total-payoff* games provide a framework to reason about quantitative games (e.g., modeling quantities such as energy consumption) with time bounds [5]. In the qualitative setting, *finitary parity* games [8, 6] and *parity games with costs* [16, 29] provide a similar framework for parity games.

Two approaches. While window games and finitary parity games (resp. parity games with costs) share the goal of allowing precise specification of time bounds, their inner mechanisms differ. The aim of our work is three-fold: (*i*) apply the **window mechanism to parity games**, (*ii*) provide a **thorough comparison** with the existing framework of finitary parity games and parity games with costs, (*iii*) **extend both approaches to the multi-dimension setting** (which was left unexplored up to now). Since all those related papers do not use a uniform terminology, we here use the following taxonomy for the two approaches.

- **Window parity (WP).** Intuitively, the *direct fixed WP* objective considers a window of size bounded by $\lambda \in \mathbb{N}_0$ (given as a parameter) sliding over an infinite play and declare this play winning if in all positions, the window is such that the minimal priority within it is even. For *direct bounded WP*, the size of the window is not fixed as a parameter but a play is winning if there exists a bound λ for which the condition holds. We also consider the *fixed WP* and *bounded WP* objectives which are essentially prefix-independent variants of the previous ones. All those objectives are based on the window mechanism introduced in [5] and our work presents the first implementation of this mechanism for parity games.
- **Parity-response (PR).** The *direct fixed PR* objective asks that along a play, any odd priority be followed by a smaller even priority in at most $\lambda \in \mathbb{N}_0$ (given as a parameter) steps. As for the WP setting, we also consider the *direct bounded PR* objective where a play is winning if there exists a bound λ such that the condition holds, along with the respective prefix-independent variants: the

<table>
<tr><th></th><th colspan="3">one-dimension</th><th colspan="3">multi-dimension</th></tr>
<tr><th></th><th>complexity</th><th>$\mathscr{P}_1$ mem.</th><th>$\mathscr{P}_2$ mem.</th><th>complexity</th><th>$\mathscr{P}_1$ mem.</th><th>$\mathscr{P}_2$ mem.</th></tr>
<tr><td>Fixed WP</td><td>P-c.</td><td colspan="2">polynomial</td><td rowspan="4">EXPTIME-c.</td><td colspan="2" rowspan="2">exponential</td></tr>
<tr><td>Fixed PR</td><td>PSPACE-c.</td><td>exponential</td><td>$\leq$ exponential</td></tr>
<tr><td>Bounded WP</td><td rowspan="2">P-c.</td><td rowspan="2">memoryless</td><td rowspan="2">infinite</td><td rowspan="2">exponential</td><td rowspan="2">infinite</td></tr>
<tr><td>Bounded PR</td></tr>
</table>

Table 1: Complexity of deciding the winner and memory required for winning strategies in window parity (WP) and parity-response (PR) games. All results hold for both the *prefix-independent* and the *direct* (Dir) variants of all the objectives, except for the memory of $\mathscr{P}_2$ in the direct bounded cases: in one-dimension games, linear memory is both sufficient and necessary (for both WP and PR) and in multi-dimension games, exponential memory is both sufficient and necessary. All bounds are tight unless specified by the $\leq$ symbol. New results are in bold.

fixed PR and the *bounded PR* objectives. The *bounded PR* objective was studied for one-dimension games (under the name *finitary parity*) in [8]: deciding the winner is in P and memoryless strategies suffice for $\mathscr{P}_1$ while $\mathscr{P}_2$ needs infinite memory. The *fixed PR* objective for one-dimension games was very recently proved to be PSPACE-complete, with exponential memory bounds for both players [29] (this work is presented in the more general context of *parity games with costs*). Our work provides the first study of the parity-response approach in multi-dimension games.

Our contributions. Given the number of variants studied, we give an overview of our results in Table 1. Our main contributions are as follows.

1. We prove that *bounded WP* and *bounded PR* objectives coincide, even in multi-dimension games (Proposition 3).
2. We establish that *bounded WP* (and thus *bounded PR*) games are P-hard in one-dimension (Theorem 6, P-membership follows from [8]) and that they are EXPTIME-complete in multi-dimension (Theorem 9). The EXPTIME-membership follows from a reduction to a variant of *request-response games* [28] presented in [8] under the name of *finitary Streett games*. The EXPTIME-hardness is proved via a reduction from the membership problem in alternating polynomial-space Turing machines.
3. We show that in multi-dimension *bounded WP* (and thus *bounded PR*) games, exponential memory is both sufficient and necessary for $\mathscr{P}_1$ while infinite memory is needed for $\mathscr{P}_2$ (Theorem 9).
4. We prove that one-dimension *fixed WP* games provide a **conservative approximation of parity games** (Proposition 3) **computable in polynomial time** (Theorem 8). This is in contrast to the PSPACE-completeness of *fixed PR* games [29] (actually, the proof in [29] is for a more general model but already holds for *fixed PR* games).
5. While *fixed PR* games are PSPACE-complete, we establish two polynomial-time algorithms (Theorem 7) to solve fixed-parameter sub-cases: (i) the bound λ is fixed, or (ii) the number of priorities is fixed.
6. In multi-dimension, we prove that both *fixed PR* (Theorem 11) and *fixed WP* (Theorem 12) games are EXPTIME-complete. Membership relies on different techniques and algorithms for each case while hardness is based on the same reduction as for the *bounded* variants.

7. In one-dimension games, we also establish that for *fixed WP*, polynomial memory is both sufficient and necessary for both players, whereas exponential memory is required for *fixed PR* [29]. In multi-dimension games, we prove that for both *fixed PR* and *fixed WP*, exponential memory is both sufficient and necessary for both players. The upper bounds follow from the EXPTIME algorithms mentioned above whereas the lower bounds in one-dimension are shown thanks to appropriate families of games and in multi-dimension are obtained through reduction from *generalized reachability games* [15].
8. We establish the existence of values of λ such that the *fixed* objectives become equivalent to the *bounded* ones, both in one-dimension (Theorem 6) and in multi-dimension (Theorem 9).

While all the aforementioned results are for the *prefix-independent* variants of our objectives, we also obtain closely related complexities and memory bounds for the *direct* ones (Table 1). We obtain our results using a variety of techniques, sometimes inspired by [8, 5]. Our focus is on giving the full picture for the two approaches toward including time bounds in parity games: window parity and parity-response. We sum up the key comparison points in the next paragraph.

Comparison. The *parity-response* and *window parity* approaches turn out to be equivalent in the *bounded* context, i.e., when the question is the existence of a bound $\lambda \in \mathbb{N}_0$ for which the corresponding *fixed* objective holds. Hence, the focus of the comparison is the *fixed* variants. Observe that those variants are of interest for applications where the time bound is part of the specification: parameter λ grants flexibility in the specification as it can be adjusted to specific requirements of the application. Let us review the complexities of the *fixed PR* and *fixed WP* objectives.

In one-dimension games, *fixed PR* is PSPACE-complete whereas *fixed WP provides a framework with similar flavor that enjoys increased tractability*: it is P-complete. Hence, *fixed WP does provide a polynomial-time conservative approximation of parity games* (Proposition 3). Interestingly, the *fixed WP* objective also permits to approximate the *fixed PR* one in both directions, and in polynomial time: we prove in Proposition 3 that the *fixed PR* objective for time bound λ can be framed by the *fixed WP* objective for two well-chosen values of the time bound λ' and λ''.

In multi-dimension, both *fixed PR* and *fixed WP* games are EXPTIME-complete. Still, while the *fixed PR* algorithm requires exponential time in *both* the number of dimensions *and* the number of priorities (which can be as large as the game graph), *solving the fixed WP case only requires exponential time in the number of dimensions*. This distinction may have *impact on practical applications* where, usually, the size of the model (the game graph) can be very large while the specification (hence the number of dimensions) is comparatively small. Note that for both objectives, the multi-dimension algorithms are *pseudo*-polynomial in the time bound λ, hence also exponential in the length of its binary encoding.

Finally, let us compare *window parity* games with *window mean-payoff (WMP)* games [5]. First, one could naturally wonder if WP games could be solved by encoding them into WMP games, following a reduction similar in spirit to the one developed by Jurdzinski for classical parity games [20]. This is indeed possible, but leads to increased complexities in comparison to the ad hoc analysis developed in this work. For example, multi-dimension *fixed WP* games would require exponential time in the number of priorities too. Second, observe that *fixed WP* games can be solved in polynomial time whatever the bound $\lambda \in \mathbb{N}_0$ whereas *fixed WMP* games require pseudo-polynomial time, i.e., also polynomial in the bound λ. Finally, multi-dimension *bounded WMP* games are known to be non-primitive-recursive-hard and their decidability is still open [5]. On the contrary, multi-dimension *bounded WP* games are EXPTIME-complete. This suggests that the colossal complexity of *bounded WMP* games is a result of the quantitative nature of mean-payoff mixed with windows, and not an inherent drawback of the window mechanism.

Other related work. In addition to the aforementioned articles, we mention two papers where logical formalisms dealing with time bounds are studied. In [23], Kupferman et al. introduced Prompt-LTL, which is strongly linked with the finitary conditions discussed above. In [1], Baier et al. also studied an extension of LTL that can express properties based on the window mechanism of [5]. The study of logical fragments corresponding to *WP* games is an interesting question left open for future work.

Outline. Section 2 presents the needed definitions and known results about classical objectives. Section 3 introduces the different objectives studied in this paper and establishes the links between them. Section 4 and Section 5 respectively present our results for one-dimension and multi-dimension games. Full proofs and detailed results, as well as additional discussion of related topics, can be found in the full version of this paper on arXiv [3].

2 Preliminaries

Game structures. We consider zero-sum turn-based games played by two players, $\mathscr{P}_1$ and $\mathscr{P}_2$, on a finite directed graph. A *game structure* is a tuple $G = (V_1, V_2, E)$ where (i) (V, E) is a finite directed graph, with $V = V_1 \cup V_2$ the set of vertices and $E \subseteq V \times V$ the set of edges such that for each $v \in V$, there exists $(v, v') \in E$ for some $v' \in V$ (no deadlock), (ii) (V_1, V_2) forms a partition of V such that V_i is the set of vertices controlled by player $\mathscr{P}_i$ with $i \in \{1, 2\}$.

A *play* of G is an infinite sequence of vertices $\pi = v_0 v_1 \ldots \in V^\omega$ such that $(v_k, v_{k+1}) \in E$ for all $k \in \mathbb{N}$. We denote by $\mathsf{Plays}(G)$ the set of plays in G. *Histories* of G are finite sequences $\rho = v_0 \ldots v_k \in V^*$ defined in the same way. Given a play $\pi = v_0 v_1 \ldots$, the history $v_k \ldots v_{k+l}$ is denoted by $\pi[k, k+l]$; in particular, $v_k = \pi[k]$. We also use notation $\pi[k, \infty]$ for the suffix $v_k v_{k+1} \ldots$ of π.

Strategies. A *strategy* σ_i for $\mathscr{P}_i$ is a function $\sigma_i \colon V^* V_i \to V$ assigning to each history $\rho v \in V^* V_i$ a vertex $v' = \sigma_i(\rho v)$ such that $(v, v') \in E$. It is *finite-memory* if it can be encoded by a deterministic *Moore machine*. The *size* of the strategy is the size of its Moore machine. It is *memoryless* if $\sigma_i(\rho v) = \sigma_i(\rho' v)$ for all histories $\rho v, \rho' v$ ending with the same vertex v, that is, if σ_i is a function $\sigma_i \colon V_i \to V$.

Given a strategy σ_i of $\mathscr{P}_i$, we say that a play $\pi = v_0 v_1 \ldots$ of G is *consistent* with σ_i if $v_{k+1} = \sigma_i(v_0 \ldots v_k)$ for all $k \in \mathbb{N}$ such that $v_k \in V_i$. Consistency is naturally extended to histories in a similar fashion. Given an *initial vertex* v_0, and a strategy σ_i of each player $\mathscr{P}_i$, we have a unique play consistent with both strategies. This play is called the *outcome* of the game and is denoted by $\mathsf{Out}(v_0, \sigma_1, \sigma_2)$.

Objectives and winning sets. Let $G = (V_1, V_2, E)$ be a game structure. An *objective for* $\mathscr{P}_1$ is a set of plays $\Omega \subseteq \mathsf{Plays}(G)$. A play π is *winning* for $\mathscr{P}_1$ if $\pi \in \Omega$, and losing otherwise (i.e., winning for $\mathscr{P}_2$). We thus consider *zero-sum* games such that the objective of player $\mathscr{P}_2$ is $\overline{\Omega} = \mathsf{Plays}(G) \backslash \Omega$. In the following, we always take the point of view of $\mathscr{P}_1$ by assuming that Ω is his objective, and we denote by (G, Ω) the corresponding *game*. Given an initial vertex v_0 of a game (G, Ω), a strategy σ_1 for $\mathscr{P}_1$ is *winning* from v_0 if $\mathsf{Out}(v_0, \sigma_1, \sigma_2) \in \Omega$ for all strategies σ_2 of $\mathscr{P}_2$. Vertex v_0 is also called *winning* for $\mathscr{P}_1$ and the *winning set* $\mathsf{Win}_1^G(\Omega)$ is the set of all his winning vertices. Similarly the winning vertices of $\mathscr{P}_2$ are those from which $\mathscr{P}_2$ can ensure to satisfy his objective $\overline{\Omega}$ against all strategies of $\mathscr{P}_1$, and $\mathsf{Win}_2^G(\overline{\Omega})$ is his winning set. If $\mathsf{Win}_1^G(\Omega) \cup \mathsf{Win}_2^G(\overline{\Omega}) = V$, we say that the game is *determined*. It is known that every turn-based game with a Borel objective is determined [24]. This in particular applies to the objectives studied in this paper.

Decision problem. Given a game (G, Ω) and an initial vertex v_0, we want to decide whether $\mathscr{P}_1$ has a winning strategy from v_0 for the objective Ω or not (in which case, $\mathscr{P}_2$ has one for $\overline{\Omega}$). We want to study the complexity class of this decision problem as well as the memory requirements of winning strategies of both players. In this paper, we focus on several variants of the parity objective and we consider two settings: the *one-dimension* case with one objective Ω and the *multi-dimension* case with the intersection of several objectives $\cap_{m=1}^{n} \Omega_m$.

Parity objective. Let G be a game structure. Let π be a play, we define $\mathsf{Inf}(\pi)$ as the set of vertices seen infinitely often in π. Formally, $\mathsf{Inf}(\pi) = \{v \in V \mid \forall k \geq 0, \exists l \geq k, \pi[l] = v\}$.

Given a *priority function* $p\colon V \to \{0,1,\ldots,d\}$ that maps every vertex to an integer priority where d is even and $d \leq |V|+1$ (w.l.o.g.), the *parity objective* $\mathsf{Parity}(p)$ asks that of the vertices that are visited infinitely often, the smallest priority be even. Formally, the parity objective is defined as $\mathsf{Parity}(p) = \{\pi \in \mathsf{Plays}(G) \mid \min_{v \in \mathsf{Inf}(\pi)} p(v) \text{ is even}\}$. As smallest even priorities have a specific role in parity objectives, we define a partial order $\preceq$ on priorities as follows. For $c, c' \in \{0,\ldots,d\}$, we have $c \preceq c'$ if and only if c is even and $c \leq c'$. In this case we say that c is $\preceq$-smaller than c'. State-of-the-art results about parity games were already discussed in Section 1.

Other useful objectives. We recall some useful results for several classical objectives. Let G be a game structure and $U \subseteq V$ be a set of vertices. A *reachability* objective $\mathsf{Reach}(U)$ asks to visit a vertex of U at least once, whereas a *safety* objective $\mathsf{Safe}(U)$ asks to visit no vertex of $V \setminus U$. Deciding the winner in reachability games and safety games is known to be P-complete with an algorithm in time $O(|V|+|E|)$, and memoryless winning strategies suffice for both objectives and both players [2, 17, 19]. A *Büchi* objective $\mathsf{Buchi}(U)$ asks to visit a vertex of U infinitely often, whereas a *co-Büchi* objective $\mathsf{CoBuchi}(U)$ asks to visit no vertex of $V \setminus U$ infinitely often. Deciding the winner in Büchi games and co-Büchi games is also P-complete with an algorithm in time $O(|V|^2)$, and memoryless strategies also suffice for both objectives and both players [7, 13, 17].

3 Adding time bounds to parity games

In this section, we introduce the two approaches discussed in this paper: *window parity* (WP) and *parity-response* (PR) games.

Window parity and parity-response objectives. The intuition for both approaches is as follows. The parity-response objective asks that every priority be followed by a $\preceq$-smaller priority in a bounded number of steps. In a window parity game, a *window* with a bounded size is sliding along the play, and one asks to find a $\preceq$-smallest[1] priority inside this window, and this for all positions along the play. We derive four variants for each of these objectives, according to whether the bound is given as a parameter or not (*fixed* or *bounded* variant), and whether the objective must be satisfied directly or eventually (*direct* or *undirect* variant). The *undirect* variants are thus prefix-independent.[2] Formally:

Definition 1. *Given a game structure $G = (V_1, V_2, E)$, a priority function $p\colon V \to \{0,1,\ldots,d\}$, and a* bound $\lambda \in \mathbb{N}_0$, *we define the eight following objectives:*

$$\begin{aligned}
\mathsf{DirFixPR}(\lambda, p) &= \{\pi \in \mathsf{Plays}(G) \mid \forall j \geq 0,\ \exists l < \lambda,\ p(\pi[j+l]) \preceq p(\pi[j])\},\\
\mathsf{DirFixWP}(\lambda, p) &= \{\pi \in \mathsf{Plays}(G) \mid \forall j \geq 0,\ \exists l < \lambda,\ \forall k \leq l,\ p(\pi[j+l]) \preceq p(\pi[j+k])\},
\end{aligned}$$

and given $\mathsf{X} \in \{\mathsf{PR}, \mathsf{WP}\}$,

$$\begin{aligned}
\mathsf{FixX}(\lambda, p) &= \{\pi \in \mathsf{Plays}(G) \mid \exists i \geq 0,\ \pi[i,\infty] \in \mathsf{DirFixX}(\lambda, p)\},\\
\mathsf{DirBndX}(p) &= \{\pi \in \mathsf{Plays}(G) \mid \exists \lambda \in \mathbb{N}_0,\ \pi \in \mathsf{DirFixX}(\lambda, p)\},\\
\mathsf{BndX}(p) &= \{\pi \in \mathsf{Plays}(G) \mid \exists i \geq 0,\ \pi[i,\infty] \in \mathsf{DirBndX}(p)\}.
\end{aligned}$$

[1] Notice the difference: smallest vs. smaller.

[2] An objective Ω is *prefix-independent* if for any play $\pi = \rho\pi'$, it holds that $\pi \in \Omega \iff \pi' \in \Omega$.

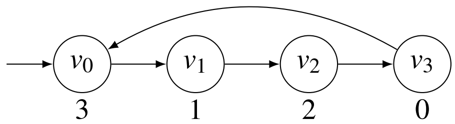

Figure 1: A simple example of one-player-game: all vertices belong to $\mathscr{P}_1$.

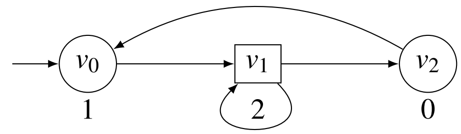

Figure 2: A game where $\mathscr{P}_1$ wins for parity but loses for all variants of objectives WP and PR.

Thus, in the direct fixed parity-response objective $\mathsf{DirFixPR}(\lambda, p)$, for all positions $j \geq 0$, the priority $p(\pi[j])$ must be followed by a $\preceq$-smaller priority $p(\pi[j+l])$ within at most $\lambda - 1$ steps. The undirect fixed variant $\mathsf{FixPR}(\lambda, p)$ asks this objective to be satisfied eventually (i.e., for all positions $j \geq i$, for some i). The direct bounded variant $\mathsf{DirBndPR}(p)$ (resp. the undirect bounded variant $\mathsf{BndPR}(p)$) asks for the existence of a bound λ for which $\mathsf{DirFixPR}(\lambda, p)$ (resp. $\mathsf{FixPR}(\lambda, p)$) is satisfied.

For window parity objectives, we call λ the *window size*. Given a play $\pi = v_0v_1\ldots$, a λ-*window at position* j is a window of size λ placed along π from position j to $j+\lambda-1$. The direct fixed window parity objective $\mathsf{DirFixWP}(\lambda, p)$ asks that for all $j \geq 0$, inside the λ-window at position j, one can find a priority $p(\pi[j+l])$ with $l \leq \lambda - 1$ that is the $\preceq$-smallest one in $\pi[j, j+l]$. When the window size λ is clear from the context, we drop the prefix λ and simply talk about windows instead of λ-windows.

Example 2. We illustrate the definitions on a simple example where all vertices belong to $\mathscr{P}_1$ (Figure 1). In this example and in the sequel, the priority $p(v)$ is always put under vertex v, and circle (resp. square) vertices all belong to $\mathscr{P}_1$ (resp. $\mathscr{P}_2$). In the game of Figure 1, there is a unique play from the initial vertex v_0 equal to $\pi = (v_0v_1v_2v_3)^\omega$. On the one hand, we have that $\pi \in \mathsf{DirFixPR}(\lambda, p)$ for $\lambda = 3$. Indeed, the odd priority 3 (resp. 1) is followed by the even priority 2 (resp. 0) in exactly $\lambda - 1 = 2$ steps, whereas the even priority 2 (resp. 0) is "followed" by itself in 0 steps. Similarly, π also belongs to the three variants $\mathsf{FixPR}(\lambda, p)$, $\mathsf{DirBndPR}(p)$ and $\mathsf{BndPR}(p)$. On the other hand, $\pi \notin \mathsf{DirFixWP}(3, p)$. Indeed, in the 3-window at position 0, there is no $l \in \{0,1,2\}$ such that $p(v_l)$ is the $\preceq$-smallest priority in $\pi[0,l]$ because $p(v_0)$ and $p(v_1)$ are odd, and $p(v_2)$ is even but $p(v_2) = 2 \not\preceq 1 = p(v_1)$. However, one can check that $\pi \in \mathsf{DirFixWP}(4, p)$, and it also belongs to $\mathsf{FixWP}(4, p)$, $\mathsf{DirBndWP}(p)$ and $\mathsf{BndWP}(p)$. $\triangleleft$

Relationship between objectives. We now detail the inclusions and equalities between the various objectives introduced in Definition 1 as well as with the parity objective.

Proposition 3. *Let $G = (V_1, V_2, E)$ be a game structure and p be a priority function. Let $\lambda \in \mathbb{N}_0$.*

1. *For all* $\mathsf{X} \in \{\mathsf{FixPR}(\lambda, p), \mathsf{FixWP}(\lambda, p), \mathsf{BndPR}(p), \mathsf{BndWP}(p)\}$, $\mathsf{DirX} \subseteq \mathsf{X}$.
2. *For all* $\mathsf{X} \in \{\mathsf{PR}, \mathsf{WP}\}$, $(\mathsf{Dir})\mathsf{FixX}(\lambda, p) \subseteq (\mathsf{Dir})\mathsf{BndX}(p)$.
3. *For all $\lambda' > \lambda$, for all* $\mathsf{X} \in \{\mathsf{FixPR}, \mathsf{FixWP}\}$, $(\mathsf{Dir})\mathsf{X}(\lambda, p) \subseteq (\mathsf{Dir})\mathsf{X}(\lambda', p)$.
4. $(\mathsf{Dir})\mathsf{FixWP}(\lambda, p) \subseteq (\mathsf{Dir})\mathsf{FixPR}(\lambda, p)$.
5. $(\mathsf{Dir})\mathsf{FixPR}(\lambda, p) \subseteq (\mathsf{Dir})\mathsf{FixWP}(\frac{d}{2} \cdot \lambda, p)$.
6. $(\mathsf{Dir})\mathsf{BndPR}(p) = (\mathsf{Dir})\mathsf{BndWP}(p)$.
7. $(\mathsf{Dir})\mathsf{BndWP}(p) \subseteq \mathsf{Parity}(p)$.

We give here an intuitive explanation of Item 5, as it is the most interesting one technically. Assume we have a play $\pi \in \mathsf{DirFixPR}(\lambda, p)$, like the one depicted in Figure 3. Since it satisfies objective $\mathsf{DirFixPR}(\lambda, p)$, we know that each odd priority is followed by a smaller even priority in at most $(\lambda - 1)$ steps. We argue that it belongs to $\mathsf{DirFixWP}(\frac{d}{2} \cdot \lambda, p)$, i.e., that for any position $j \geq 0$, the $(\frac{d}{2} \cdot \lambda)$-window at position j sees a $\preceq$-smallest priority in some position $j+l$ with $l < \frac{d}{2} \cdot \lambda$. The key idea is depicted

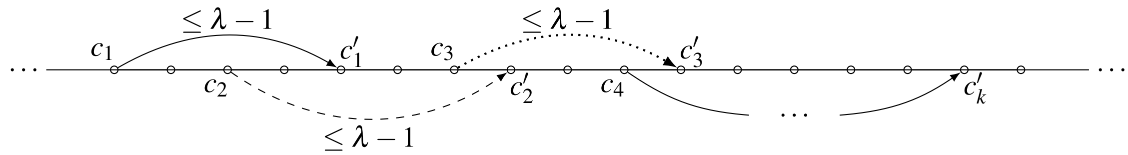

Figure 3: Illustration of inclusion $(\mathsf{Dir})\mathsf{FixPR}(\lambda, p) \subseteq (\mathsf{Dir})\mathsf{FixWP}(\frac{d}{2} \cdot \lambda, p)$.

in Figure 3. Let c_1 be an odd priority. It must be followed by a $\preceq$-smaller priority c_1' in at most $(\lambda - 1)$ steps. If c_1' is the minimal priority encountered from c_1 to c_1', then we are done. Assume it is not, then there exists c_2 between c_1 and c_1' such that c_2 is odd and $c_1' \not\preceq c_2$. But again, c_2 must be followed by $c_2' \preceq c_2$ in at most $(\lambda - 1)$ steps. Repeating this argument, we obtain that c_1 is followed by a priority c_k' in strictly less than $\frac{d}{2} \cdot \lambda$ steps[3] (as there are $\frac{d}{2}$ odd priorities and each of them is answered in $(\lambda - 1)$ steps) such that c_k' is even and smaller than all priorities encountered from c_1 to c_k'. Therefore, c_k' is the $\preceq$-smallest priority in $\pi[j, j+l]$ for some $l < \frac{d}{2} \cdot \lambda$. Since this argument can be repeated for any position $j \geq 0$, we obtain that the play satisfies $\mathsf{DirFixWP}(\frac{d}{2} \cdot \lambda, p)$ as claimed.

From the inclusions $\Omega \subseteq \Omega'$ of Proposition 3, we immediately derive the inclusions $\mathsf{Win}_1^G(\Omega) \subseteq \mathsf{Win}_1^G(\Omega')$. It yields two interesting observations mentioned in Section 1. Notice that the inclusions of Proposition 3 are strict in general. This is also the case when one replaces the objectives by the winning sets of $\mathscr{P}_1$ for these objectives. We briefly sketch the most interesting case here.

Example 4. Consider the game in Figure 2. The initial vertex v_0 is winning for the parity objective but is losing for all variants of objectives WP and PR: $\mathscr{P}_2$ has the possibility to use the self-loop on v_1 to delay for an arbitrarily long time the visit of the $\preceq$-smaller priority 0 after seeing priority 1, and can do so repeatedly using the other loop, thus defeating both direct and undirect variants of the objectives, as $\mathscr{P}_1$ is never able to ensure a bound on the window size needed to see a $\preceq$-smallest priority. To win for the undirect bounded variants, $\mathscr{P}_2$ must use *infinite memory* and play in rounds, increasing the time spent looping in v_1 at each round, thus preventing the existence of a bound. $\triangleleft$

We close this section by establishing that for the sub-case of games with priorities in $\{0, 1, 2\}$, WP and PR objectives coincide.

Lemma 5. *Let G be a game structure and $p\colon V \to \{0, 1, 2\}$ be a priority function. For all $\lambda \in \mathbb{N}_0$, we have that* $(\mathsf{Dir})\mathsf{FixPR}(\lambda, p) = (\mathsf{Dir})\mathsf{FixWP}(\lambda, p)$.

4 One-dimension games

We begin our study of WP and PR objectives with one-dimension games: in this setting, there is a unique priority function p and the objective Ω is a single objective $(\mathsf{Dir})\mathsf{FixX}$ or $(\mathsf{Dir})\mathsf{BndX}$ for $\mathsf{X} \in \{\mathsf{PR}, \mathsf{WP}\}$.

Bounded variants. Recall that by Proposition 3, the bounded variants are equivalent. Furthermore, it is already known that games with objective $(\mathsf{Dir})\mathsf{BndPR}(p)$ are solvable in polynomial time [8]. The next theorem sums up the complexity landscape for bounded variants and enrich it by proving P-hardness for the associated decision problems. The result is obtained via a reduction from reachability games. In terms of memory requirements, $\mathscr{P}_1$ can play without memory whereas Example 4 already illustrated that $\mathscr{P}_2$ requires infinite memory in general. The linear memory bound for $\mathscr{P}_2$ and the direct variant was established in [16].

[3] Actually, $\frac{d}{2} \cdot (\lambda - 1) + 1$ but we use the simpler bound $\frac{d}{2} \cdot \lambda$ from now on for the sake of readability.

Theorem 6. *Let $G = (V_1, V_2, E)$ be a game structure, v_0 be an initial vertex, p be a priority function, and Ω be the objective* $\mathsf{DirBndPR}(p)$ *or* $\mathsf{DirBndWP}(p)$ *(resp.* $\mathsf{BndPR}(p)$ *or* $\mathsf{BndWP}(p)$*).*

1. *Deciding the winner in (G, Ω) from v_0 is* P*-complete with an algorithm in $\mathcal{O}(|V| \cdot |E|)$ (respectively $\mathcal{O}(|V|^2 \cdot |E|)$) time, memoryless strategies are sufficient for $\mathcal{P}_1$, and linear-memory strategies are necessary and sufficient for $\mathcal{P}_2$ (respectively infinite memory is necessary for $\mathcal{P}_2$).*
2. *$\forall \lambda \geq |V|$, $\forall \lambda' \geq \frac{d}{2} \cdot |V|$, the winning sets for the objectives* $\mathsf{BndPR}(p)$, $\mathsf{FixPR}(\lambda, p)$, $\mathsf{BndWP}(p)$, *and* $\mathsf{FixWP}(\lambda', p)$ *are all equal. The same equalities hold for the direct variants* (Dir).

The fixed variants are more interesting: the PR and WP approaches yield different results in this setting. We start with the PR one, for which we provide two polynomial-time algorithms for fixed-parameter sub-cases, hence significantly reducing the complexity of the problem (which is PSPACE-complete in the general case).

Fixed parity-response objectives. Deciding the winner in (Dir)FixPR games was very recently proved to be PSPACE-complete [29]. As mentioned in Section 1, the proof was actually provided for a more general model, but already holds for both FixPR and DirFixPR games. Observe that the PSPACE-hardness only holds for time bounds $\lambda < |V|$ since we know by Theorem 6, Item 2, that for larger values, the objectives are equivalent to the bounded variants, hence the corresponding decision problems lie in P. We focus on the case $\lambda < |V|$: we show in the next theorem that when we fix either the largest priority d or the bound λ, the complexity collapses to P. We briefly sketch the corresponding algorithms here.

First, consider the case where d is fixed. We reduce the $\mathsf{FixPR}(\lambda, p)$ (resp. $\mathsf{DirFixPR}(\lambda, p)$) game to a co-Büchi (resp. safety) game on an extended graph where we keep track of additional information in the vertices. Namely, we keep a vector that represents, for each odd priority c, the number of steps since seeing c without seeing any $\preceq$-smaller priority iboundsn the meantime. When this number reaches λ for any odd priority, we visit a special "bad vertex" and then reset the counters in the vector and resume the game. Essentially, winning for $\mathsf{FixPR}(\lambda, p)$ (resp. $\mathsf{DirFixPR}(\lambda, p)$) boils down to eventually (resp. completely) avoiding those bad vertices, hence to a co-Büchi (resp. safety) game. This extended game has size $\mathcal{O}(|V| \cdot \lambda^{\frac{d}{2}})$ and can be solved in polynomial time since $\lambda < |V|$ and d is fixed.

Second, consider the case where λ is fixed. We also reduce the $\mathsf{FixPR}(\lambda, p)$ (resp. $\mathsf{DirFixPR}(\lambda, p)$) game to a co-Büchi (resp. safety) game, but with a different extended graph. Specifically, we here keep track of the last λ vertices seen in the original game, and we want to avoid vertices of the extended graph that correspond to histories where an odd priority c is not followed by a priority $c' \preceq c$ within $(\lambda - 1)$ steps. Again, this can be expressed as either a co-Büchi or a safety objective depending on whether we are interested in the undirect or the direct variant respectively. The extended game has size $\mathcal{O}(|V|^{\lambda})$ hence can be solved in polynomial time since λ is fixed.

Theorem 7. *Let $G = (V_1, V_2, E)$ be a game structure, v_0 be an initial vertex, $p \colon V \to \{0, \dots, d\}$ be a priority function, and Ω be the objective* $\mathsf{DirFixPR}(\lambda, p)$ *(resp.* $\mathsf{FixPR}(\lambda, p)$*) for some $\lambda < |V|$. If either d is fixed or λ is fixed, deciding the winner in (G, Ω) from v_0 is in* P. *More precisely, if d is fixed, deciding the winner can be done in $\mathcal{O}((|V| + |E|) \cdot \lambda^{\frac{d}{2}})$ (resp. $\mathcal{O}(|V|^2 \cdot \lambda^d)$) time, and if λ is fixed, deciding the winner can be done in $\mathcal{O}((|V| + |E|) \cdot |V|^{\lambda - 1})$ (resp. $\mathcal{O}(|V|^{2\lambda})$). In both cases, polynomial-memory strategies are sufficient for both players, and memory is necessary even in one-player games.*

Fixed window parity objectives. Whereas (Dir)FixPR games are PSPACE-complete, we now establish that (Dir)FixWP games are P-complete. Observe that if $\lambda \geq \frac{d}{2} \cdot |V|$, the problem boils down to solving the bounded variant thanks to Theorem 6. Hence, we focus on the case where $\lambda < \frac{d}{2} \cdot |V|$.

Our algorithm is inspired by the approach developed for *window mean-payoff games* in [5]. It can be sketched as follows. As for the fixed-parameter algorithms for (Dir)FixPR games presented in Theorem 7, we want to reduce the FixWP and DirFixWP games to co-Büchi and safety games respectively, where $\mathscr{P}_1$ wants to avoid "bad vertices" representing a violation of the condition at stake. Here, such a violation represents a λ-*bad window*, i.e., a window for which no even minimum priority is found before λ steps. Detecting such λ-bad windows can be achieved by considering an extended game structure where we encode additional information for the minimum priority of the current window and the number of steps in this window. A "bad vertex" is visited whenever we reach the end of a λ-window with an odd minimum priority. If an even minimum is found, it is also a minimum for the windows at intermediate positions and the step counter is reset. The extended game has size $\mathcal{O}(|V| \cdot d \cdot \lambda)$, hence polynomial size since $\lambda < \frac{d}{2} \cdot |V|$. Therefore, we can solve it in polynomial time. This is in contrast to window mean-payoff games where the fixed variant requires *pseudo*-polynomial time in general [5].

Upper bounds on the memory are obtained by construction of our reduction and we prove polynomial lower bounds in the extended version of this paper [3].

Theorem 8. *Let $G = (V_1, V_2, E)$ be a game structure, v_0 be an initial vertex, p be a priority function, and Ω be the objective $\mathsf{DirFixWP}(\lambda, p)$ (resp. $\mathsf{FixWP}(\lambda, p)$) for some $\lambda < |V|$. Then deciding the winner in (G, Ω) from v_0 is P-complete with an algorithm in $\mathcal{O}((|V| + |E|) \cdot d \cdot \lambda)$ (resp. $\mathcal{O}(|V|^2 \cdot d^2 \cdot \lambda^2)$) time. Polynomial-memory strategies are both sufficient and necessary for both players.*

5 Multi-dimension games

We now consider multi-dimension games: in this setting, there are n priority functions $p_1, \ldots, p_n$ and the objective Ω is the *conjunction* of *identical* objectives Ω_m for each "dimension" (i.e., priority function).

5.1 Bounded variants

Recall that Proposition 3 established the equality of objectives $(\mathsf{Dir})\mathsf{BndWP}(p)$ and $(\mathsf{Dir})\mathsf{BndPR}(p)$ in the one-dimension setting. This equality trivially carries over to the multi-dimension setting, i.e., we have that $\cap_{m=1}^{n}(\mathsf{Dir})\mathsf{BndWP}(p_m) = \cap_{m=1}^{n}(\mathsf{Dir})\mathsf{BndPR}(p_m)$ since the individual objectives (one per priority function) are equal. Hence, it suffices to obtain our results for either WP or PR objectives.

Overview. The next theorem presents an overview of our results. For a comparison of those results with related models, see Section 1. We sketch the key points to prove the theorem in the following paragraphs.

Theorem 9. *Let $G = (V_1, V_2, E)$ be a game structure, v_0 be an initial vertex, $p_1, \ldots, p_n$ be n priority functions, and Ω be the objective $\cap_{m=1}^{n}\mathsf{DirBndPR}(p_m)$ or $\cap_{m=1}^{n}\mathsf{DirBndWP}(p_m)$ (resp. $\cap_{m=1}^{n}\mathsf{BndPR}(p_m)$ or $\cap_{m=1}^{n}\mathsf{BndWP}(p_m)$). Let $b = |V| \cdot 2^{n \cdot \frac{d}{2}} \cdot n \cdot \frac{d}{2}$.*

1. *Deciding the winner in (G, Ω) from v_0 is $\mathsf{EXPTIME}$-complete with an algorithm in $\mathcal{O}(b^2)$ (resp. $\mathcal{O}(|V| \cdot b^2)$) time, and exponential-memory strategies are necessary and sufficient for both players (resp. for $\mathscr{P}_1$ and infinite-memory is necessary for $\mathscr{P}_2$).*

2. *$\forall \lambda \geq b$, $\forall \lambda' \geq b \cdot \frac{d}{2}$, the winning sets for the following objectives are all equal: $\cap_{m=1}^{n}\mathsf{BndPR}(p_m)$, $\cap_{m=1}^{n}\mathsf{FixPR}(\lambda, p_m)$, $\cap_{m=1}^{n}\mathsf{BndWP}(p_m)$, and $\cap_{m=1}^{n}\mathsf{FixWP}(\lambda', p_m)$. The same equalities hold for the direct variants* (Dir).

Exponential-time algorithm and upper bounds on memory. To prove $\mathsf{EXPTIME}$-membership, we introduce related games from the literature. First, *request-response games* [28, 8]. Consider r sets of vertices $Rq_1, \ldots, Rq_r$ representing requests and r sets of vertices $Rp_1, \ldots, Rp_r$ representing corresponding responses ($Rq_i, Rp_i \subseteq V$ for all i). The *request-response* objective, denoted by $\mathsf{RR}((Rq_i, Rp_i)_{i=1}^{r})$, requires that *for all* i, whenever a vertex of Rq_i is visited, then, later on, a vertex of Rp_i is also visited.[4] Observe that by definition, this objective is *direct*, i.e., the condition must hold from the start, not only eventually. Several variants of these games have been studied in the literature, under various names. Those variants include *direct bounded request-response* games: the objective $\mathsf{DirBndRR}((Rq_i,Rp_i)_{i=1}^r)$ asks that there exist a bound $b \in \mathbb{N}_0$ such that if a request is visited, then the corresponding response is visited within b steps. Its undirect variant BndRR have also been considered. The results of interest for us are (i) the EXPTIME-membership of all those variants, (ii) that exponential memory suffices for both players in all variants except for $\mathscr{P}_2$ and BndRR where infinite memory is necessary, and (iii) that whatever the game G, if $\mathscr{P}_1$ can win for any of those objectives, he can ensure that eventually all requests are answered in at most $b = |V| \cdot 2^r \cdot r$ steps [28, 8].

We establish a polynomial-time reduction from multi-dimension DirBndWP and BndWP games (or equivalently, DirBndPR and BndPR games) to DirBndRR and BndRR games respectively. The crux is to model each odd priority in each dimension as a request whose corresponding response is the occurrence of a $\preceq$-smaller priority in the same dimension. Since we have $\frac{d}{2}$ odd priorities and n dimensions, we need $n \cdot \frac{d}{2}$ pairs of requests and responses. We thus obtain an exponential-time algorithm for multi-dimension (Dir)BndWP and (Dir)BndPR games, along with exponential upper bounds on memory in all cases except the one of $\mathscr{P}_2$ in BndPR and BndWP games.

Equalities between objectives. The key ingredient for the last item of Theorem 9 is the aforementioned bound given in [8] for (Dir)BndRR games, and by extension, for (Dir)BndPR and (Dir)BndWP games thanks to our reduction. The rest follows the same lines as in the one-dimension case, i.e., it builds upon the inclusions and equalities presented in Proposition 3.

Lower bound on complexity. To prove the EXPTIME-hardness of objective (Dir)BndWP (and equivalently, of objective (Dir)BndPR), we establish a reduction from the *membership problem for alternating polynomial-space Turing machines (APTMs)* [4]. Our proof is adapted from the reduction presented in [5, Lemma 23] in the related context of window mean-payoff games. Since technical details are similar to [5, Lemma 23], we only include a high-level sketch of the reduction in the full version of this paper [3]. The main change is the way we deal with windows: whereas weights were used for window mean-payoff games, we need here to emulate the same actions with adapted priorities. Interestingly, our proof also shows EXPTIME-hardness of the fixed variants, (Dir)FixWP and (Dir)FixPR. Furthermore, the hardness already holds with only three priorities ($d = 2$).

Lower bounds on memory. The last missing pieces to the proof of Theorem 9 are the exponential lower bounds on memory. Recall that for $\mathscr{P}_2$ in *undirect* bounded WP or PR games, we already proved that infinite memory is necessary in Example 4. To cover all remaining cases and establish exponential lower bounds matching the upper bounds obtained above, we prove a polynomial-time reduction from *generalized reachability games* [15] to multi-dimension (Dir)BndWP and (Dir)BndPR games. Given $U_1,\dots,U_n$ a family of n subsets of V, a generalized reachability objective $\mathsf{GenReach}(U_1,...,U_n) = \cap_{m=1}^n \mathsf{Reach}(U_m)$ asks to visit a vertex of U_m at least once, for each $m \in \{1,\dots,n\}$. Since GenReach games are known to require exponential memory for both players [15], the reduction yields the desired lower bounds. A similar reduction is presented for window mean-payoff games in [5]. Interestingly, the same technique also works for multi-dimension (Dir)FixWP and (Dir)FixPR games.

Let us sketch the reduction from GenReach to multi-dimension FixWP games (the other cases are similar). Intuitively, if the generalized reachability objective asks to visits n different target sets, we will use n dimensions. We create a modified version of the game structure such that, at the start of the

[4]Note that a single response Rp_i suffices to answer all pending requests Rq_i, in the same spirit as for priorities in the *parity-response* objective.

game, we see priority 1 in all dimensions, and such that the only way to see priority 0 in dimension $m \in \{1, \ldots, n\}$ is to visit the m-th target set. We also modify the game by giving $\mathscr{P}_2$ the possibility to force seeing 0 in all dimensions and restart the game by seeing 1's again: this is necessary to ensure that the prefix-independence of objective FixWP cannot help $\mathscr{P}_1$ to win without visiting all target sets. Finally, we use the fact that if $\mathscr{P}_1$ has a winning strategy in a GenReach game with n targets, then he has one that wins in strictly less than $n \cdot |V|$ steps (i.e., edges), to define an appropriate window size $\lambda = 2 \cdot n \cdot |V|$ for which the reduction to objective $\cap_{m=1}^{n} \mathsf{FixWP}(\lambda, p_m)$ on our modified game structure holds. As in the reduction for EXPTIME-hardness, we only need three priorities here ($d = 2$).

For the reader's interest, we complement this reduction with an example illustrating the need for exponential memory for $\mathscr{P}_1$ in FixWP games (it also works for the other objectives).

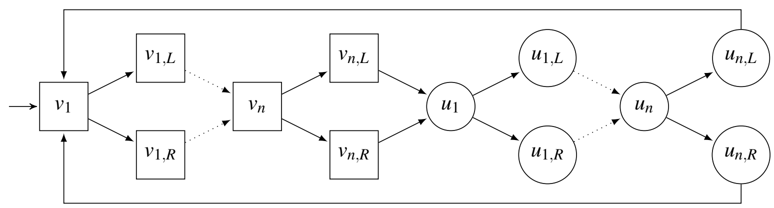

Figure 4: Family of multi-dimension games requiring exponential memory for $\mathscr{P}_1$ for objective FixWP with $\lambda = 3n$.

Example 10. Consider the family of game structures depicted in Figure 4. This family is parameterized by $n \in \mathbb{N}_0$ and is inspired by a similar one proposed in [10] for a different context (i.e., energy games). For each of these games structures, the number of vertices is linear in n ($|V| = 6n$) and we define $2n$ priority functions in the following way: for all $i \in \{1, \ldots, n\}$ and for all $m \in \{1, \ldots, 2n\}$, $p_m(v_i) = p_m(u_i) = 2$,

$$p_m(v_{i,L}) = \begin{cases} 1 & \text{if } m = 2i - 1 \\ 2 & \text{otherwise} \end{cases}, \qquad p_m(v_{i,R}) = \begin{cases} 1 & \text{if } m = 2i \\ 2 & \text{otherwise} \end{cases},$$

$$p_m(u_{i,L}) = \begin{cases} 0 & \text{if } m = 2i - 1 \\ 2 & \text{otherwise} \end{cases}, \qquad p_m(u_{i,R}) = \begin{cases} 0 & \text{if } m = 2i \\ 2 & \text{otherwise} \end{cases}.$$

Let $\Omega = \cap_{m=1}^{2n} \mathsf{FixWP}(3n, p_m)$ be the objective of $\mathscr{P}_1$. In order to prevent $(3n)$-bad windows, $\mathscr{P}_1$ has to choose $u_{i,L}$ (resp. $u_{i,R}$) whenever $\mathscr{P}_2$ chooses $v_{i,L}$ (resp. $v_{i,R}$). Hence in order to prevent outcomes with infinitely-many $(3n)$-bad windows, $\mathscr{P}_1$ must be able to record 2^n different histories from v_1 to u_1. This obviously requires exponential memory in n, hence in the size of the game. ◁

5.2 Fixed variants

As in one-dimension, some differences arise for the fixed variants. See Section 1 for a comparison.

Parity-response objectives. To establish an exponential-time algorithm for multi-dimension DirFixPR (resp. FixPR) games, we reduce them to safety (resp. co-Büchi) games on an exponentially-larger game structure. Our reduction is in the same spirit[5] as the one for Theorem 7 for the case d fixed in one-dimension. That is, the extended structure encodes for each odd priority in each dimension, the number of

[5]Note that the other algorithm suggested in Theorem 7 and exponential in λ is not interesting here, since λ can be exponential before the fixed variant becomes equivalent to the bounded one (Theorem 9), hence this algorithm would take *doubly*-exponential time.

steps since seeing the odd priority without seeing a $\preceq$-smaller priority in the meantime. The complexity and memory lower bounds follow from the reductions sketched for the bounded variants.

Theorem 11. *Let $G = (V_1, V_2, E)$ be a game structure, v_0 be an initial vertex, $p_1, \ldots, p_n$ be n priority functions, and Ω be the objective $\bigcap_{m=1}^{n} \mathsf{DirFixPR}(\lambda, p_m)$ (resp. $\bigcap_{m=1}^{n} \mathsf{FixPR}(\lambda, p_m)$) for $\lambda \in \mathbb{N}_0$. Deciding the winner in (G, Ω) from v_0 is* EXPTIME*-complete with an algorithm in $\mathcal{O}((|V| + |E|) \cdot \lambda^{\frac{d}{2} \cdot n})$ (resp. $\mathcal{O}(|V|^2 \cdot \lambda^{d \cdot n})$) time. Exponential memory is both sufficient and necessary for both players.*

Window parity objectives. To conclude our study of PR and WP objectives, it remains to establish an exponential-time algorithm for multi-dimension DirFixWP (resp. FixWP) games. Again, we reduce those games to safety (resp. co-Büchi) games on an exponentially-larger game structure. Our reduction is here based on the one used in the one-dimension setting (Theorem 8). That is, the extended structure encodes, for each dimension, the minimum priority of the current window, and the number of steps in that window. The complexity and memory lower bounds follow from the reductions sketched for the bounded variants.

Theorem 12. *Let $G = (V_1, V_2, E)$ be a game structure, v_0 be an initial vertex, $p_1, \ldots, p_n$ be n priority functions, and Ω be the objective $\bigcap_{m=1}^{n} \mathsf{DirFixWP}(\lambda, p_m)$ (resp. $\bigcap_{m=1}^{n} \mathsf{FixWP}(\lambda, p_m)$) for $\lambda \in \mathbb{N}_0$. Deciding the winner in (G, Ω) from v_0 is* EXPTIME*-complete with an algorithm in $\mathcal{O}((|V| + |E|) \cdot (d \cdot \lambda)^n)$ (resp. $\mathcal{O}(|V|^2 \cdot (d \cdot \lambda)^{2 \cdot n})$) time. Exponential memory is both sufficient and necessary for both players.*

Acknowledgments. We express our gratitude to Jean-François Raskin (Université libre de Bruxelles, Belgium) and Martin Zimmermann (Saarland University, Germany) for insightful discussions.

References

[1] C. Baier, J. Klein, S. Klüppelholz & S. Wunderlich (2014): *Weight monitoring with linear temporal logic: complexity and decidability*. In: Proc. of CSL-LICS, ACM, pp. 11:1–11:10, doi:10.1145/2603088.2603162.

[2] C. Beeri (1980): *On the Membership Problem for Functional and Multivalued Dependencies in Relational Databases*. ACM Trans. Database Syst. 5(3), pp. 241–259, doi:10.1145/320613.320614.

[3] V. Bruyère, Q. Hautem & M. Randour (2016): *Window parity games: an alternative approach toward parity games with time bounds*. CoRR abs/1606.01831. Available at `http://arxiv.org/abs/1606.01831`.

[4] A.K. Chandra, D. Kozen & L.J. Stockmeyer (1981): *Alternation*. J. ACM 28(1), pp. 114–133, doi:10.1145/322234.322243.

[5] K. Chatterjee, L. Doyen, M. Randour & J.-F. Raskin (2015): *Looking at mean-payoff and total-payoff through windows*. Information and Computation 242, pp. 25–52, doi:10.1016/j.ic.2015.03.010.

[6] K. Chatterjee & N. Fijalkow (2013): *Infinite-state games with finitary conditions*. In: Proc. of CSL, LIPIcs 23, Schloss Dagstuhl - LZI, pp. 181–196, doi:10.4230/LIPIcs.CSL.2013.181.

[7] K. Chatterjee & M. Henzinger (2014): *Efficient and Dynamic Algorithms for Alternating Büchi Games and Maximal End-Component Decomposition*. J. ACM 61(3), pp. 15:1–15:40, doi:10.1145/2597631.

[8] K. Chatterjee, T.A. Henzinger & F. Horn (2009): *Finitary winning in ω-regular games*. ACM Trans. Comput. Log. 11(1), doi:10.1145/1614431.1614432.

[9] K. Chatterjee, T.A. Henzinger & N. Piterman (2007): *Generalized Parity Games*. In: Proc. of FOSSACS, LNCS 4423, Springer, pp. 153–167, doi:10.1007/978-3-540-71389-0_12.

[10] K. Chatterjee, M. Randour & J.-F. Raskin (2014): *Strategy synthesis for multi-dimensional quantitative objectives*. Acta Informatica 51(3-4), pp. 129–163, doi:10.1007/s00236-013-0182-6.

[11] S. Dziembowski, M. Jurdzinski & I. Walukiewicz (1997): *How Much Memory is Needed to Win Infinite Games?* In: *Proc. of LICS*, IEEE Computer Society, pp. 99–110, doi:10.1109/LICS.1997.614939.

[12] E.A. Emerson & C.S. Jutla (1988): *The Complexity of Tree Automata and Logics of Programs (Extended Abstract)*. In: *Proc. of FOCS*, pp. 328–337, doi:10.1109/SFCS.1988.21949.

[13] E.A. Emerson & C.S. Jutla (1991): *Tree Automata, Mu-Calculus and Determinacy (Extended Abstract)*. In: *Proc. of FOCS*, IEEE Computer Society, pp. 368–377, doi:10.1109/SFCS.1991.185392.

[14] E.A. Emerson, C.S. Jutla & A.P. Sistla (1993): *On Model-Checking for Fragments of μ-Calculus*. In: *Proc. of CAV*, LNCS 697, Springer, pp. 385–396, doi:10.1007/3-540-56922-7_32.

[15] N. Fijalkow & F. Horn (2010): *The surprizing complexity of generalized reachability games*. *CoRR* abs/1010.2420. Available at http://arxiv.org/abs/1010.2420.

[16] N. Fijalkow & M. Zimmermann (2014): *Parity and Streett Games with Costs*. *LMCS* 10(2), doi:10.2168/LMCS-10(2:14)2014.

[17] E. Grädel, W. Thomas & T. Wilke, editors (2002): *Automata, Logics, and Infinite Games: A Guide to Current Research*. LNCS 2500, Springer, doi:10.1007/3-540-36387-4.

[18] F. Horn (2005): *Streett Games on Finite Graphs*. In GDV'05.

[19] N. Immerman (1981): *Number of Quantifiers is Better Than Number of Tape Cells*. *J. Comput. Syst. Sci.* 22(3), pp. 384–406, doi:10.1016/0022-0000(81)90039-8.

[20] M. Jurdzinski (1998): *Deciding the Winner in Parity Games is in UP $\cap$ co-UP*. *Inf. Process. Lett.* 68(3), pp. 119–124, doi:10.1016/S0020-0190(98)00150-1.

[21] M. Jurdzinski (2000): *Small Progress Measures for Solving Parity Games*. In: *Proc. of STACS*, LNCS 1770, Springer, pp. 290–301, doi:10.1007/3-540-46541-3_24.

[22] M. Jurdzinski, M. Paterson & U. Zwick (2008): *A Deterministic Subexponential Algorithm for Solving Parity Games*. *SIAM J. Comput.* 38(4), pp. 1519–1532, doi:10.1137/070686652.

[23] O. Kupferman, N. Piterman & M.Y. Vardi (2009): *From liveness to promptness*. *FMSD* 34(2), pp. 83–103, doi:10.1007/s10703-009-0067-z.

[24] D.A. Martin (1975): *Borel determinacy*. *Annals of Mathematics* 102(2), pp. 363–371, doi:10.2307/1971035.

[25] M. Randour (2013): *Automated Synthesis of Reliable and Efficient Systems Through Game Theory: A Case Study*. In: *Proc. of ECCS 2012*, Springer Proceedings in Complexity XVII, Springer, pp. 731–738, doi:10.1007/978-3-319-00395-5_90.

[26] S. Schewe (2007): *Solving Parity Games in Big Steps*. In: *Proc. of FSTTCS*, LNCS 4855, Springer, pp. 449–460, doi:10.1007/978-3-540-77050-3_37.

[27] W. Thomas (1997): *Languages, Automata, and Logic*. In: *Handbook of Formal Languages*, chapter 7, 3, Beyond Words, Springer, pp. 389–455, doi:10.1007/978-3-642-59126-6_7.

[28] N. Wallmeier, P. Hütten & W. Thomas (2003): *Symbolic Synthesis of Finite-State Controllers for Request-Response Specifications*. In: *Proc. of CIAA*, LNCS 2759, Springer, pp. 11–22, doi:10.1007/3-540-45089-0_3.

[29] A. Weinert & M. Zimmermann (2016): *Easy to Win, Hard to Master: Optimal Strategies in Parity Games with Costs*. In: *Proc. of CSL*, LIPIcs, Schloss Dagstuhl - LZI. To appear.

[30] W. Zielonka (1998): *Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees*. *Theor. Comput. Sci.* 200(1-2), pp. 135–183, doi:10.1016/S0304-3975(98)00009-7.