Games with Weighted Multiple Objectives*

Orna Kupferman and Noam Shenwald

School of Computer Science and Engineering, The Hebrew University, Israel

Abstract. Games with multiple objectives arise naturally in synthesis of reactive systems. We study *games with weighted multiple objectives*. The winning objective in such games consists of a set α of underlying objectives, and a weight function $w : 2^\alpha \to \mathbb{N}$ that maps each subset S of α to a reward earned when exactly all the objectives in S are satisfied. The goal of a player may be to maximize or minimize the reward. As a special case, we obtain games where the goal is to maximize or minimize the number of satisfied objectives, and in particular satisfy them all (a.k.a. generalized conditions). A weight function allows for a much richer reference to the underlying objectives: prioritizing them, referring to desired and less desired combinations, and addressing settings where we cannot expect all sub-specifications to be satisfied together.

We focus on settings where the underlying objectives are all *Büchi*, *co-Büchi*, *reachability*, or *avoid* objectives, and the weight function is non-decreasing (a.k.a. *free disposal*). For each of the induced classes (that is, type of underlying condition, type of optimization, and type of weight function), we solve the problem of deciding the game and analyze its tight complexity. We also study the tight memory requirements for each of the players. Finally, we consider general weight functions, which make the setting similar to the one of Boolean Muller objectives.

1 Introduction

Synthesis is the automated construction of a system from its specification [37]. A *reactive* system interacts with its environment and has to satisfy its specification in all environments [23]. A useful way to approach synthesis of reactive systems is to consider the situation as a *game* between the system and its environment. In the *turn-based* setting, the game is played on a graph whose vertices are partitioned between the system and the environment: starting from an initial vertex, the players jointly generate a *play*, namely a path in the graph, with each player deciding the successor vertex when the play reaches a vertex she owns. The objectives of the players refer to the infinite play that they generate. Each objective α defines a subset of V^ω [33], where V is the set of vertices of the game graph. In some settings, the specification of α is *behavioral*: the vertices in V are labeled by assignments to a set AP of atomic propositions – these with respect to which the system is defined, and α is a language of infinite words

* This research was supported by the Israel Science Foundation, Grant 2357/19, and European Research Council, Advanced Grant ADVANSYNT.

in $(2^{AP})^\omega$. In other settings (and in the process of reasoning about behavioral specifications), the specification of α is *structural*: it is specified as an ω-regular objective on V [4].

The most basic ω-regular objectives are *reachability* [27] and *Büchi* [9], and their respective dual *avoid* (also known as *safety*) and *co-Büchi* objectives. Both are given by a set of vertices $\alpha \subseteq V$. A play satisfies a reachability objective α if it visits a vertex in α at least once, and it satisfies a Büchi objective α if it visits some vertex in α infinitely often.

Traditional synthesis is Boolean: a computation satisfies a specification or it does not. In many applications, the setting is not Boolean. Quantitative aspects may arise from the setting itself, for example when actions involve costs or rewards (cf., energy games [11]) or when the assignments to the atomic propositions are multi-valued [22]. A key challenge in these settings is an evaluation of an infinite sequence of values using measures like its mean-payoff, discounted sum, etc. [18, 24, 6, 5]. Quantitative aspects may also arise from the specification, which may refer to the *quality* in which the specification is satisfied [1]. In particular, in *Objective LTL* [29], a specification consists of a set of LTL formulas, along with a function mapping each subset of them to a reward earned when exactly all the formulas in the subset are satisfied.

The quantitative setting enables the designer to combine many aspects of the synthesized system, for example when studying trade-offs between the satisfaction value of a multi-valued specification and the budget used for activating sensors [17, 2], gaining control [3, 31], paying tolls [32], or consuming energy [10]. The richness of the setting, as well as the fact that synthesis typically handles a conjunction of requirements, makes games with *multiple objectives* of special interest. In these games, the objectives of the players are specified by a collection of ω-regular objectives. In the Boolean setting, this includes games with *generalized Büchi* [13], *generalized co-Büchi* [30], *generalized parity* [14], *generalized reachability* [21], and *generalized reactivity* (GR(1)) [36, 13] objectives. In addition to the deterministic turn-based setting, these games have been studied in various richer settings, like concurrent, stochastic, and energy games [7, 16, 12, 41].

In this work we add a quantitative aspect to multiple objectives by weighing them. We introduce and study *games with weighted multiple objectives*. Consider a game graph with vertices in V. An objective in our game is a tuple $\langle \alpha, w, t \rangle$ where $\alpha \subseteq 2^V$, is a set of objectives that are all Büchi (B), co-Büchi (C), reachability (R), or avoid (A) objectives, $w : 2^\alpha \to \mathbb{N}$ is a weight function that maps each subset S of α to a reward earned when exactly all the objectives in S are satisfied, and $t \geq 0$ is a threshold. Consider a play ρ, and let $S \subseteq \alpha$ be the set of objectives in α that ρ satisfies. For example, if α consists of Büchi objectives, then exactly all the sets in S are visited infinitely often in ρ, and if α consists of avoid objectives, then exactly all the sets in S are never visited along ρ. Then, the satisfaction value of the play ρ in the game is $w(S)$. An objective can be viewed as a *maximization* objective, in which case the goal is to maximize its satisfaction value (and t serves as a lower bound) or a *minimization* objective, in which case the goal is to minimize its satisfaction value (and t serves as an upper bound).[1]

Weighted objectives enable the user to prioritize different scenarios. The different objectives in α may correspond, for example, to different types of grants given by a server or different storage options in the cloud. Using the weight function w, the user can then express the utility of different combinations of grants, storage options, locations, and more. As a concrete example, consider a warehouse robot assigned to patrol and fulfill item retrieval requests from shelves. The environment issues requests for items, and the system directs the robot. The setting can be modeled by a game graph whose states correspond to locations in the warehouse. The robot's objective encompasses both the retrieval of items from appropriate shelves and logistical tasks such as visiting charging stations or navigating areas covered by security cameras. Different sets of locations within the warehouse are associated with varying rewards, reflecting diverse priorities related to requested items, specific shelves for retrieval, and the logistical considerations.

Studying games with weighted multiple objectives, we focus on *non-decreasing* weight functions (a.k.a. functions that respect *free disposal* [35]): for every two sets $S, S' \subseteq \alpha$, if $S \subseteq S'$, then $w(S) \leq w(S')$. We use the acronym MaxWB in order to denote a weighted maximization objective game with underlying Büchi objectives, and similarly for minimization objectives and the other classes of underlying objectives. For example, in MinWR games, the system aims to minimize the weight of reachability objectives. We also consider the special case where w counts the number of objectives satisfied, thus $w(S) = |S|$, and so the goal is to maximize (Max) or minimize (Min) the number of objectives satisfied. Note that the generalized conditions mentioned above are a special case of the latter, with $t = |\alpha|$. Here, we call them *All* objectives, and refer also to their dual *Exists* objectives. For example, AllB is a generalized Büchi game.

For all classes of games with weighted multiple objectives, we study the problem of *deciding the winner* in the game: the system wins a MaxW (MinW) game with objective $\langle \alpha, w, t \rangle$ if it has a strategy ensuring a satisfaction value at least (at most, respectively) t in all plays. Note that while the winning criterion is Boolean, by searching for the largest or smallest t with which the system wins, we can solve also the *optimization* (rather than *decision*) variants of the problem.

In order to satisfy an objective, a player may need to choose different successors of a vertex in different visits to the vertex. Indeed, a-priori, choices may depend not only on the current vertex but also on the history of the play so far. The number of histories is unbounded, and extensive research has concerned the *memory requirements* for players in games with ω-regular objectives, namely the minimal number of equivalence classes to which the histories can be partitioned [39, 19, 8]. In particular, *memoryless* strategies depend only on the current ver-

[1] As we elaborate in Remark 1, our setting can easily capture also a semantics in which the satisfaction value of the play ρ is $w(S) - w(\alpha \setminus S)$; thus when players are punished (in the maximization variant, or rewarded, in the minimization variant) for objectives that are not satisfied.

tex in the game, thus all histories are in one equivalence class. We study the memory requirements in games with weighted multiple objectives.

Our results about the complexity of deciding a game and the memory requirements are summarized in Table 1 below.[2] Hardness results hold already for the cases of Max and Min objectives, thus when w is uniform. It is not hard to see that games with weighted multiple objectives can be dualized in two manners: a MaxWB objective can be dualized to either a MinWB objective or a MaxWC objective. Consequently, when we study both MaxWB and MinWB games, we also cover MaxWC and MinWC games, and similarly for reachability and avoid games. Thus, with an appropriate dualization of the weight function and the threshold (see Proposition 1), the table includes the results also for co-Büchi and avoid objectives.

	deciding the winner	memory requirements
AllB	PTIME [13]	$\lvert\alpha\rvert$ [19]
ExistsC		1 [20]
MaxWB	co-NP-complete (Theorem 5)	$\mathsf{width}(\alpha, w, t)$ (Theorem 4)
MinWB	NP-complete (Theorem 5)	1 (Theorem 2)
AllR	PSPACE-complete [21]	$2^{\lvert\alpha\rvert} - 1$ [21]
ExistsA		$\binom{\lvert\alpha\rvert}{\lvert\alpha\rvert/2}$ [21]
MaxWR	PSPACE-complete (Theorem 12)	$\mathsf{equiv}(\alpha, w, t)$ (Theorem 8)
MinWR		$\mathsf{sepwidth}(\alpha, w, t)$ (Theorem 11)

Table 1. Complexity and memory requirements for objectives $\langle \alpha, w, t\rangle$. The definitions of the measures $\mathsf{width}(\alpha, w, t)$, $\mathsf{equiv}(\alpha, w, t)$, and $\mathsf{sepwidth}(\alpha, w, t)$ can be found in Sections 3 and 4.

Below we highlight the main conclusions from our complexity results. Recall that single-objective reachability and Büchi games can be solved in PTIME, and both players have memoryless strategies [27, 40]. Interestingly, while reachability objectives are typically easier than Büchi (in particular, the PTIME above is linear for reachability and quadratic for Büchi), the fact Büchi refers to limit behaviors makes it easier when we move to multiple objectives. Indeed, moving to AllB and AllR objectives, the complexity stays in PTIME for Büchi and jumps to PSPACE for reachability [13, 21]. Also, the system player now needs memory, polynomial in the case of Büchi and exponential in the case of reacha-

[2] While the problem of deciding games has several parameters (the game, the number of objectives, the weight function, and the threshold), a parameterized analysis is not of much interest, as fixing the game also fixes the number of objectives, and, as we shall show, once the number of objectives is fixed, all the decision problems can be solved in polynomial time.

bility [19, 21]. Intuitively, this follows from the fact that detecting satisfaction of Büchi objectives, one may ignore visits to the objectives along the play, whereas detecting satisfaction of reachability objectives, one must keep track of all visits. In particular, in AllB objectives, one can detect the underlying objectives in a round-robin fashion, which is impossible in AllR. How does this difference affect weighted multiple objectives? Our results show that the jump to PSPACE in AllR games is sufficiently high to solve also MaxWR games, and the memory requirements are bounded by these in AllR games. That is, as detailed in Section 4, the measures $\mathsf{equiv}(\alpha, w, t)$ and $\mathsf{sepwidth}(\alpha, w, t)$ are bounded by $2^{|\alpha|} - 1$ and $\binom{|\alpha|}{|\alpha|/2}$, respectively. For Büchi objectives, MaxWB games are still easier than MaxWR games, yet are much harder than AllB games, and the memory requirements are higher. In particular, as detailed in Section 3, the measure $\mathsf{width}(\alpha, w, t)$ may be exponential in $|\alpha|$.

From a technical point of view, while both MaxWB and MaxWR objectives can be translated to equivalent AllB and AllR objectives, such a translation lead to optimal upper bounds only in the case of Büchi. Also, our lower bounds involve games and reductions that capture the difference between generalized and weighted objectives.

Finally, we show that games with weighted Büchi and reachability objectives with general (that is, not necessarily non-decreasing) weight functions correspond to games with Boolean *Muller* objectives, which describe the exact sets of vertices that should be visited infinitely often (termed B-Muller), or reached (termed R-Muller, a.k.a. *weak-Muller* or *Staiger-Wagner*). The correspondence enables us to lift known results about games with Muller objectives. In particular, deciding Muller games of both types is PSPACE-complete [26, 34], implying membership in PSPACE for MaxWB and MaxWR games with general weight function. Thus, restricting attention to non-decreasing function makes a difference in the complexity only when the underlying objectives are Büchi. As for the memory requirements, for B-Muller, it can be analyzed using *Zielonka trees* [19], and for R-Muller it was studied in [25]. In fact, results relating the structure of Zielonka trees and the memory requirements for the corresponding objective [8] suggest an alternative proof also for the analysis of the memory requirements for non-decreasing weight functions, which we describe in Appendix B.

In Section 6, we discuss further extensions of the setting, in particular the connection between changes in the type of the underlying objectives (for example, to parity) and changes to the class of weight functions.

2 Preliminaries

2.1 Two-player games

A *two-player game graph* is a tuple $G = \langle V_1, V_2, v_0, E\rangle$, where V_1, V_2 are disjoint sets of vertices, controlled by Player 1 and Player 2, respectively, and we let $V = V_1 \cup V_2$. Then, $v_0 \in V$ is an initial vertex, and $E \subseteq V \times V$ is a total edge relation, thus for every $v \in V$, there is $u \in V$ such that $(v, u) \in E$. The size of G, denoted $|G|$, is $|E|$, namely the number of edges in it.

In the beginning of a play in the game, a token is placed on v_0. Then, in each turn, the player that owns the vertex that hosts the token chooses a successor vertex and moves the token to it. Together, the players generate a *play* $\rho = v_0, v_1, \ldots$ in G, namely an infinite path that starts in v_0 and respects E: for all $i \geq 0$, we have that $(v_i, v_{i+1}) \in E$.

For $i \in \{1, 2\}$, a *strategy* for Player i is a function $f_i : V^* \cdot V_i \to V$ that maps prefixes of plays that end in a vertex that belongs to Player i to possible extensions in a way that respects E. That is, for every $\rho \in V^*$ and $v \in v_i$, we have that $(v, f_i(\rho \cdot v)) \in E$. Intuitively, a strategy for Player i directs her how to move the token, and the direction may depend on the history of the game so far.

A strategy is *finite-memory* if it is possible to replace the unbounded histories in $V^* \cdot V_i$ by finite memories. Formally, a *memory structure* for a game graph $G = \langle V_1, V_2, v_0, E \rangle$ is $\mathcal{M} = \langle M, \mu_0, \delta \rangle$, consisting of a finite set M of memory states, an initial memory state $\mu_0 \in M$, and an update function $\delta : M \times E \to M$. A memory structure is similar to an automaton with alphabet E, which is executed in parallel to the game: it starts from μ_0 and reads the edges traversed by the token. Formally, a strategy for Player i that relies on $\mathcal{M}$ replaces the dependency on the history of the play by dependency on the current vertex of the game and the current memory state of $\mathcal{M}$. Thus, the strategy is given by a function $f_i : M \times V_i \to V$, such that for all $\mu \in M$ and $v \in V_i$, we have that $(v, f_i(\mu, v)) \in E$. When the current memory state is μ and the token is in vertex $v \in V_i$, Player i moves the token to $f_i(\mu, v)$ and $\mathcal{M}$ moves to state $\delta(\mu, (v, f_i(\mu, v)))$. The strategy f_i is *memoryless* if it relies on a memory structure with a single memory state. It is thus given by a function $f_i : V_i \to V$.

A *profile* is a tuple $\pi = \langle f_1, f_2 \rangle$ of strategies, one for each player. The *outcome* of a profile $\pi = \langle f_1, f_2 \rangle$ is the play obtained when the players follow their strategies in π. Formally, $\mathsf{Outcome}(\pi) = v_0, v_1, \ldots \in V^\omega$ is such that for all $j \geq 0$, we have that $v_{j+1} = f_i(v_0, v_1, \ldots, v_j)$, where $i \in \{1, 2\}$ is such that $v_j \in V_i$. For finite-memory strategies, the definition is similar, with f_i being defined over memory states.

A *two-player game* is a pair $\mathcal{G} = \langle G, \psi \rangle$, where $G = \langle V_1, V_2, v_0, E \rangle$ is a two-player game graph, and ψ is a winning condition for Player 1, specifying a subset of V^ω, namely the set of plays in which Player 1 wins. The game is zero-sum, thus Player 2 wins when the play does not satisfy ψ. A strategy f_1 is a *winning strategy* for Player 1 if for every strategy f_2 for Player 2, we have that Player 1 wins in $\langle f_1, f_2 \rangle$, thus $\mathsf{Outcome}(\langle f_1, f_2 \rangle)$ satisfies ψ. Dually, a strategy f_2 for Player 2 is a winning strategy for Player 2 if for every strategy f_1 for Player 1, we have that Player 2 wins in $\langle f_1, f_2 \rangle$. We say that Player i *wins* in $\mathcal{G}$ if she has a winning strategy. A game is *determined* if Player 1 or Player 2 wins it. For $m \geq 1$, we say that Player i wins in $\mathcal{G}$ with memory m if she has a winning strategy that relies on a memory structure of size m.

2.2 Weighted multiple objectives

For a play $\rho = v_0, v_1, \ldots$, we denote by *reach*(ρ) the set of vertices that are visited at least once along ρ, and we denote by *inf*(ρ) the set of vertices that are visited infinitely often along ρ. That is, *reach*$(\rho) = \{v \in V : \text{there exists } i \geq 0 \text{ such that } v_i = v\}$, and *inf*$(\rho) = \{v \in V : \text{there are infinitely many } i \geq 0 \text{ such that } v_i = v\}$. For a set of vertices $\alpha \subseteq V$, a play ρ satisfies the *reachability* objective α iff *reach*$(\rho) \cap \alpha \neq \emptyset$, and satisfies the *Büchi* objective α iff *inf*$(\rho) \cap \alpha \neq \emptyset$. The objectives dual to reachability and Büchi are *avoid* and *co-Büchi*, respectively. Formally, a play ρ satisfies an avoid objective α iff *reach*$(\rho) \cap \alpha = \emptyset$, and satisfies a co-Büchi objective α iff *inf*$(\rho) \cap \alpha = \emptyset$.

A *weighted objective* is a pair $\langle \alpha, w \rangle$, where $\alpha = \{\alpha_1, \ldots, \alpha_m\}$ is a set of m objectives, all of the same type, and $w : 2^\alpha \to \mathbb{N}$ is a weight function that maps subsets of objectives in α to natural numbers. [3] For $m \geq 1$, let $[m] = \{1, \ldots, m\}$. We assume that w is *non-decreasing*: for every sets $S, S' \subseteq \alpha$, if $S \subseteq S'$, then $w(S) \leq w(S')$. In the context of game theory, non-decreasing functions are very useful, as they correspond to settings with *free disposal*, namely when satisfaction of additional objectives does not decrease the utility [35]. We also assume that $w(\emptyset) = 0$. Note that we can set $w(\emptyset)$ to 0 preserving the non-decrease of w. A non-decreasing weight function is *additive* if for every set $S \subseteq \alpha$, the weight of S equals to the sum of weights of the singleton subsets that constitute S. That is, $w(S) = \sum_{\alpha_i \in S} w(\{\alpha_i\})$. An additive weight function is thus given by $w : \alpha \to \mathbb{N}$, and is extended to sets of objectives in the expected way, thus $w(S) = \sum_{\alpha_i \in S} w(\alpha_i)$, for every $S \subseteq \alpha$. Finally, an additive weight function is *uniform* if $w(\alpha_i) = 1$ for all $\alpha_i \in \alpha$. Thus, for all $S \subseteq \alpha$, we have that $w(S) = |S|$.

For a play ρ, let $\mathsf{sat}(\rho, \alpha) \subseteq \alpha$ be the set of objectives in α that are satisfied in ρ. The *satisfaction value* of $\langle \alpha, w \rangle$ in ρ, denoted $\mathsf{val}(\rho, \alpha, w)$, is then the weight of the set of objectives in α that are satisfied in ρ. That is, $\mathsf{val}(\rho, \alpha, w) = w(\mathsf{sat}(\rho, \alpha))$.

For every vertex $v \in V$, we denote by $\mathsf{obj}(v)$ the set of objectives that contain v, thus $\mathsf{obj}(v) = \{\alpha_i \in \alpha : v \in \alpha_i\}$. We extend the function obj to sets of vertices in the expected way; thus, for $U \subseteq V$, we have that $\mathsf{obj}(U) = \bigcup_{v \in U} \mathsf{obj}(v) = \{\alpha_i \in \alpha : \alpha_i \cap U \neq \emptyset\}$.

Weighted objectives can be viewed as either maximization or minimization objectives. That is, in general non-decreasing weighted functions, the goal is to maximize or minimize the weight of the set of objectives satisfied, and in uniform additive weight functions, the goal is to maximize or minimize their number. A special case of the latter, known in the literature as *generalized* conditions, is when we aim to satisfy all or at least one objective. We denote different classes of weighted objectives by acronyms in $\{\text{MaxW, MinW, Max, Min, All, Exists}\} \times$

[3] One could also define multiple weighted objectives with mixed types of objectives. Essentially, combining objectives of the same polarity (that is, Buchi and reachability, or co-Buchi and avoid), we expect the properties of the game to be dominated by the complexity of the harder objective. Then, combining objectives with different polarities (for example, Buchi and co-Buchi), things become more complicated, as the combined objectives have the flavor of Rabin or Streett objectives.

{R, A, B, C}, where the first letter describes the way we refer to the satisfaction value, and the second letter describes the objectives type (reachability, avoid, Büchi, or co-Büchi).

Formally, for a play $\rho \in V^\omega$, an objective type $\gamma \in \{\text{R, A, B, C}\}$, an objective $\alpha \subseteq 2^V$, a weight function $w : 2^\alpha \to \mathbb{N}$, and a threshold $t \in \mathbb{N}$, we have the following winning conditions.

- ρ satisfies a MaxW-γ objective $\langle \alpha, w, t \rangle$ if $\mathsf{val}(\rho, \alpha, w) \geq t$.
- ρ satisfies a MinW-γ objective $\langle \alpha, w, t \rangle$ if $\mathsf{val}(\rho, \alpha, w) \leq t$.
- ρ satisfies a Max-γ objective $\langle \alpha, t \rangle$ if $|\mathsf{sat}(\rho, \alpha)| \geq t$.
- ρ satisfies a Min-γ objective $\langle \alpha, t \rangle$ if $|\mathsf{sat}(\rho, \alpha)| \leq t$.
- ρ satisfies an All-γ objective α if $|\mathsf{sat}(\rho, \alpha)| = |\alpha|$.
- ρ satisfies an Exists-γ objective α if $|\mathsf{sat}(\rho, \alpha)| \geq 1$.

We also consider *strict satisfaction*, where the bound set by the threshold is strict. For example, ρ strictly satisfies a MinW-γ objective $\langle \alpha, w, t \rangle$ if $\mathsf{val}(\rho, \alpha, w) < t$. Note that as a consequence of Martin's determinacy theorem [33], games with multiple weighted objectives are determined.

Weighted objectives may be dualized in two ways, by complementing either the type of the objective or the way we refer to the satisfaction value. For an objective type $\gamma \in \{\text{R, A, B, C}\}$, let $\tilde{\gamma}$ be the dual objective, thus $\tilde{R} = A$ and $\tilde{B} = C$. Consider an objective $\psi = \langle \alpha, w, t \rangle$. The dual weight function of w, denoted $\tilde{w}$, is defined, for every $S \subseteq \alpha$, by $\tilde{w}(S) = w(\alpha) - w(\alpha \setminus S)$. The dual threshold of t, denoted $\tilde{t}$, is defined as $w(\alpha) - t$. Then, the *dual objective of* ψ, denoted $\tilde{\psi}$, is $\langle \alpha, \tilde{w}, \tilde{t} \rangle$. Note that for an additive weight function, we have that $\tilde{w} = w$. The following lemma follows directly from the definitions.

Lemma 1. *For every objective ψ, we have that $\tilde{\tilde{\psi}} = \psi$. Also, if ψ is non-decreasing, then $\tilde{\psi}$ is non-decreasing.*

Proof. For the first claim, recall that $w(\emptyset) = 0$ and note that $\tilde{\tilde{t}} = \tilde{w}(\alpha) - \tilde{t} = (w(\alpha) - w(\emptyset)) - (w(\alpha) - t) = t$, and for every $S \subseteq \alpha$, we have that $\tilde{\tilde{w}}(S) = \tilde{w}(\alpha) - \tilde{w}(\alpha \setminus S) = (w(\alpha) - w(\emptyset)) - (w(\alpha) - w(\alpha \setminus (\alpha \setminus S))) = w(S)$.

For the second claim, assume that w is non-decreasing, and consider two sets $S, S' \subseteq \alpha$. If $S \subseteq S'$, then $(\alpha \setminus S') \subseteq (\alpha \setminus S)$. Since w is non-decreasing, this implies that $w(\alpha \setminus S') \leq w(\alpha \setminus S)$, and so $\tilde{w}(S) = w(\alpha) - w(\alpha \setminus S) \leq w(\alpha) - w(\alpha \setminus S') = \tilde{w}(S')$, as required. □

Proposition 1 below formalizes the different types of dualities.

Proposition 1. *Consider a play ρ, and an objective $\psi = \langle \alpha, w, t \rangle$ with objectives of type γ. The following are equivalent.*

1. *The play ρ satisfies the MaxW-γ objective ψ.*
2. *The play ρ does not strictly satisfy the MinW-γ objective ψ.*
3. *The play ρ does not strictly satisfy the MaxW-$\tilde{\gamma}$ objective $\tilde{\psi}$.*
4. *The play ρ satisfies the MinW-$\tilde{\gamma}$ objective $\tilde{\psi}$.*

Proof. The equivalence between 1 and 2, as well as between 3 and 4, follow immediately from the definitions. We prove the equivalence between 1 and 4, which completes the proof. Let S be the set of γ-objectives in α that ρ satisfies. The set of $\tilde{\gamma}$-objectives in α that ρ satisfies is then $\alpha \setminus S$. Clearly, $w(S) \geq t$ iff $w(\alpha) - w(S) \leq w(\alpha) - t$. Thus, by the definition of $\tilde{w}$ and $\tilde{t}$, we have that $w(S) \geq t$ iff $\tilde{w}(\alpha \setminus S) \leq \tilde{t}$. Hence, ρ satisfies the MaxW-γ objective $\langle \alpha, w, t \rangle$ iff ρ satisfies the MinW-$\tilde{\gamma}$ objective $\langle \alpha, \tilde{w}, \tilde{t} \rangle$, and we are done. □

A special case of Proposition 1 is that ρ satisfies an All-γ condition, which is a special case of Max-γ with $t = m$, iff it does not satisfy the Exists-$\tilde{\gamma}$ condition. Finally, note that for $\gamma \in \{R, B\}$, we have that ρ satisfies an Exists-γ objective α iff ρ satisfies the Exists-γ singleton objective $\{\cup\alpha\}$. Dually, for $\gamma \in \{A, C\}$, we have that ρ satisfies an All-γ objective α iff ρ satisfies the All-γ singleton objective $\{\cup\alpha\}$.

Recall that in zero-sum games, the objectives of the players complement each other. Thus, by Proposition 1, for every objective type γ, when Player 1 has a MaxW-γ or, equivalently, a MinW-$\tilde{\gamma}$ objective, Player 2 has a MinW-γ or, equivalently, a MaxW-$\tilde{\gamma}$ objective. In addition, our definition below of the size of an objective ψ is such that $|\psi| = |\tilde{\psi}|$. Consequently, Proposition 1 enables us to lift results on B and R objectives to C and A objectives, respectively.

We define the *size* of a game G as the size of its edge relation, and define the size of an objective $\psi = \langle \alpha, w, t \rangle$ with a non-decreasing weight function as the size of w, defined as follows. First, we define the *length* of a weight function w, denoted $|w|$, as $\sum_{S \subseteq \alpha} w(S)$. Then, we define the *size* of w as $\min\{|w|, |\tilde{w}|\}$. Thus, the size of w as the length of the shorter function among w and $\tilde{w}$. Note that w can indeed be encoded in $|w|$ bits (in fact, for our upper bounds, one can also replace $w(S)$, which corresponds to an encoding of the weights in unary, by $\log w(S)$, which corresponds to a binary encoding), and our upper bounds are such that one can work with an encoding of either w or $\tilde{w}$. Moreover, when w is additive, our bounds hold also when we define its length by $\sum_{\alpha_i \in \alpha} w(\alpha_i)$.

Remark 1. **[Games with penalties]** In *games with penalties*, the satisfaction value of an objective $\psi = \langle \alpha, w, t \rangle$ in a play ρ, denoted $\mathsf{valp}(\rho, \alpha, w)$, is $w(\mathsf{sat}(\rho, \alpha)) - w(\alpha \setminus \mathsf{sat}(\rho, \alpha))$. Thus, objectives that are not satisfied reduce the satisfaction value. Note that the satisfaction value of a play may be negative. For example, if $\mathsf{sat}(\rho, \alpha) = \emptyset$, then $\mathsf{valp}(\rho, \alpha, w) = w(\emptyset) - w(\alpha) = -w(\alpha)$.

Our setting can easily capture games with penalties. Formally, as we show below, for every objective $\psi = \langle \alpha, w, t \rangle$, we can define, in linear time, an objective $\psi' = \langle \alpha, w', t' \rangle$, such that for every play ρ, we have that $\mathsf{valp}(\rho, \alpha, w) \geq t$ iff $\mathsf{val}(\rho, \alpha, w') \geq t'$. Accordingly, a game with penalties with objective ψ is equivalent to a game in our setting and objective ψ'.

Given ψ, we define $w'(S) = \frac{1}{2}(w(\alpha) + w(S) - w(\alpha \setminus S))$, for all $S \subseteq \alpha$, and define $t' = \frac{1}{2}(w(\alpha) + t)$. Note that indeed, for every $S \subseteq \alpha$, we have that $\mathsf{val}(\rho, \alpha, w) \geq t$ iff $w(S) - w(\alpha \setminus S) \geq t$ iff $\frac{1}{2}(w(\alpha) + w(S) - w(\alpha \setminus S)) \geq \frac{1}{2}(w(\alpha) + t)$ iff $\mathsf{valp}(\rho, \alpha, w') \geq t'$.

For example, if w is additive and uniform, then the satisfaction value of a play ρ with $\mathsf{sat}(\rho, \alpha) = S$ in a setting with penalties is $|S| - |\alpha \setminus S| = 2 \cdot |S| - |\alpha|$.

Accordingly, $\mathsf{valp}(\rho, \alpha, w) \geq t$ iff $2|S| - |\alpha| \geq t$. Note that then, $w'(S) = \frac{1}{2}(|\alpha| + |S| - |\alpha \setminus S|) = |S|$ and $t' = \frac{1}{2}(|\alpha| + t)$. Thus, $\mathsf{val}(\rho, \alpha, w) \geq t$ iff $2|S| - |\alpha| \geq t$ iff $|S| \geq \frac{1}{2}(|\alpha| + t)$ iff $\mathsf{valp}(\rho, \alpha, w') \geq t'$, as required. □

3 Weighted Büchi and co-Büchi Games

In this section we study weighted Büchi and co-Büchi games: the problem of deciding the winner, and the memory requirements for winning strategies for both players. We first show that every MaxWB objective has an equivalent AllB objective of exponential size. This enables us to use known results about AllB games, yet involves an exponential increase in the complexity. We show that MaxWB objectives (in fact, already MaxB objectives) are indeed exponentially more complex than AllB objectives.

Consider a MaxWB game $\langle G, \langle \alpha, w, t \rangle \rangle$, with $\alpha = \{\alpha_1, \ldots, \alpha_m\}$. We say that a set $S \subseteq \alpha$ is *t-short* if $w(S) < t$, and is *max-t-short* if S is t-short and maximal, in the sense that $w(S') \geq t$ for all $S' \subseteq \alpha$ such that $S \subsetneq S'$. A set $S \subseteq \alpha$ is *t-essential* if $\alpha \setminus S$ is max-t-short. Thus, S is t-essential iff $w(\alpha \setminus S) < t$ and S is minimal. We define the *width* of $\langle \alpha, w, t \rangle$, denoted $\mathsf{width}(\alpha, w, t)$, as the number of max-$t$-short (or, equivalently, t-essential) subsets of α.

Remark 2. **[On the size of** $\mathsf{width}(\alpha, w, t)$**]** It is not hard to see that $\mathsf{width}(\alpha, w, t)$ need not be polynomial in $|\alpha|$ or t. In particular, for MaxB games with m objectives and threshold t, namely when $w(\alpha_i) = 1$ for all $\alpha_i \in \alpha$, we have that $\mathsf{width}(\alpha, w, t) = \binom{m}{t-1}$. That is, the number of different subsets of α of size $t - 1$. In the general case, the calculation of $\mathsf{width}(\alpha, w, t)$ coincides with calculating the number of 0/1 Knapsack solutions for a knapsack of size t and items with weights w. The problem is $\#P$-complete, with some known polynomial-time approximation schemes [38]. □

Theorem 1. *Every MaxWB objective $\langle \alpha, w, t \rangle$ has an equivalent AllB objective of size $\mathsf{width}(\alpha, w, t)$. That is, for every MaxWB game $\langle G, \langle \alpha, w, t \rangle \rangle$, there is an AllB objective α' such that the winning vertices for Player 1 in $\langle G, \langle \alpha, w, t \rangle \rangle$ coincide with these in $\langle G, \alpha' \rangle$, and $|\alpha'| = \mathsf{width}(\alpha, w, t)$.*

Proof. Let $\psi = \langle \alpha, w, t \rangle$, with $\alpha = \{\alpha_1, \ldots, \alpha_m\}$. We define α' as the AllB objective that contains every set $\cup S$ for sets $S \subseteq \alpha$ that are t-essential. Recall that $\cup S \subseteq V$. We prove that the MaxWB condition ψ is equivalent to the AllB condition α'. Consider a play ρ in G, and let $S = \mathsf{sat}(\rho, \alpha)$. We show that $w(S) \geq t$ iff $S \cap S' \neq \emptyset$ for every t-essential set S'. We then conclude that ρ satisfies ψ iff ρ satisfies the Büchi objective $\cup S'$ for every t-essential set S', thus satisfies α'.

Assume first that $w(S) \geq t$. For every t-essential set S' we have that $w(\alpha \setminus S') < t$. Since w is non-decreasing and $w(S) \geq t$, it follows that $S \not\subseteq \alpha \setminus S'$, therefore $S \cap S' \neq \emptyset$. Assume now that $w(S) < t$, and let S' be a max-t-short set S' such that $S \subseteq S'$. Hence, for the t-essential set $\alpha \setminus S'$ it holds that $S \cap (\alpha \setminus S') = \emptyset$. □

3.1 Memory requirements

In this section we study the memory requirements of strategies in MaxWB games. We show that while Player 2 can use memoryless strategies, Player 1 needs memory of exponential size, thus exponentially more than the one required in AllB games.

We start with Player 2, where things are easy.

Theorem 2. *Player 2 wins a MaxWB game iff she has a memoryless winning strategy.*

Proof. By Theorem 1, for every MaxWB game $\langle G, \psi \rangle$, there exists an AllB objective α' such that Player 2 wins in $\langle G, \psi \rangle$ iff Player 2 wins in $\langle G, \alpha' \rangle$. Player 2 in AllB games has an ExistsC objective. The latter is a special case of a Rabin objective, and hence, by [20], Player 2 wins $\langle G, \psi \rangle$ iff she has a memoryless winning strategy. $\square$

For Player 1, the construction in the proof of Theorem 1 also gives an upper bound on the memory requirements. Indeed, since a winning strategy of Player 1 in an AllB game with m sets needs memory of size at most m [19], the construction implies a $\mathsf{width}(\alpha, w, t)$ upper bound on the size of the memory required for Player 1 in MaxWB games. Below we prove a matching lower bound. In Appendix B, we provide an alternative analysis of the memory requirements in MaxWB games, based on an analysis of the Zielonka trees they induce.

Theorem 3. *For every $m, t \in \mathbb{N}$ and a non-decreasing function $f : 2^{[m]} \to \mathbb{N}$, we can construct a MaxWB game $\mathcal{G}_{m,f,t} = \langle G_{m,f,t}, \langle \alpha, w, t \rangle \rangle$, such that all the following hold.*

1. $|\alpha| = m$.
2. $w(\{\alpha_i : i \in S\}) = f(S)$ *for every* $S \subseteq [m]$.
3. *Player 1 wins $\mathcal{G}_{m,w,t}$, yet every winning strategy for Player 1 requires memory* $\mathsf{width}(\alpha, w, t)$.

Proof. Consider $m, t \in \mathbb{N}$, and a non-decreasing function $f : 2^{[m]} \to \mathbb{N}$. We say that a set $S \subseteq [m]$ is *t-good* if $f(S) \geq t$, and is *min-t-good* if S is t-good and minimal, in the sense that $f(S') < t$ for all $S' \subset S$. If for every $i \in [m]$, there exists a min-t-good set S such that $i \in S$, then we define $\mathcal{X}$ as the set of min-t-good subsets of $[m]$. Otherwise, we define $\mathcal{X}$ as the set of t-good subsets of $[m]$. Note that for every $i \in [m]$, there exists $S \in \mathcal{X}$ such that $i \in S$. [4]

The MaxWB game $\mathcal{G}_{m,f,t}$ proceeds as follows (see exact definition in Appendix A.1). From the initial vertex v_0, Player 2 chooses a set $S \in \mathcal{X}$, Player 1 chooses an objective vertex $i \in S$, and the game returns to the initial vertex, where again Player 2 chooses a set, and so on. The objective of Player 1 is to visit infinitely often a set S of objectives vertices such that $f(S) \geq t$.

[4] The proof is valid also when $\mathcal{X}$ is defined as the set of t-good sets. Our definition, however, emphasizes the intuition behind the memory requirements for Player 1, which is the fact that every $i \in [m]$ is included in some of the t-good sets.

Since Player 2 can only choose among t-good sets, it is easy to see that Player 1 has a winning strategy in $\mathcal{G}_{m,f,t}$. We prove that every winning strategy f_1 for Player 1 needs at least $\mathsf{width}([m], f, t)$ different memory states. Since $w(\{\alpha_i : i \in S\}) = f(S)$ for every $S \subseteq [m]$, we conclude that every winning strategy f_1 for Player 1 needs at least $\mathsf{width}(\alpha, w, t)$ different memory states. We do this by presenting a strategy for Player 2 that forces every winning strategy f_1 to visit the initial vertex v_0 with $\mathsf{width}([m], f, t)$ different memory states.

Consider $i \in [m]$ and a memory state μ of f_1. We say that i is *avoided* in μ if for every play that starts in v_0 while in μ, the next objective vertex that f_1 chooses is not i. That is, i is avoided in μ if no matter which set vertex S Player 2 chooses from v_0, the strategy f_1 chooses from S an objective vertex that is not i. For every memory state μ, we denote by $\mathit{avoid}(\mu)$ the set of $i \in [m]$ that are avoided in μ.

Consider a max-t-short set $S \subseteq [m]$. We show that Player 2 has a strategy that leads the play to v_0 with a memory state μ_S of f_1 such that $S \subseteq \mathit{avoid}(\mu_S)$. This is done by describing a strategy f_S for Player 2 such that if not all $i \in S$ are avoided in the current memory state μ of f_1, the play returns to v_0 while visiting only objective vertices $i \in S$. Since $f(S) < t$, f is non-decreasing, and f_1 is a winning strategy, the above cannot continue forever, and so f_1 eventually reaches a memory state in which all $i \in S$ are avoided. The strategy f_S is defined as follows. Recall that not all $i \in S$ are avoided in μ. Thus, by definition, and since for every $i \in [m]$ there exists $S' \in \mathcal{X}$ such that $i \in S'$, there exists $i \in S$ and a set S' such that when Player 2 goes to the set vertex S', the strategy f_1 proceeds from S' to i. The strategy f_S then proceeds to S'. As required, the play then returns to v_0 while visiting only objectives vertices $i \in S$.

To complete the proof, we show that Player 2 can lead the play to a memory state μ with $\mathit{avoid}(\mu) = S$, for every max-t-short set S of $[m]$. We then proceed to describe a strategy for Player 2 that forces the play to visit as many memory states of f_1 as there are max-t-short sets, which implies the claim.

Consider a max-t-short set S of $[m]$, and let μ_S be such that $S \subseteq \mathit{avoid}(\mu_S)$. Since max-$t$-short sets are maximal, we have that $f(S \cup \{i\}) \geq t$ for all $i \in [m]$ such that $i \notin S$. Therefore, $S \cup \{i\} \not\subseteq \mathit{avoid}(\mu_S)$. Indeed, when Player 2 proceeds from v_0 to a vertex set S' such that S' is a min-t-good set and $S' \subseteq S \cup \{i\}$, Player 1 must proceed to an objective in S', which thus cannot be avoided. Hence, $S = \mathit{avoid}(\mu_S)$.

By the above, for every max-t-short set S of $[m]$, Player 2 has a strategy f_S that leads the play to v_0 with a memory state μ_S of f_1 such that $\mathit{avoid}(\mu_S) = S$. Consider a strategy of Player 2 that follows some order $S_1, S_2, \ldots, S_{\mathsf{width}([m],f,t)}$ on the max-t-short subsets of $[m]$, starts with $i = 1$, applies the strategy f_{S_i} until the play reaches v_0 with a memory state μ_{S_i} such that $\mathit{avoid}(\mu_{S_i}) = S_i$, and then switches to $f_{S_{i+1}}$, and so on until all max-t-short subsets are covered. Then, $\mathsf{Outcome}(\langle f_1, f_2 \rangle)$ visits v_0 with $\mathsf{width}([m], f, t) = \mathsf{width}(\alpha, w, t)$ different memory states of f_1, and we are done. □

We can now conclude with a tight bound for the memory required to Player 1.

Theorem 4. *A winning strategy for Player 1 in a MaxWB game* $\mathcal{G} = \langle G, \langle \alpha, w, t \rangle \rangle$ *requires memory* $\mathsf{width}(\alpha, w, t)$.

3.2 Decidability

In this section we study the decidability of MaxWB games, and show that they (in fact, already their MaxB special cases) are exponentially harder than AllB games.

Theorem 5. *Deciding MinWB (MaxWB) games is NP-complete (co-NP-complete, respectively). Hardness in NP (co-NP) applies already for MinB (MaxB, respectively) games. Fixing the number of underlying objectives, the problem is in PTIME.*

Proof. We prove the results for MinWB. The ones for MaxWB follow from determinacy of MinWB games.

We start with the upper bound. By Theorem 2, Player 1 wins a MinWB game iff she has a memoryless winning strategy. As we argue below, checking whether a given memoryless strategy for Player 1 is winning can be done in polynomial time, implying membership in NP.

Consider a MinWB game $\langle G, \langle \alpha, w, t \rangle \rangle$, with $\alpha = \{\alpha_1, \ldots, \alpha_m\}$. Let $G = \langle V_1, V_2, v_0, E \rangle$. For a memoryless strategy $f_1 : V_1 \to V$ for Player 1, let $G_{f_1} = \langle V, E_{f_1} \rangle$ be the sub-graph of G obtained by removing all edges not taken by f_1. Thus $\langle v, u \rangle \in E_{f_1}$ iff $f_1(v) = u$. Clearly, f_1 is winning iff all the paths from v_0 in G_{f_1} satisfy $\langle G, \langle \alpha, w, t \rangle \rangle$.

Note that there exists a path ρ in G_{f_1} such that $\mathsf{sat}(\rho, \alpha, w) \geq t$ iff there exists a non-trivial strongly connected component (SCC) C in G_{f_1} such that $w(\mathsf{obj}(C)) \geq t$. Accordingly, f_1 is a winning strategy for Player 1 iff $w(\mathsf{obj}(C)) < t$ for every non-trivial SCC C reachable from the initial vertex in G_{f_1}. Since the latter can be checked in polynomial time, we are done. Note that the check is polynomial also when the weight function is dualized, thus when $\tilde{w}$ is given. Indeed, we need to calculate $w(\mathsf{obj}(C))$ only for linearly many SCCs.

When the number of underlying objectives is fixed, we have that the corresponding AllB game from the proof of Theorem 1 is of polynomial size. Thus, a MaxWB game can be decided in polynomial time by solving the corresponding AllB game.

For the lower bound, we describe a reduction from SAT to MinB (for details, see Appendix A.2). That is, given a propositional formula φ in CNF, we construct a MinB game $\mathcal{G}_\varphi$ such that φ is satisfiable iff Player 1 wins $\mathcal{G}_\varphi$. Intuitively, we define $\mathcal{G}_\varphi$ so that Player 2 can require Player 1 to prove that every clause of φ is evaluated to **true** by asking Player 1 to choose a literal in the clause that is assigned **true**. Choosing a literal l involves a visit in a vertex associated with l. Accordingly, there exists a satisfying assignment to φ iff Player 1 can choose her responses so that at most n literals are visited infinitely often. □

4 Weighted Reachability and Avoid Games

In this section we study weighted reachability and avoid games: the problem of deciding the winner, and the memory requirements for both players.

First, it is not hard to see that the construction in Theorem 1 applies also to reachability objectives, thus every MaxWR objective has an equivalent AllR objective of exponential size.

Theorem 6. *Every MaxWR objective $\langle \alpha, w, t \rangle$ has an equivalent AllR objective of size* $\mathsf{width}(\alpha, w, t)$*.*

Known results about AllR games then imply upper bounds on the complexity of MaxWR games. Specifically, deciding AllR games is PSPACE-complete and the memory requirement for the players in AllR games are exponential in the number of reachability objectives. Combining this with the construction in Theorem 6, suggests a solution for MaxWR games. Since, however, $\mathsf{width}(\alpha, w, t)$ need not be polynomial in α, this is not optimal, and below we describe an algorithm that works on the MaxWR objective without translating it to an AllR objective, and is optimal.

Consider a MaxWR game $\mathcal{G} = \langle G, \langle \alpha, w, t \rangle \rangle$, with $G = \langle V_1, V_2, v_0, E \rangle$ and $\alpha = \{\alpha_1, \ldots, \alpha_m\}$. Recall that a set $S \subseteq \alpha$ is t-short if $w(S) < t$. We denote by $\mathfrak{S}$ the set of all t-short subsets of α, and we define the *volume* of $\langle \alpha, w, t \rangle$, denoted $\mathsf{volume}(\alpha, w, t)$, as the number of t-short subsets of α. Recall that $\mathsf{width}(\alpha, w, t)$ is the number of max-t-short subsets of α, thus the measure volume is much bigger than the measure width. Below we describe a reachability game $\mathcal{G}'$ of size $|G| \cdot \mathsf{volume}(\alpha, w, t)$, such that Player 1 wins $\mathcal{G}$ iff Player 1 wins $\mathcal{G}'$. Intuitively, $\mathcal{G}'$ follows the play in G while remembering the set of reachability objectives satisfied so far. Player 1 wins in a play in G' when the corresponding play in G satisfies $\langle \alpha, w, t \rangle$. We use the construction of $\mathcal{G}'$ in order to prove that Player 1 wins $\mathcal{G}$ iff she can force the satisfaction of $\langle \alpha, w, t \rangle$ within $|V| \cdot |\alpha|$ steps (see full proof in Appendix A.3), which we then use to prove various results for MaxWR games. Note that the reduction also implies that tools for solving reachability games can be used for solving MaxWR games.

Theorem 7. *For every MaxWR game $\mathcal{G} = \langle G, \langle \alpha, w, t \rangle \rangle$, there exists a reachability game $\mathcal{G}' = \langle G', \alpha' \rangle$ such that $|G'| = |G| \cdot \mathsf{volume}(\alpha, w, t)$ and the following are equivalent.*

1. *Player 1 wins in $\mathcal{G}$.*
2. *Player 1 wins in $\mathcal{G}'$.*
3. *Player 1 has a winning strategy in $\mathcal{G}'$ that satisfies α' within $|V| \cdot |\alpha|$ steps.*
4. *Player 1 has a winning strategy in $\mathcal{G}$ that satisfies $\langle \alpha, w, t \rangle$ within $|V| \cdot |\alpha|$ steps.*

4.1 Memory requirements

In this section we study the memory requirements of winning strategies for both players in MaxWR games.

Memory requirements for Player 1 We start with studying the memory requirement for Player 1 in a MaxWR game $\mathcal{G} = \langle G, \langle \alpha, w, t \rangle \rangle$. For a pair of t-short sets $S, S' \in \mathfrak{S}$, we say that S and S' are *equivalent* if for every t-short set $Q \in \mathfrak{S}$, we have that $w(S \cup Q) \geq t$ iff $w(S' \cup Q) \geq t$. Then, for every t-short set $S \in \mathfrak{S}$, the *equivalence class* of S, denoted $[S]$ is the set of t-short sets $S' \in \mathfrak{S}$ such that S and S' are equivalent. We denote by $\mathfrak{E}$ the set of different equivalence class of $\mathfrak{S}$. Below we show that the memory requirement for Player 1 is the number of equivalence classes of the MaxWR objective, denoted $\mathsf{equiv}(\alpha, w, t)$.

Note that for every MaxWR objective $\langle \alpha, w, t \rangle$ in which every t-short set is equivalent only to itself, for example, every MaxR objective, we have that $\mathsf{equiv}(\alpha, w, t) = \mathsf{volume}(\alpha, w, t)$. Also note that for an AllR objective α, we have that $\mathsf{equiv}(\alpha) = \mathsf{volume}(\alpha) = 2^{|\alpha|} - 1$, which coincides with the known memory requirement for Player 1 in AllR games [21].

Theorem 8. *A winning strategy for Player 1 in a MaxWR game with objective $\langle \alpha, w, t \rangle$ requires memory $\mathsf{equiv}(\alpha, w, t)$.*

Proof. The upper bound follows from the proof of Theorem 7. Indeed, it can be shown that Player 1 wins $\mathcal{G}$ iff she has a winning strategy that relies on a memory structure with memory states in $\mathfrak{S}$, and it is easy to see that we can replace them with memory states in $\mathfrak{E}$ in the expected way, and $|\mathfrak{E}| = \mathsf{equiv}(\alpha, w, t)$.

For the lower bound, for every $m, t \in \mathbb{N}$ and a non-decreasing function $f : 2^{[m]} \to \mathbb{N}$, we construct a MaxWR game that proceeds as follows (see exact definition and example in Appendix A.4). From the initial vertex v_0, Player 2 chooses an *init-set vertex* S, for some $S \in \mathfrak{S}$. From S, the game proceeds to a vertex v_1, from which Player 1 chooses an *equivalence-class vertex* $[S]$, for some $[S] \in \mathfrak{E}$. From $[S]$, Player 2 chooses a *final-set vertex* S', for some $S' \in \mathfrak{S}$ such that $f(S \cup S') \geq t$, or chooses an *inter-set vertex* S', for some $S' \in \mathfrak{S}$ such that $f(S \cup S') < t$, and $[S] \neq [S \cup S']$. Every final-set vertex is self-looped with no other outgoing edges, and from every inter-set vertex the game proceeds back to v_1. The objective of Player 1 is to reach set vertices that together form a set $S \subseteq [m]$ such that $f(S) \geq t$.

It is easy to see that Player 1 has a winning strategy in $\mathcal{G}_{m,f,t}$, as f is non-decreasing, and from every equivalence-class vertex $[S]$ Player 2 chooses a final-set vertex that corresponds to $S' \in \mathfrak{S}$ such that $f(S \cup S') \geq t$, or an inter-set vertex that corresponds to $S' \in \mathfrak{S}$ such that $[S] \neq [S \cup S']$. We prove that every winning strategy f_1 for Player 1 has at least $|\mathfrak{E}| = \mathsf{equiv}(\alpha, w, t)$ different memory states.

We show that for every $[S] \in \mathfrak{E}$, Player 2 has a strategy $f_{[S]}$ that causes f_1 to eventually proceed from v_1 to the equivalence-class vertex $[S]$. Since there are $\mathsf{equiv}(\alpha, w, t)$ equivalence class vertices, that are all reached via the same vertex for Player 1, we conclude that f_1 has at least $\mathsf{equiv}(\alpha, w, t)$ different memory states.

Consider $[S] \in \mathfrak{E}$. The strategy $f_{[S]}$ for Player 2 proceeds as follows. From the initial vertex v_0, the strategy $f_{[S]}$ chooses the init-set vertex S. Then, while Player 1 chooses from v_1 an equivalence-class vertex $[S']$ such that $[S'] \neq [S]$: if there exists $Q \in \mathfrak{S}$ such that $f(S' \cup Q) \geq t$ and $f(S \cup Q) < t$, the strategy $f_{[S]}$ proceeds from $[S']$ to the final-set vertex Q, thus the game ends while satisfying the set of reachability objectives $\{\alpha_i \in \alpha : i \in S \cup Q\}$; otherwise, the strategy $f_{[S]}$ proceeds from $[S']$ to the inter-set vertex S, thus the set of reachability objectives satisfied is still $\{\alpha_i \in \alpha : i \in S\}$. Below we show that indeed Player 2 can choose the inter-set vertex S from the equivalnce-class vertex $[S']$. Then, since f_1 is a winning strategy and $f(S) < t$, we have that eventually the play $\mathsf{Outcome}(\langle f_1, f_{[S]}\rangle)$ reaches the equivalence-class vertex $[S]$.

It is thus left to show that for every $[S'] \in \mathfrak{E}$ such that there does not exist $Q \in \mathfrak{S}$ for which $f(S' \cup Q) \geq t$ and $f(S \cup Q) < t$, we have that Player 2 can proceed from the equivalence-class vertex $[S']$ to the inter-set vertex S. Since $[S] \neq [S']$ and $f(S \cup Q) \geq t$ for every $Q \in \mathfrak{S}$ such that $f(S' \cup Q) \geq t$, we have that $f(S' \cup S) < t$. We also have that there exists $Q \in \mathfrak{S}$ such that $f(S \cup Q) \geq t$ and $f(S' \cup Q) < t$; thus, $f(S' \cup S \cup Q) \geq t$, and so $[S'] \neq [S' \cup S]$. Since $f(S' \cup S) < t$ and $[S'] \neq [S' \cup S]$, we have that there exists an edge between $[S']$ and the inter-set vertex S by the definition of $G_{m,f,t}$. □

Memory requirements for Player 2 We continue to Player 2, and study her memory requirement in a MaxWR game $\mathcal{G} = \langle G, \langle \alpha, w, t\rangle\rangle$. For this, we introduce the *separated-width* measure of $\langle \alpha, w, t\rangle$, and show that the memory requirement for Player 2 coincides with it.

Consider a MaxWR objective $\langle \alpha, w, t\rangle$, with $\alpha = \{\alpha_1, \ldots, \alpha_m\}$. For two t-short sets S, S' of α, we say that S and S' are *separated* iff there exist max-t-short sets $Q \subseteq \alpha$ and $Q' \subseteq \alpha$ such that $S \subseteq Q$ and $S' \not\subseteq Q$, and $S' \subseteq Q'$ and $S \not\subseteq Q'$. We say that a set $\mathfrak{P} \subseteq \mathfrak{S}$ of t-short sets is *pairwise separated* iff for every two sets $S, S' \in \mathfrak{P}$, we have that S and S' are separated. Then, the *separated-width* of $\langle \alpha, w, t\rangle$, denoted $\mathsf{sepwidth}(\alpha, w, t)$, is the maximal size of a pairwise separated set of t-short sets of α. That is, $\mathsf{sepwidth}(\alpha, w, t) = \max\{|\mathfrak{P}| : \mathfrak{P} \subseteq \mathfrak{S} \text{ and } \mathfrak{P} \text{ is pairwise separated}\}$.

Note that for the special case of AllR objectives, two sets $S, S' \in \mathfrak{S}$ are separated iff $S \not\subset S'$ and $S' \not\subset S$: First, it is easy to see that if $S \subset S'$ (similarly, if $S' \subset S$), then S and S' are not separated, since for every max-m-short set Q such that $S' \subseteq Q$, we also have that $S \subseteq Q$. For the second direction, if $S \not\subset S'$ and $S' \not\subset S$, then for every $\alpha_i \in S' \setminus S$, we have that $Q = \alpha \setminus \{\alpha_i\}$ is a max-m-short set such that $S \subseteq Q$ and $S' \not\subseteq Q$, and for every $\alpha_i \in S \setminus S'$, we have that $Q' = \alpha \setminus \{\alpha_i\}$ is a max-m-short set such that $S' \subseteq Q'$ and $S' \not\subseteq Q'$, thus S and S' are separated. Hence, $\mathsf{sepwidth}(\alpha)$ is the maximal size of a set $\mathfrak{P} \subseteq \mathfrak{S}$ such that $S \not\subset S'$, for every $S, S' \in \mathfrak{P}$. In particular, $\mathsf{sepwidth}(\alpha) = \binom{m}{m/2}$, which coincides with the known memory requirement for Player 2 in AllR games.

We return to the case of non-decreasing weight functions and start with the upper bound. Consider a vertex $v \in V$ and a t-short set S of α. We say that Player 2 *S-wins from v* if Player 2 has a strategy to win from v, given that the objectives in S have already been satisfied. Such a strategy is called an *S-winning strategy.* Note that since w is non-decreasing, then if Player 2 S-wins from v, then Player 2 also S'-wins from v, for every $S' \subseteq S$. We denote by $\mathsf{good}(v)$ the set of sets $S \subseteq \alpha$ such that Player 2 S-wins from v and S is maximal, in the sense that Player 2 does not S'-win from v, for all $S' \subseteq \alpha$ such that $S \subset S'$. First, we state that the set $\mathsf{good}(v)$ is pairwise separated (see full proof in Appendix A.5).

Lemma 2. *For every vertex $v \in V$ and two sets $S, S' \in \mathsf{good}(v)$, we have that S and S' are separated.*

We conclude that $|\mathsf{good}(v)| \leq \mathsf{sepwidth}(\alpha, w, t)$ for every vertex $v \in V$. We use this bound in order to prove an upper bound on the memory requirement for Player 2, which we prove to be tight.

Theorem 9. *Player 2 wins a MaxWR game $\mathcal{G} = \langle G, \langle \alpha, w, t\rangle\rangle$ iff she has a winning strategy that uses at most $\mathsf{sepwidth}(\alpha, w, t)$ different memory states.*

Proof. Consider a MaxWR game $\mathcal{G} = \langle G, \langle \alpha, w, t\rangle\rangle$, and assume Player 2 wins $\mathcal{G}$. Recall the reachability game $\mathcal{G}' = \langle G', \alpha'\rangle$ constructed from $\mathcal{G}$ in the proof of Theorem 7. Also recall that Player 2 wins $\mathcal{G}$ iff Player 2 wins $\mathcal{G}'$. Since $\mathcal{G}'$ is a reachability game, Player 2 has a memoryless winning strategy, which we use to construct a winning strategy for Player 2 in $\mathcal{G}$ with $\mathsf{sepwidth}(\alpha, w, t)$ memory states.

For every vertex $v \in V$, let $\pi_v : [|\mathsf{good}(v)|] \to \mathsf{good}(v)$ be an arbitrary function that induces an order on the sets in $\mathsf{good}(v)$. By Lemma 2, we have that $|\mathsf{good}(v)| \leq \mathsf{sepwidth}(\alpha, w, t)$. Indeed, $\mathsf{good}(v)$ is pairwise separated, thus its size is bounded by the maximal size of a pairwise separated subset of $\mathfrak{S}$, which is $\mathsf{sepwidth}(\alpha, w, t)$. Below we describe a winning strategy for Player 2 in $\mathcal{G}$ with memory states in $[\mathsf{sepwidth}(\alpha, w, t)]$. Intuitively, when the play is in a vertex v and a memory state $1 \leq i \leq \mathsf{sepwidth}(\alpha, w, t)$, the set of reachability objectives satisfied so far is some $S \subseteq \pi_v(i)$.

Recall that a play in G' follows the play in G while remembering the set S of reachability objectives satisfied so far. Thus, Player 2 wins from a vertex $\langle v, S\rangle$ in G' iff Player 2 S-wins from v in G. Consider a memoryless winning strategy f_2 for Player 2 in $\mathcal{G}'$. We use f_2 to define the following strategy f'_2 for Player 2 in $\mathcal{G}$. The strategy f'_2 relies on the memory structure $\mathcal{M} = \{\{1, \ldots, \mathsf{sepwidth}(\alpha, w, t)\}, i_0, \delta\}$, where the initial memory state is $i_0 = \min\{i \in [|\mathsf{good}(v_0)|] : \mathsf{obj}(v_0) \subseteq \pi_{v_0}(i)\}$, and $\delta(i, \langle v, u\rangle) = \min\{j \in [|\mathsf{good}(u)|] : \pi_v(i) \cup \mathsf{obj}(u) \subseteq \pi_u(j)\}$, for every $1 \leq i \leq \mathsf{sepwidth}(\alpha, w, t)$ and $\langle v, u\rangle \in E$. Then, for every $v \in V_2$ and $1 \leq i \leq |\mathsf{good}(v)|$, we define $f'_2(v, i) = f_2(\langle v, \pi_v(i)\rangle)$.

We show that f'_2 is a winning strategy for Player 2 in $\mathcal{G}$. The strategy f_2 is winning from every vertex $\langle v, S\rangle$ in G' such that $S \in \mathsf{good}(v)$, thus f'_2 is an $\pi_v(i)$-winning strategy from v while in a memory state $1 \leq i \leq |\mathsf{good}(v)|$. Since the play induced by f'_2 is in a vertex v and a memory state i when the set S of reachability objectives satisfied so far is such that $S \subseteq \pi_v(i)$, and every $\pi_v(i)$-winning strategy is also an S-winning strategy for Player 2 from v, we conclude that f'_2 is a winning strategy for Player 2 in $\mathcal{G}$. ☐

Theorem 10. *For every $m, t \in I\!N$ and a non-decreasing function $f : 2^{[m]} \to I\!N$, we can construct a MaxWR game $\mathcal{G}_{m,f,t} = \langle G_{m,f,t}, \langle \alpha, w, t\rangle\rangle$ such that all the following hold.*

1. $|\alpha| = m$.
2. $w(\{\alpha_i : i \in S\}) = f(S)$, *for every* $S \subseteq [m]$.
3. *Player 2 wins* $\mathcal{G}_{m,f,t}$, *yet every winning strategy for Player 2 requires memory* $\mathsf{sepwidth}(\alpha, w, t)$.

Proof. For every $m, t \in \mathbb{N}$ and a non-decreasing function $f : 2^{[m]} \to \mathbb{N}$, let $\mathfrak{P}$ be a pairwise separated subset of $\mathfrak{S}$ of maximal size. The MaxWR game $\mathcal{G}_{m,f,t}$ proceeds as follows (see exact definition in Appendix A.6): First, Player 1 chooses a set $S \in \mathfrak{P}$, then Player 2 chooses a set $S' \in \mathfrak{P}$, from which Player 1 can choose any max-t-short set Q such that $f(S' \cup Q) < t$. Then, the objective of Player 1 is to reach set vertices that form a set $S \subseteq [m]$ with $f(S) \geq t$.

As we formally prove in Appendix A.6, every winning strategy for Player 2 matches the choice of set in $\mathfrak{P}$ Player 1 makes from v_1, thus every winning strategy for Player 2 requires $|\mathfrak{P}|$ different memory states.

Consider a strategy f_2 for Player 2 in $\mathcal{G}_{m,f,t}$ such that there exist two different sets $S, S' \in \mathfrak{P}$ such that when Player 1 chooses S, Player 2 chooses S'. Recall that there exists a maximal set $Q \in \mathfrak{S}$ such that $S' \subseteq Q$ and $S \not\subseteq Q$. It is easy to see that in this case f_2 is not a winning strategy for Player 2. Indeed, when Player 1 chooses S and Q as described above, we have that the satisfaction value is $f(S \cup Q)$. Since $S \not\subseteq Q$ and Q is maximal, we have that $f(S \cup Q) \geq t$. □

We can now conclude with a tight bound for the memory required to Player 2.

Theorem 11. *A winning strategy for Player 2 in a MaxWR game with objective* $\langle \alpha, w, t \rangle$ *requires memory* $\mathsf{sepwidth}(\alpha, w, t)$.

4.2 Decidability

We continue to the complexity of deciding whether Player 1 wins a MaxWR game.

Theorem 12. *Deciding the winner in a MaxWR game is PSPACE-complete. Fixing the number of underlying objectives, the problem is in PTIME.*

Proof. (sketch, see full proof in Appendix A.7) For the upper bound, we describe an alternating Turing machine (ATM) T that runs in polynomial time, such that T accepts a MaxWR game $\mathcal{G}$ iff Player 1 wins in $\mathcal{G}$. The idea is similar to the upper-bound proof for deciding AllR games [21]. The ATM simulates the given MaxWR game $\mathcal{G} = \langle G, \langle \alpha, w, t \rangle \rangle$ for $|V| \cdot |\alpha|$ steps, and writes on the tape the set S of reachability objectives from α that are satisfied. After $|V| \cdot |\alpha|$ steps are completed, the ATM T calculates $w(S)$, proceeds to an accepting state if $w(S) \geq t$, and proceeds to a rejecting state otherwise.

The lower bound follows from the PSPACE-hardness of deciding whether Player 1 wins in AllR games [21].

When the number of underlying objectives is fixed, we have that the corresponding reachability game from the proof of Theorem 7 is of polynomial size. Thus, a MaxWR game can be decided in polynomial time by solving the corresponding reachability game. □

5 General Weight Functions

In this section we show that weighted multiple objectives with general (that is, not necessarily non-decreasing) weight functions are equivalent to *Muller* objectives, and we lift results known for Muller games to games with weighted multiple objectives.

Consider a game $G = \langle V_1, V_2, v_0, E\rangle$, and a finite set C of colors. A *Muller objective* is a pair $\langle \mathcal{F}, \chi\rangle$, where $\mathcal{F} \subseteq 2^C$ is a set of subsets of C, and $\chi : V \to 2^C$ is a coloring function that maps vertices to colors in C. We extend χ to sets of vertices in the expected way, thus $\chi(U) = \bigcup\{\chi(v) : v \in U\}$. We define the *size* of a Muller objective $\langle \mathcal{F}, \chi\rangle$ as $|\mathcal{F}|$, namely the number of sets in $\mathcal{F}$. Then, the size of a general weight function $w : 2^\alpha \to \mathbb{N}$ is $\sum_{S\in 2^\alpha} w(S)$.

Muller objectives have been studied mainly with a semantics that refers to the set of states visited in a play infinitely often. Here, we denote them as B-Muller. A play ρ satisfies the B-Muller objective $\langle \mathcal{F}, \chi\rangle$ iff $\chi(\mathit{inf}(\rho)) \in \mathcal{F}$. That is, if the set of colors of the vertices that appear infinitely often in ρ is a member of $\mathcal{F}$. For a semantics that refers to the set of states that the play reaches (a.k.a. weak-Muller games or Staiger-Wagner games [34, 25]), we have that ρ satisfies $\langle \mathcal{F}, \chi\rangle$ iff $\chi(\mathit{reach}(\rho)) \in \mathcal{F}$. We refer to this semantics as R-Muller objectives.

A *two-player B-Muller (or R-Muller) game* is a tuple $\mathcal{G} = \langle G, \psi\rangle$, where G is a two-player game graph, and $\psi = \langle \mathcal{F}, \chi\rangle$ is a B-Muller (or R-Muller) objective for Player 1.

Theorem 13 below shows that Muller games are reducible to weighted multiple objective games with general weight functions, and vice versa.

Theorem 13. *1. Every MaxWB (and MaxWR) objective $\psi = \langle \alpha, w, t\rangle$ with a general weight function has an equivalent B-Muller (R-Muller, respectively) objective of size $|\{S \subseteq \alpha : w(S) \geq t\}|$.*
2. Every B-Muller (and R-Muller) objective $\psi = \langle \mathcal{F}, \chi\rangle$ with a set of colors C has an equivalent MaxWB (MaxWR, respectively) objective with a general weight function of size $|\psi|$.

Proof. We start with the first claim. Let $\psi = \langle \alpha, w, t\rangle$. We define $\psi' = \langle \mathcal{F}, \chi\rangle$ with the set of colors α, where the Muller set $\mathcal{F} \subseteq 2^\alpha$ is such that $\mathcal{F} = \{S \subseteq \alpha : w(S) \geq t\}$, and the coloring function $\chi : V \to 2^\alpha$ is such that $\chi(v) = \mathsf{obj}(v)$ for every $v \in V$. It is easy to see that for every play ρ, we have that ρ satisfies ψ iff ρ satisfies ψ', and $|\psi'| = |\{S \subseteq \alpha : w(S) \geq t\}|$.

For the second claim, consider a set of colors C, and let $\psi = \langle \mathcal{F}, \chi\rangle$, with $\mathcal{F} \subseteq 2^C$. We define $\psi' = \langle \alpha, w, 1\rangle$, where $\alpha = \{\alpha_i\}_{i\in C}$, with $\alpha_i = \chi^{-1}(i)$, for every $i \in C$. Then, for every $S \in 2^\alpha$, we have that $w(S) = 1$ for every $S \in \mathcal{F}$, and $w(S) = 0$ for every $S \notin \mathcal{F}$. It is easy to see that for every play ρ, we have that ρ satisfies ψ iff ρ satisfies ψ', and since $|w| = |\{S \subseteq \alpha : w(S) = 1\}|$, we have that $|w| = |\psi|$. □

Since the problems of deciding B-Muller and R-Muller games are PAPCE-complete [26, 19, 34], we can conclude with the following.

Theorem 14. *Deciding the winner in MaxWB and MaxWR games with general weight functions is PSPACE-complete.*

As for the memory requirements for winning strategies, by Theorem 13, given a MaxWB or MaxWR objective ψ with a general weight function, the memory requirement for a winning strategy for Player 1 in a game with objective ψ coincides with the memory requirement for a winning strategy for Player 1 in the Muller game with an objective $\langle \mathcal{F}, \chi \rangle$ equivalent to ψ. For B-Muller games, the picture is well understood – the memory requirements are these required for the Zielonka tree $\mathcal{Z}_{\mathcal{F}}$. For R-Muller games, the memory requirements are exponential, and heuristics for their minimization have been studied in [25].

6 Discussion

We studied many aspects of weighted multi-objectives games. In the area of ω-regular games, researchers have extensively studied the expressive power and complexity of different types of Boolean objectives [28]. In the context of weighted multiple objectives, there are two parameters to the type of an objective: the class of the underlying objectives and the class of the weight function.

We focused on non-decreasing weight functions and showed that moving to general weight functions leads to the expressive power (and computational price) of Boolean Muller games. In the context of *bidding* in game theory, researchers have studied additional interesting classes of weight functions [35] (Chapter 11). We find it interesting to study restrictions on weight functions as a mean for defining objectives that are simpler than Muller, and are different from the simplification studied so far in the Boolean setting.

One can also study weighted multiple objective games in which the underlying objectives are stronger than Büchi and reachability, in particular games with underlying *parity* objectives. By [15], generalized parity games have the flavor of Streett games (in particular, their decidability is co-NP-complete). Also, it is not hard to see that Muller objectives can be translated to weighted multiple parity objectives with a non-decreasing weight function (as opposed to general weight functions, required in the translation of Muller to weighted multiple Büchi objectives). Thus, while weighted multiple parity objectives are of interest, the inherited complexity of parity dominates the complexity that has to do with the weights. Consequently, we find their study less interesting than that of multiple Büchi or reachability objectives.

References

1. S. Almagor, U. Boker, and O. Kupferman. Formalizing and reasoning about quality. *Journal of the ACM*, 63(3):24:1–24:56, 2016.
2. S. Almagor, D. Kuperberg, and O. Kupferman. Sensing as a complexity measure. *Int. J. Found. Comput. Sci.*, 30(6-7):831–873, 2019.
3. G. Avni, T.A. Henzinger, and V. Chonev. Infinite-duration bidding games. *Journal of the ACM*, 66(4):31:1–31:29, 2019.

4. R. Bloem, K. Chatterjee, and B. Jobstmann. Graph games and reactive synthesis. In *Handbook of Model Checking.*, pages 921–962. Springer, 2018.
5. A. Bohy, V. Bruyère, E. Filiot, and J-F. Raskin. Synthesis from LTL specifications with mean-payoff objectives. In *Proc. 19th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 7795 of *Lecture Notes in Computer Science*, pages 169–184. Springer, 2013.
6. U. Boker, K. Chatterjee, T.A. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. In *Proc. 26th IEEE Symp. on Logic in Computer Science*, pages 43–52, 2011.
7. P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Concurrent games with ordered objectives. In *Proc. 15th Int. Conf. on Foundations of Software Science and Computation Structures*, volume 7213 of *Lecture Notes in Computer Science*, pages 301–315. Springer, 2012.
8. P. Bouyer, Fijalkow N, M. Randour, and P. Vandenhove. How to play optimally for regular objectives?, 2023.
9. J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. Int. Congress on Logic, Method, and Philosophy of Science. 1960*, pages 1–12. Stanford University Press, 1962.
10. K. Chatterjee and L. Doyen. Energy and mean-payoff parity markov decision processes. In *36th Int. Symp. on Mathematical Foundations of Computer Science*, pages 206–218, 2011.
11. K. Chatterjee and L. Doyen. Energy parity games. *Theoretical Computer Science*, 458:49–60, 2012.
12. K. Chatterjee, L. Doyen, T.A. Henzinger, and J-F. Raskin. Generalized mean-payoff and energy games. In *Proc. 30th Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 8 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.
13. K. Chatterjee, W. Dvorak, M. Henzinger, and V. Loitzenbauer. Conditionally optimal algorithms for generalized büchi games. In *41st Int. Symp. on Mathematical Foundations of Computer Science*, volume 58 of *LIPIcs*, pages 25:1–25:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.
14. K. Chatterjee, T. A. Henzinger, and N. Piterman. Generalized parity games. In *Proc. 10th Int. Conf. on Foundations of Software Science and Computation Structures*, volume 4423 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2007.
15. K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. In *Proc. 18th Int. Conf. on Concurrency Theory*, pages 59–73, 2007.
16. K. Chatterjee, Z. Komárková, and J. Kretínský. Unifying two views on multiple mean-payoff objectives in markov decision processes. In *Proc. 30th IEEE Symp. on Logic in Computer Science*, pages 244–256, 2015.
17. K. Chatterjee, R. Majumdar, and T. A. Henzinger. Controller synthesis with budget constraints. In *Proc 11th International Workshop on Hybrid Systems: Computation and Control*, volume 4981 of *Lecture Notes in Computer Science*, pages 72–86. Springer, 2008.
18. L. de Alfaro, T.A. Henzinger, and R. Majumdar. Discounting the future in systems theory. In *Proc. 30th Int. Colloq. on Automata, Languages, and Programming*, volume 2719 of *Lecture Notes in Computer Science*, pages 1022–1037, 2003.
19. S. Dziembowski, M. Jurdzinski, and I. Walukiewicz. How much memory is needed to win infinite games. In *Proc. 12th IEEE Symp. on Logic in Computer Science*, pages 99–110, 1997.

20. E.A. Emerson and C. Jutla. The complexity of tree automata and logics of programs. In *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 328–337, 1988.
21. Nathanaël Fijalkow and Florian Horn. The surprizing complexity of reachability games. *CoRR*, abs/1010.2420, 2010.
22. A. Gurfinkel and M. Chechik. Multi-valued model-checking via classical model-checking. In *Proc. 14th Int. Conf. on Concurrency Theory*, pages 263–277. Springer-Verlag, 2003.
23. D. Harel and A. Pnueli. On the development of reactive systems. In K. Apt, editor, *Logics and Models of Concurrent Systems*, volume F-13 of *NATO Advanced Summer Institutes*, pages 477–498. Springer, 1985.
24. T.A. Henzinger. From Boolean to quantitative notions of correctness. In *Proc. 37th ACM Symp. on Principles of Programming Languages*, pages 157–158, 2010.
25. M. Holtmann. *Memory and delay in regular infinite games.* PhD thesis, RWTH Aachen University, 2011.
26. P. Hunter and A. Dawar. Complexity bounds for regular games. In *30th Int. Symp. on Mathematical Foundations of Computer Science*, volume 3618, pages 495–506. Springer, 2005.
27. N. Immerman. Number of quantifiers is better than number of tape cells. *Journal of Computer and Systems Science*, 22(3):384–406, 1981.
28. O. Kupferman. Automata theory and model checking. In *Handbook of Model Checking*, pages 107–151. Springer, 2018.
29. O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. *Annals of Mathematics and Artificial Intelligence*, 78(1):3–20, 2016.
30. O. Kupferman, N. Piterman, and M.Y. Vardi. Safraless compositional synthesis. In *Proc. 18th Int. Conf. on Computer Aided Verification*, volume 4144 of *Lecture Notes in Computer Science*, pages 31–44. Springer, 2006.
31. O. Kupferman and N. Shenwald. Games with trading of control. In *Proc. 34th Int. Conf. on Concurrency Theory*, volume 279 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.
32. O. Kupferman and T. Tamir. Alternating reachability games with behavioral and revenue objectives. In *Proc. 22nd Int. Conf. on Logic for Programming Artificial Intelligence and Reasoning*, 2018.
33. D.A. Martin. Borel determinacy. *Annals of Mathematics*, 65:363–371, 1975.
34. J. Neumann, A. Szepietowski, and I. Walukiewicz. Complexity of weak acceptance conditions in tree automata. *Inf. Process. Lett.*, 84(4):181–187, 2002.
35. N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. *Algorithmic Game Theory.* Cambridge University Press, 2007.
36. N. Piterman, A. Pnueli, and Y. Saar. Synthesis of reactive(1) designs. In *Proc. 7th Int. Conf. on Verification, Model Checking, and Abstract Interpretation*, volume 3855 of *Lecture Notes in Computer Science*, pages 364–380. Springer, 2006.
37. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th ACM Symp. on Principles of Programming Languages*, pages 179–190, 1989.
38. R. Rizzi and A.I. Tomescu. Faster fptases for counting and random generation of knapsack solutions. *Information and Computation*, 267:135–144, 2019.
39. W. Thomas. On the synthesis of strategies in infinite games. In *Proc. 12th Symp. on Theoretical Aspects of Computer Science*, volume 900 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1995.
40. M.Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and Systems Science*, 32(2):182–221, 1986.

41. Y. Velner, K. Chatterjee, L. Doyen, T.A. Henzinger, A. Rabinovich, and J-F. Raskin. The complexity of multi-mean-payoff and multi-energy games. *Information and Computation*, 241:177–196, 2015.

A Proofs

A.1 Missing details in the proof of Theorem 3

Formally, $G_{m,f,t} = \langle V_1, V_2, v_0, E\rangle$ is defined as follows (see $G_{4,f,2}$ in Figure 1, where f is additive and $f(\{i\}) = 1$ for every $i \in [4]$).

1. $V_1 = \mathcal{X} \cup [m]$. The vertices in $\mathcal{X}$ are called *set vertices*, and the vertices in $[m]$ are called *objective vertices*.
2. $V_2 = \{v_0\}$.
3. The set E contains the following edges.
   (a) $\langle v_0, S\rangle$, for every $S \in \mathcal{X}$. That is, in v_0 Player 2 chooses a set from $\mathcal{X}$.
   (b) $\langle S, i\rangle$, for every $S \in \mathcal{X}$, and $i \in S$. That is, Player 1 chooses $i \in S$.
   (c) $\langle i, v_0\rangle$, for every $i \in [m]$. That is, the game returns to v_0.

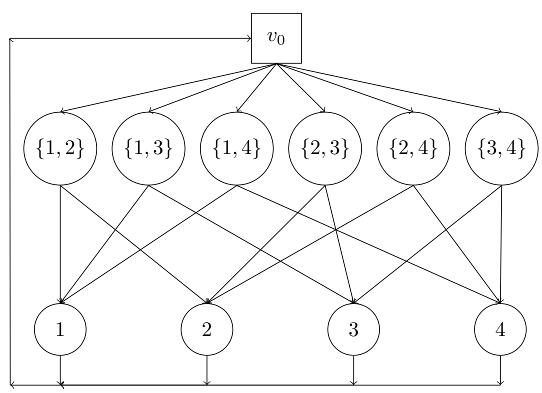

Fig. 1. The game graph $G_{4,f,2}$, where $f(i) = 1$ for all $i \in [4]$. The circles are vertices owned by Player 1, and the square v_0 is a vertex owned by Player 2.

We define $\mathcal{G}_{m,f,t} = \langle G_{m,f,t}, \langle \alpha, w, t\rangle\rangle$, with $\alpha_i = \{i\}$ and $w(\{\alpha_i : i \in S\}) = f(S)$, for every $S \subseteq [m]$.

A.2 Proof of Theorem 5

We prove the results for MinWB. The ones for MaxWB follow from determinacy of MinWB games.

We start with the upper bound. By Theorem 2, Player 1 wins a MinWB game iff she has a memoryless winning strategy. As we argue below, checking whether a given memoryless strategy for Player 1 is winning can be done in polynomial time, implying membership in NP.

Consider a MinWB game $\langle G, \langle \alpha, w, t\rangle\rangle$, with $\alpha = \{\alpha_1, \ldots, \alpha_m\}$. Let $G = \langle V_1, V_2, v_0, E\rangle$. For a memoryless strategy $f_1 : V_1 \to V$ for Player 1, let $G_{f_1} = \langle V, E_{f_1}\rangle$ be the sub-graph of G obtained by removing all edges not taken by f_1. Thus $\langle v, u\rangle \in E_{f_1}$ iff $f_1(v) = u$. Clearly, f_1 is winning iff all the paths from v_0 in G_{f_1} satisfy $\langle G, \langle \alpha, w, t\rangle\rangle$.

Note that there exists a path ρ in G_{f_1} such that $\mathsf{sat}(\rho, \alpha, w) \geq t$ iff there exists a non-trivial strongly connected component (SCC) C in G_{f_1} such that $w(\mathsf{obj}(C)) \geq t$. Accordingly, f_1 is a winning strategy for Player 1 iff $w(\mathsf{obj}(C)) < t$ for every non-trivial SCC C reachable from the initial vertex in G_{f_1}. Since the latter can be checked in polynomial time, we are done. Note that the check is polynomial also when the weight function is dualized, thus when $\tilde{w}$ is given. Indeed, we need to calculate $w(\mathsf{obj}(C))$ only for linearly many SCCs.

For the lower bound, we describe a reduction from SAT to MinB, which is a special case of MinWB with additive and uniform weight functions. That is, given a propositional formula φ in CNF, we construct a MinB game $\mathcal{G}_\varphi$ such that φ is satisfiable iff Player 1 wins $\mathcal{G}_\varphi$.

For a set of variables $X = \{x_1, \ldots, x_n\}$, let $\overline{X} = \{\overline{x_1}, \ldots, \overline{x_n}\}$. Consider a propositional formula φ given in CNF over $X \cup \overline{X}$. That is, $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_k$, for some $k \geq 1$, and for every $1 \leq i \leq k$, we have $C_i = (l_i^1 \vee l_i^2 \vee \cdots \vee l_i^{j_i})$, with $l_i^1, l_i^2, \ldots, l_i^{j_i} \in X \cup \overline{X}$. We assume that for every variable $x_i \in X$, the formula φ contains the clause $(x_i \vee \overline{x_i})$. Note that otherwise, we can add such clauses, maintaining the satisfiability of φ, and keeping the size of φ linear in its original size.

Intuitively, we define $\mathcal{G}_\varphi$ so that Player 2 can choose $1 \leq i \leq k$ and require Player 1 to prove that C_i is evaluated to **true** by asking Player 1 to choose a literal of C_i that is assigned **true**. Choosing a literal l involves a visit in a vertex associated with l. Accordingly, there exists a satisfying assignment to φ iff Player 1 can choose her responses so that at most n literals are visited infinitely often. In particular, the clauses of the form $(x_i \vee \overline{x_i})$ force Player 1 to commit on the truth value of all variables.

Formally, $\mathcal{G}_\varphi = \langle G_\varphi, \alpha, t\rangle$, where G_φ, α, and t are defined as follows.

1. The game graph $G_\varphi = \langle V_1, V_2, v_0, E\rangle$ has the following components (see example in Fig. 2).
   (a) $V_1 = \{C_1, \ldots, C_k\} \cup X \cup \overline{X}$. The vertices in $\{C_1, \ldots, C_k\}$ are *clause vertices*, and the vertices in $X \cup \overline{X}$ are *literal vertices*.
   (b) $V_2 = \{v_0\}$.
   (c) The set E of edges includes the following edges.
      i. $\langle v_0, C_i\rangle$, for every $i \in [k]$. By proceeding from v_0 to C_i, Player 2 requires Player 1 to prove that the clause C_i is evaluated to **true**.
      ii. $\langle C_i, l_i^h\rangle$, for every $i \in [k]$ and $1 \leq h \leq j_i$. By proceeding from C_i to l_i^h, Player 1 states that the literal l_i^h is evaluated to **true**.
      iii. $\langle l, v_0\rangle$, for every $l \in X \cup \overline{X}$.

2. The set of Büchi objectives is $\alpha = \{\{l\}\}_{l \in X \cup \overline{X}}$. That is, every literal vertex defines a singleton Büchi objective.
3. The threshold is $t = n$. That is, Player 1 aims for at most n literal vertices to be visited infinitely often.

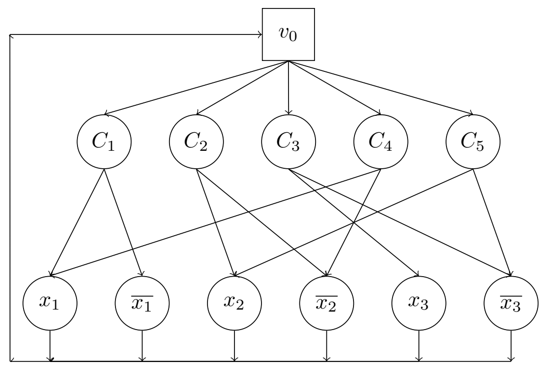

Fig. 2. The game graph G_φ for $\varphi = (x_1 \vee \overline{x_1}) \wedge (x_2 \vee \overline{x_2}) \wedge (x_3 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2}) \wedge (x_2 \vee \overline{x_3})$. The circles are vertices owned by Player 1, and the square v_0 is a vertex owned by Player 2.

We prove the correctness of the construction. Assume first that φ is satisfiable. Consider a satisfying assignment $\xi : X \to \{\mathbf{true}, \mathbf{false}\}$ to the variables in X, and consider the strategy f_1 for Player 1 that for every $i \in [k]$, goes from C_i to l_i^h such that l_i^h is evaluated to **true** in ξ.

We show that f_1 is a winning strategy for Player 1 in $\mathcal{G}_\varphi$. It is easy to see that for every strategy f_2 for Player 2 and $i \in [n]$, there is at most one set in $\{\{x_i\}, \{\overline{x_i}\}\}$ that is visited infinitely often. Indeed, since ξ is a satisfying assignment, for every clause C_i there exists a literal that is evaluated to **true** in ξ.

Assume now that φ is not satisfiable. Let f_2 be the strategy for Player 2 that repeatedly requires Player 1 to prove that the clause C_i is evaluated to **true** for every $i \in [k]$. Note that each such iteration visits $n+1$ different literal vertices. Indeed, Player 1 chooses an assignment ξ to the variables in X in the first n clause vertices, and since ξ does not satisfy φ, there exists a clause C_i for some $i > n$ such that every literal in C_i is evaluated to **false** by ξ. Hence, there are (at least) $n+1$ different literal vertices that are visited infinitely often.

We continue to the case in which the number of underlying objectives is fixed. By Theorem 1, for every MaxWB game $\langle G, \psi \rangle$ with $\psi = \langle \alpha, w, t \rangle$ there exists an AllB objective α' of size $\mathsf{width}(\alpha, w, t)$ such that Player 1 wins $\langle G, \psi \rangle$ iff Player 1 wins $\langle G, \alpha' \rangle$. If $|\alpha|$ is fixed, then $\mathsf{width}(\alpha, w, t)$ is fixed as well, thus deciding $\langle G, \psi \rangle$ can be done in polynomial time by calculating and solving the AllB game $\langle G, \alpha' \rangle$.

A.3 Proof of Theorem 7

Consider a MaxWR game $\mathcal{G} = \langle G, \langle \alpha, w, t \rangle \rangle$, with $\alpha = \{\alpha_1, \ldots, \alpha_m\}$. Assume that $w(\mathsf{obj}(v_0)) < t$ (otherwise, Player 1 wins in every play in G). We define the reachability game $\mathcal{G}' = \langle G', \alpha' \rangle$ as follows. The game graph G' contains a copy of the game graph G for every $S \in \mathfrak{S}$, and a vertex v_{sat}. Intuitively, a play in G' follows a play in G, maintaining in the $\mathfrak{S}$-component of its vertices the subset of α satisfied so far. When the corresponding play in G satisfies $\langle \alpha, w, t \rangle$, it moves to v_{sat}. Then, the objective of Player 1 in $\mathcal{G}'$ is to reach the vertex v_{sat}.

Formally, $\mathcal{G}' = \langle G', \{v_{sat}\} \rangle$, where the game graph $G' = \langle V_1', V_2', v_0', E' \rangle$ is defined as follows.

1. $V_1' = (V_1 \times \mathfrak{S}) \cup \{v_{sat}\}$.
2. $V_2' = V_2 \times \mathfrak{S}$.
3. $v_0' = \langle v_0, \mathsf{obj}(v_0) \rangle$.
4. The set E' contains the following edges.
   (a) $\langle \langle v, S \rangle, \langle u, S \cup \mathsf{obj}(u) \rangle \rangle$, for every edge $\langle v, u \rangle \in E$ and set $S \in \mathfrak{S}$ such that $S \cup \mathsf{obj}(u) \in \mathfrak{S}$.
   (b) $\langle \langle v, S \rangle, v_{sat} \rangle$, for every vertex $v \in V$ and set $S \in \mathfrak{S}$, such that $w(S \cup \mathsf{obj}(u)) \geq t$ for some edge $\langle v, u \rangle \in E$.
   (c) $\langle v_{sat}, v_{sat} \rangle$.

We show the equivalence of the conditions in the theorem.

1. $1 \Rightarrow 2$: Assume Player 1 has a winning strategy f_1 in $\mathcal{G}$, and consider the strategy f_1' for Player 1 in $\mathcal{G}'$ that agrees with f_1. By the definition of G' and since w is non-decreasing, a path ρ reaches the vertex v_{sat} iff the corresponding path in G satisfies $\langle \alpha, w, t \rangle$. Thus, since f_1 forces the satisfaction of $\langle \alpha, w, t \rangle$ in G, the strategy f_1' forces a visit in v_{sat} in G'.
2. $2 \Rightarrow 3$: Assume Player 1 wins in $\mathcal{G}'$. Every simple path from the initial vertex to v_{sat} in G' is of size of at most $|V| \cdot m$. Indeed, the set of satisfied reachability objectives in the corresponding play in G can only increase in size. Since $\mathcal{G}'$ is a reachability game, Player 1 has a memoryless winning strategy f_1, which induces a simple path from the initial vertex to v_{sat}.
3. $3 \Rightarrow 4$: Consider a memoryless winning strategy f_1 for Player 1 in $\mathcal{G}'$. Let f_1' be the strategy for Player 1 in $\mathcal{G}$ that agrees with f_1. That is, f_1' relies on the memory structure $\mathcal{M} = \langle \mathfrak{S}, \mathsf{obj}(v_0), \delta \rangle$, such that $\delta(S, \langle v, u \rangle) = S \cup \mathsf{obj}(u)$ if $S \cup \mathsf{obj}(u) \in \mathfrak{S}$, and $\delta(S, \langle v, u \rangle) = \emptyset$ if $S \in \mathfrak{S}$ and $S \cup \mathsf{obj}(u) \notin \mathfrak{S}$, for every $S \in \mathfrak{S}$ and $\langle v, u \rangle \in E$. Then, for every $v \in V_1$ and $S \in \mathfrak{S}$, we have that $f_1'(v, S) = f_1(\langle v, S \rangle)$. It is easy to see that, until the play in G satisfies $\langle \alpha, w, t \rangle$, being in a memory state S of f_1' corresponds to being in the copy S of G in G', in the sense that it indicates the set of reachability objectives satisfied so far. Hence, f_1' forces the satisfaction of $\langle \alpha, w, t \rangle$ within the same number of steps.
4. $4 \Rightarrow 1$: If Player 1 has a winning strategy in $\mathcal{G}$ then Player 1 wins in $\mathcal{G}$.

A.4 The game $G_{m,f,t}$ from the proof of Theorem 8

The game $G_{m,f,t} = \langle V_1, V_2, v_0, E\rangle$ is defined as follows (see Example 1).

1. $V_1 = \{v_1\} \cup \{\langle S, init\rangle, \langle S, final\rangle, \langle S, inter\rangle : S \in \mathfrak{S}\}$. The vertex v_1 is a *choice vertex*, and the vertices in $\{\langle S, init\rangle : S \in \mathfrak{S}\}$,$\{\langle S, final\rangle : S \in \mathfrak{S}\}$, and $\{\langle S, inter\rangle : S \in \mathfrak{S}\}$ are *init-set* vertices, *final set* vertices, and *inter-set* vertices, respectively.
2. $V_2 = \{v_0\} \cup \mathfrak{E}$. The vertex v_0 is the initial vertex, and the vertices in $\mathfrak{E}$ are *equivalence-class vertices*.
3. The set E contains the following edges.
   (a) $\langle v_0, \langle S, init\rangle\rangle$ and $\langle\langle S, init\rangle, v_1\rangle$, for every $S \in \mathfrak{S}$. That is, Player 2 chooses from v_0 a set $S \in \mathfrak{S}$, from which the game proceeds to v_1.
   (b) $\langle v_1, [S]\rangle$, for every $[S] \in \mathfrak{E}$. That is, Player 1 chooses from v_1 an equivalence class from $\mathfrak{E}$.
   (c) $\langle [S], \langle S', final\rangle\rangle$, for every $[S] \in \mathfrak{E}$ and $S' \in \mathfrak{S}$ such that $f(S \cup S') \geq t$, and $\langle [S], \langle S', inter\rangle\rangle$, for every $[S] \in \mathfrak{E}$ and $S' \in \mathfrak{S}$ such that $f(S \cup S') < t$ and $[S] \neq [S \cup S']$.
   (d) $\langle\langle S, final\rangle, \langle S, final\rangle\rangle$, for every $S \in \mathfrak{S}$.
   (e) $\langle\langle S, inter\rangle, v_1\rangle$, for every $S \in \mathfrak{S}$.

We then define $\mathcal{G}_{m,f,t} = \langle G_{m,f,t}, \langle \alpha, w, t\rangle\rangle$, where $\alpha_i = \{\langle S, init\rangle, \langle S, final\rangle, \langle S, inter\rangle : S \in \mathfrak{S}, i \in S\}$ and $w(\{\alpha_i : i \in S\}) = f(S)$, for every $S \subseteq [m]$. That is, the reachability objective α_i contains every set vertex that correspond to $S \in \mathfrak{S}$ such that $i \in S$.

Example 1. Figure 3 describes the game graph $G_{3,f,4}$, for the additive function f with $f(a) = f(b) = 2$ and $f(c) = 1$, and $t = 4$. The objective is $\langle \alpha, w, 4\rangle$ with $\alpha = \{\alpha_1, \alpha_2, \alpha_3\}$. The objective α_1 is the set of set-vertices that contain a, the objective α_2 is the set of set-vertices that contain b, and the objective α_3 is the set of set-vertices that contain 3. The weight function w is such that $w(\alpha_i) = f(i)$, for every $i \in [3]$. The set of 4-short sets is $\{\emptyset, \{a\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}\}$, and the equivalent sets are $\{a\}$ and $\{a, c\}$, $\{b\}$ and $\{b, c\}$, and $\emptyset$ and $\{c\}$. Thus there are three equivalence classes. The circles are vertices owned by Player 1, and the squares are vertices owned by Player 2.

A.5 Proof of Lemma 2

Consider a vertex $v \in V$, and two different sets $S, S' \in \mathsf{good}(v)$. Let f_2 be an S-winning strategy for Player 2 from v. Since S is maximal, f_2 is not an $(S \cup S')$-winning strategy for Player 2 from v, thus there exists a play ρ induced by f_2 such that $w(S \cup S' \cup \mathsf{sat}(\rho, \alpha)) \geq t$. Recall that $w(S \cup \mathsf{sat}(\rho, \alpha)) < t$, hence there exists a max-t-short set Q of α such that $S \cup \mathsf{sat}(\rho, \alpha) \subseteq Q$. We also have that $S' \nsubseteq Q$. Indeed, $w(S \cup S' \cup \mathsf{sat}(\rho, \alpha)) \geq t$, $w(Q) < t$, w is non-decreasing, and $S \cup \mathsf{sat}(\rho, \alpha) \subseteq Q$. Hence, Q is a max-t-short set such that $S \subseteq Q$ and $S' \nsubseteq Q$. In a similar way, there exists a max-t-short set Q' such that $S' \subseteq Q'$ and $S \nsubseteq Q'$. Therefore, S and S' are separated.

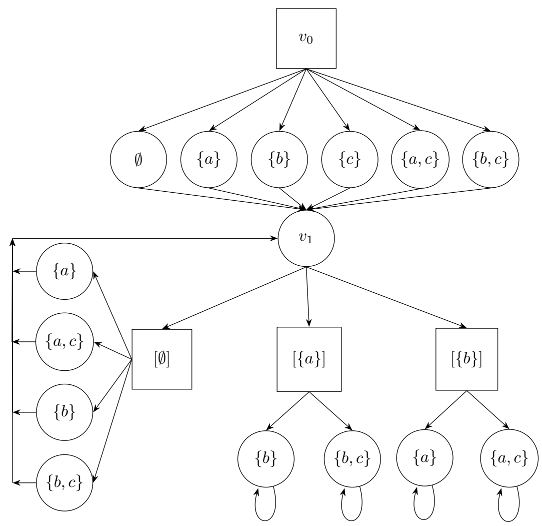

Fig. 3. The game graph $G_{3,f,4}$.

A.6 Missing details in the proof of Theorem 10

Formally, the game graph $G_{m,f,t} = \langle V_1, V_2, v_1, E\rangle$ is defined as follows.

1. $V_1 = \{v_1\} \cup (\mathfrak{P} \times \{1,2\}) \cup \{Q \in \mathfrak{S} : Q \text{ is maximal}\}$.
2. $V_2 = \{v_2\}$.
3. The set E contains the following edges.
   (a) $\langle v_1, \langle S,1\rangle\rangle$ and $\langle\langle S,1\rangle, v_2\rangle$, for every $S \in \mathfrak{P}$. That is, Player 1 chooses from v_1 a set $S \in \mathfrak{P}$, and then proceed to v_2.
   (b) $\langle v_2, \langle S,2\rangle\rangle$, for every $S \in \mathfrak{P}$. That is, Player 2 chooses from v_2 a set $S \in \mathfrak{P}$.
   (c) $\langle\langle S,2\rangle, Q\rangle$, for every $S \in \mathfrak{P}$ and a maximal set $Q \in \mathfrak{S}$ such that $S \subseteq Q$.

We define $\mathcal{G}_{m,f,t} = \langle G_{m,f,t}, \langle \alpha, w, t\rangle\rangle$, where $\alpha_i = \{S \in \mathfrak{P} \times \{1,2\} \cup \mathfrak{S} : i \in S\}$, and $w(\{\alpha_i : i \in S\}) = f(S)$, for every $S \subseteq [m]$. That is, the reachability set α_i contains every vertex that corresponds to a set $S \subseteq [m]$ such that $i \in S$. Thus, the objective of Player 1 is to reach set vertices that together form a set $S \subseteq [m]$ such that $f(S) \geq t$.

We prove that every winning strategy f_2 for Player 2 in $\mathcal{G}_{m,f,t}$ uses at least $|\mathfrak{P}|$ different memory states. Intuitively, we show that every winning strategy for Player 2 matches the choice of set in $\mathfrak{P}$ Player 1 makes from v_1.

Consider a strategy f_2 for Player 2 in $\mathcal{G}_{m,f,t}$. If the number of different memory states of f_2 is smaller than $|\mathfrak{P}|$, then there exist two different sets $S, S' \in \mathfrak{P}$ such that when Player 1 chooses $\langle S,1\rangle$ from v_1, Player 2 chooses $\langle S',2\rangle$ from v_2. Recall that there exists a maximal set $Q \in \mathfrak{S}$ such that $S' \subseteq Q$ and $S \not\subseteq Q$. It is easy to see that in this case f_2 is not a winning strategy for Player 2. Indeed, when Player 1 chooses $\langle S, 1\rangle$ and Q as described above, we have that the satisfaction value is $w(\{\alpha_i : i \in S \cup Q\}) = f(S \cup Q)$. Since $S \not\subseteq Q$ and Q is maximal, we have that $f(S \cup Q) \geq t$.

A.7 Proof of Theorem 12

For the upper bound, we describe an alternating Turing machine (ATM) T that runs in polynomial time, such that T accepts a MaxWR game $\mathcal{G}$ iff Player 1 wins in $\mathcal{G}$. The idea is similar to the upper bound proof for deciding AllR games [21].

An alternating Turing machine is a Turing machine whose states are partitioned into two sets: *existential* states and *universal* states. A configuration of T describes its state, the content of the working tape, and the location of the reading head. A configuration is existential (universal) if the state of T in the configuration is existential (universal, respectively). A run of T is a tree in which each node corresponds to a configuration of T: the root of the tree corresponds to the initial configuration; a node that corresponds to an existential configuration has a single successor, for one of the possible successor configurations; and a node that corresponds to a universal configuration has multiple successors, one for each possible successor configuration. The run is accepting iff all the branches of the tree reach an accepting configuration; that is, a configuration whose state is accepting.

In the membership problem, we get as input an ATM T and a word $x \in \Gamma^*$, and decide whether T accepts x; that is, if there exists an accepting run of T on x.

We now describe an ATM T that gets as input a description of a MaxWR game $\mathcal{G} = \langle G, \langle \alpha, w, t\rangle\rangle$, and accepts $\mathcal{G}$ iff Player 1 wins. By Theorem 7, Player 1 wins $\mathcal{G}$ iff she has a strategy that forces the satisfaction of $\langle \alpha, w, t\rangle$ within $|V| \times |\alpha|$ steps. Accordingly, T simulates the game for $|V| \cdot |\alpha|$ steps, where vertices of Player 1 correspond to existential states, and vertices of Player 2 correspond to universal states. The ATM writes on the tape the number c of steps in the game taken so far, the current vertex $v \in V$ in the game, and the set $S \subseteq \alpha$ of reachability objectives satisfied so far. As long as the number of steps taken is $c < |V| \cdot |\alpha|$, if the current vertex in the game is $v \in V_1$, the ATM T guesses a successor u of v in G, updates the current vertex in the game to be u, updates the set of reachability objectives to be $S \cup \mathsf{obj}(u)$, and updates the step counter to $c + 1$. If the current vertex of the game is $v \in V_2$, the ATM does the same, but for every successor u of v. When the step counter reaches $|V| \cdot |\alpha|$, the ATM T calculates $w(S)$, proceeds to an accepting state if $w(S) \geq t$, and proceeds to a rejecting state otherwise. Note that calculating $w(S)$ can be done in polynomial time also when the weight function is dualized, thus when $\tilde{w}$ is given.

The lower bound follows from the PSPACE-hardness of deciding whether Player 1 wins in AllR games [21].

We continue to the case in which the number of underlying objectives is fixed. By Theorem 7, for every MaxWR game $\mathcal{G} = \langle G, \psi\rangle$ with $\psi = \langle \alpha, w, t\rangle$ there exists a reachability game $\mathcal{G}'$ of size $|G| \cdot \mathsf{volume}(\alpha, w, t)$ such that Player 1 wins $\mathcal{G}$ iff Player 1 wins $\mathcal{G}'$. If $|\alpha|$ is fixed, then $\mathsf{volume}(\alpha, w, t)$ is fixed as well, thus deciding $\mathcal{G}$ can be done in polynomial time by calculating and solving the reachability game $\mathcal{G}'$.

B From MaxWB to Muller objectives

B.1 Muller objectives and Zielonka trees

Consider a finite set C of colors. A *Muller objective* is a pair $\langle \mathcal{F}, \chi \rangle$, where $\mathcal{F} \subseteq 2^C$ is a set of subsets of C, and $\chi : V \to C$ is a (partial) coloring function that maps vertices to colors in C. A play ρ satisfies the Muller objective $\langle \mathcal{F}, \chi \rangle$ iff $\{\chi(v) : v \in \mathit{inf}(\rho)\} \in \mathcal{F}$. That is, if the set of colors of the vertices that appear infinitely often in ρ is a member of $\mathcal{F}$. A *two-player Muller game* is a tuple $\mathcal{G} = \langle G, \psi \rangle$, where G is a two-player game graph, and $\psi = \langle \mathcal{F}, \chi \rangle$ is a Muller objective for Player 1.

Consider a set $\mathcal{F} \subseteq 2^C$. The *Zielonka tree for* $\mathcal{F}$, denoted $\mathcal{Z}_\mathcal{F}$, is a finite tree $\langle N, r, \text{child}, \tau \rangle$, where N is a set of nodes, $r \in N$ is a root node, $\text{child} : N \to 2^N$ maps nodes to their children nodes, and $\tau : N \to 2^C \setminus \{\emptyset\}$ maps nodes to nonempty sets of colors. The tree $\mathcal{Z}_\mathcal{F}$ is defined as follows.

1. $\tau(r) = C$.
2. Every node $n \in N$ has the following children.
   (a) If $\tau(n) \in \mathcal{F}$, then for every maximal nonempty subset $X \subset \tau(n)$ such that $X \notin \mathcal{F}$, there exists a child node $n' \in \text{child}(n)$ with $\tau(n') = X$.
   (b) If $\tau(n) \notin \mathcal{F}$, then for every maximal nonempty subset $X \subset \tau(n)$ such that $X \in \mathcal{F}$, there exists a child node $n' \in \text{child}(n)$ with $\tau(n') = X$.

That is, the root of the tree is labeled by the set C of all colors, and for every node n, if n is labeled by a member of $\mathcal{F}$, then the children of n are labeled by the maximal subsets of $\tau(n)$ that are not members of $\mathcal{F}$, and if n is labeled by a set that is not a member of $\mathcal{F}$, then the children of n are labeled by the maximal subsets of $\tau(n)$ that are members of $\mathcal{F}$.

In [19], the authors use the Zielonka tree for $\mathcal{F}$ in order to analyse the number of states required in a memory structure for winning strategies in a Muller game with objective $\mathcal{F}$. Formally, the *memory requirement* for a subtree with root $n \in N$, denoted by $\mathsf{mem}(n)$, is defined as follows.

1. If $\text{child}(n) = \emptyset$, then $\mathsf{mem}(n) = 1$.
2. If $\text{child}(n) \neq \emptyset$, then
   (a) If $\tau(n) \in \mathcal{F}$, then $\mathsf{mem}(n) = \sum_{n' \in \text{child}(n)} \mathsf{mem}(n')$.
   (b) If $\tau(n) \notin \mathcal{F}$, then $\mathsf{mem}(n) = \max\{\mathsf{mem}(n') : n' \in \text{child}(n)\}$.

Then, the memory requirement for the tree $\mathcal{Z}_\mathcal{F}$, denoted $\mathsf{mem}(\mathcal{F})$, is the memory requirement of the root of the tree.

Theorem 15. *(Theorem 6 in [19]) For every finite set of colors C, a Muller set $\mathcal{F} \subseteq 2^C$, a game graph G, and a coloring function $\chi : V \to C$ defined over the vertices in G, if Player 1 wins in the Muller game $\mathcal{G} = \langle G, \langle \mathcal{F}, \chi \rangle \rangle$, then she wins with memory $\mathsf{mem}(\mathcal{F})$.*

Theorem 16. *(Theorem 14 in [19]) For every finite set of colors C and a Muller set $\mathcal{F} \subseteq 2^C$, there exists a game graph G and a coloring function $\chi : V \to C$ defined over the vertices in G such that Player 1 wins in the a Muller game $\mathcal{G} = \langle G, \langle \mathcal{F}, \chi \rangle \rangle$, and every winning strategy for Player 1 needs memory of size at least* $\mathsf{mem}(\mathcal{F})$.

B.2 MaxWB as Muller

For the upper bound, we analyze the memory requirement of a Muller objective equivalent to a given MaxWB objective. Consider a MaxWB game $\mathcal{G} = \langle G, \psi \rangle$, where $\psi = \langle \alpha, w, t \rangle$. For every subset of vertices $U \subseteq V$, we denote by $\mathsf{obj}(U)$ the subset of objectives in α that intersect with U. That is, $\mathsf{obj}(U) = \{\alpha_i \in \alpha : U \cap \alpha_i \neq \emptyset\}$. The *Muller game equivalent to* $\mathcal{G}$ is $\mathcal{G}' = \langle G, \langle \mathcal{F}, \chi \rangle \rangle$ with the set of colors V, where the Muller set $\mathcal{F} \subseteq 2^V$ is such that $\mathcal{F} = \{U \subseteq V : w(\mathsf{obj}(U)) \geq t\}$, and the coloring function $\chi : V \to V$ is the identity function, hence $\chi(v) = v$ for every $v \in V$. It is easy to see that for every play ρ in G we have that ρ satisfies ψ iff ρ satisfies $\langle \mathcal{F}, \chi \rangle$.

Theorem 17. *For every MaxWB game $\mathcal{G} = \langle G, \alpha, w, t \rangle$, if Player 1 wins in $\mathcal{G}$, then she wins with memory of size equal to the number of maximal subsets $S \subset \alpha$ with $w(S) < t$.*

Proof. Consider a MaxWB game $\mathcal{G} = \langle G, \alpha, w, t \rangle$. We analyze the Zielonka tree for the Muller set $\mathcal{F}$ in the equivalent Muller game $\mathcal{G}' = \langle G, \langle \mathcal{F}, \chi \rangle \rangle$, and show that $\mathsf{mem}(\mathcal{F})$ is bounded by the number of maximal subsets $S \subset \alpha$ with $w(S) < t$. By Theorem 15, it then follows that if Player 1 wins in $\mathcal{G}$, then she has a winning strategy with memory bounded by the number of maximal subsets $S \subset \alpha$ with $w(S) < t$.

The Zielonka tree $\mathcal{Z} = \langle N, r, \text{child}, \tau \rangle$ for $\mathcal{F}$ is defined as follows. The root of the tree r is labeled by V, thus a member of $\mathcal{F}$. Since the root is a member of $\mathcal{F}$, it has a child node $n \in \text{child}(r)$ for every maximal subset $U \subset V$ that is not a member of $\mathcal{F}$. Equivalently, the root has a child node for every maximal subset $U \subset V$ such that $w(\mathsf{obj}(U)) < t$. Finally note that for every $U \subset V$ with $w(\mathsf{obj}(U)) < t$, and for every $U' \subset U$, we have that $w(\mathsf{obj}(U') \leq w(\mathsf{obj}(U))) < t$, hence $U' \notin \mathcal{F}$. Therefore, the children of the root of $\mathcal{Z}_{\mathcal{F}}$ have no children of their own.

We continue to calculate $\mathsf{mem}(\mathcal{F})$. Since $\tau(r) \in \mathcal{F}$, we have that $\mathsf{mem}(\mathcal{F}) = \mathsf{mem}(r) = \sum_{n \in \text{child}(r)} \mathsf{mem}(n)$. For every child node $n \in \text{child}(r)$ we have that $\text{child}(n) = \emptyset$, hence $\mathsf{mem}(n) = 1$. Therefore, $\mathsf{mem}(\mathcal{F}) = |\text{child}(r)|$.

It is then remained to show that the number of maximal subsets $U \subset V$ with $w(\mathsf{obj}(U)) < t$ is smaller or equal to the number of maximal subsets $S \subset \alpha$ with $w(S) < t$. Note that it is enough to show that for every maximal subset of objectives $S \subset \alpha$ with $w(S) < t$, there exists at most one maximal subset of vertices $U \subset V$ with $w(\mathsf{obj}(U)) < t$ such that $\mathsf{obj}(U) \subseteq S$:

Consider a maximal subset of objectives $S \subset \alpha$ with $w(S) < t$, and two different subsets of vertices $X, Y \subset V$ such that $\mathsf{obj}(X), \mathsf{obj}(Y) \subseteq S$. It is easy to see that at least one of them is not maximal. Indeed, X and Y are different, hence at least one of them is a strict subset of $X \cup Y$. As $w(\mathsf{obj}(X \cup Y)) \leq w(S) < t$, we have that at least one of X, Y is not a maximal subset of V with a weight below the threshold t. ◻

For the lower bound, we show that for every $m \in \mathbb{N}$, a weight function $w : [m] \to \mathbb{N}$, and a threshold $t \leq w([m])$, there exists a MaxWB game $\mathcal{G} = \langle G, \alpha, w, t\rangle$ with m objectives such that (with abuse of notations) $w(\alpha_i) = w(i)$ for every $i \in [m]$, Player 1 wins in $\mathcal{G}$, and the number of memory states in every winning strategy for Player 1 is at least the number of maximal subsets $S \subset [m]$ with $w(S) < t$, which equals to the number of maximal subsets $S \subset \alpha$ with $w(S) < t$.

For every $m \in \mathbb{N}$, a weight function $w : [m] \to \mathbb{N}$, and a threshold $t \leq w([m])$, we define the Muller set $\mathcal{F}_{m,w,t} = \{S \subseteq [m] : w(S) \geq t\}$. Then, for every Muller game $\mathcal{G} = \langle G, \mathcal{F}_{m,w,t}, \chi\rangle$, the *equivalent MaxWB game* is $\mathcal{G}' = \langle G, \alpha, w, t\rangle$ such that $\alpha_i = \chi^{-1}(i)$ for every $i \in [m]$, and $w(\alpha_i) = w(i)$ for every $i \in [m]$. It is easy to see that for every play ρ in G we have that ρ satisfies $\langle \mathcal{F}_{m,w,t}, \chi\rangle$ iff ρ satisfies $\langle \alpha, w, t\rangle$.

Theorem 18. *For every $m \in \mathbb{N}$, a weight function $w : [m] \to \mathbb{N}$, and a threshold $t \leq w([m])$, there exists a MaxWB game $\mathcal{G} = \langle G, \alpha, w, t\rangle$ with m objectives such that $w(\alpha_i) = w(i)$ for every $i \in [m]$, Player 1 wins in $\mathcal{G}$, and and the number of memory states in every winning strategy for Player 1 is at least the number of maximal subsets $S \subset \alpha$ with $w(S) < t$.*

Proof. Consider $m \in \mathbb{N}$, a weight function $w : [m] \to \mathbb{N}$, and a threshold $t \leq w([m])$. By Theorem 16, there exists a game graph G and a coloring function $\chi : V \to [m]$ defined over the vertices in G such that Player 1 wins in the Muller game $\mathcal{G} = \langle G, \mathcal{F}_{m,w,t}, \chi\rangle$, and every winning strategy for Player 1 in $\mathcal{G}$ requires at least $\mathsf{mem}(\mathcal{F}_{m,w,t})$ memory states. Hence, we also have that Player 1 wins in the equivalent MaxWB game $\mathcal{G}' = \langle G, \psi\rangle$, and every winning strategy for Player 1 in $\mathcal{G}$ requires at least $\mathsf{mem}(\mathcal{F}_{m,w,t})$ memory states. Therefore, all is left to show is that $\mathsf{mem}(\mathcal{F}_{m,w,t})$ is the number of maximal subsets $S \subset [m]$ with $w(S) < t$.

In a similar way to the analysis of the Zielonka tree in the proof of Theorem 17, we have that $\mathsf{mem}(\mathcal{F}_{m,w,t})$ equals to the number of children nodes of the root in the Zielonka tree $\mathcal{Z}_{\mathcal{F}_{m,w,t}}$. The root is labeled by $[m]$, hence the number of children of the root is the number of maximal subsets of $[m]$ that are not members of $\mathcal{F}_{m,w,t}$. Equivalently, the number of maximal subsets $S \subset [m]$ such that $w(S) < t$. ◻

Corollary 1. *A winning strategy for Player 1 in a MaxWB game $\mathcal{G} = \langle G, \alpha, w, t\rangle$ requires memory of size equal to the number of maximal subsets $S \subset \alpha$ with $w(S) < t$.*

In the special case of MaxB games, we have that the maximal subsets S of α with $w(S) < t$ are the subsets of α of size $t - 1$, hence the following holds.

Corollary 2. *A winning strategy for Player 1 in a MaxB game* $\mathcal{G} = \langle G, \alpha, t \rangle$ *with* m *Büchi objectives requires memory of size* $\binom{m}{t-1}$.