Non-Zero-Sum Games with Multiple Weighted Objectives

Yoav Feinstein[1[x]], Orna Kupferman[1[0000-0003-4699-6117]], and Noam Shenwald[1[0000-0003-1994-6835]]

The Hebrew University of Jerusalem, Israel

Abstract. We introduce and study *non-zero-sum multi-player games* with *weighted multiple objectives*. In these games, the objective of each player consists of a set α of underlying objectives and a weight function $w : 2^\alpha \to \mathbb{Z}$ that maps each subset X of α to the utility of the player when exactly all the objectives in X are satisfied.

We study the existence and synthesis of *stable outcomes* with *desired utilities* for the players. The problem generalizes rational synthesis and enables the synthesis of outcomes that satisfy wellness, fairness, and priority requirements.

We study the extension of the game by *payments*, with which players can incentivize each other to follow strategies that are beneficial for the paying player. We show how such payments can be used in order to *repair* systems.

We study the complexity of the setting for various classes of weight functions. In particular, general weight functions are related to Muller objectives, and the synthesis problem for them is PSPACE-complete. We study non-decreasing, additive, positive, and other classes of weight functions, and the way they affect the memory required for the players and the complexity of the synthesis problem.

1 Introduction

Synthesis is the automated construction of systems from their specifications [36]. Modern systems often consist of interacting components. The interaction is modeled by a *multi-player game* played on a finite graph. In the *turn-based* setting, the vertices of the game graph are partitioned among the players. A token is placed on an initial vertex, and in each turn, the player that owns the vertex with the token moves it to a successor vertex. Each player has a *strategy* that directs her how to move the token when it reaches vertices she own. A *profile* is a vector of strategies, one for each player. The outcome of profile is a *play* – an infinite path in the game graph, obtained when the players follow their strategies. The goal of each player is to direct the game into an outcome that is optimal from her point of view.

The simplest games are *Boolean zero-sum games*: the players compete with each other on the satisfaction of contradicting Boolean objectives. In particular, two-player zero-sum games model the interaction between a system that aims to satisfy a given specification and an environment that tries to violate the specification [8]. Researchers have studied several extensions of Boolean zero-sum games. One is to *quantitative* zero-sum games. There, the objectives or the game graph are multi-valued, and we seek strategies that satisfy the objectives in the highest possible value, maximize rewards, or minimize costs, possibly in a stochastic manner [7, 22, 2, 40, 3, 14, 13]. Another extension is to Boolean *non-zero-sum* games, namely when the objectives of the players may overlap [16, 38]. There, typical questions concern the stability of the game and the equilibria the players may reach [41]. The most common criterion for stability is the existence of a *Nash equilibrium* (NE) [32]. A profile of strategies is an NE if no (single) player can benefit from unilaterally changing her strategy. In particular, in *rational synthesis*, we seek an equilibrium in which the objectives of the players that model the system are satisfied [20, 17].

The two extensions above have been merged in *non-zero-sum games with quantitative objectives*. Specifically, in [26, 4, 9], the authors add a quantitative layer to LTL and studied rational synthesis for multi-valued extensions of LTL, and in [39, 12], the authors study equilibria in weighted games. The closest to our contribution here are non-zero-sum games with multiple ω-regular objectives. In particular, [35, 10] study games in which the underlying objectives are *ordered*, specifying priorities on different objectives.

We introduce and study non-zero-sum multi-player games with *weighted multiple objectives*. Consider a game graph with vertices in V. An objective in our game is specified by a tuple $\langle \alpha, w \rangle$ where $\alpha = \{\alpha_1, \ldots, \alpha_m\} \subseteq 2^V$, is a set of underlying *Büchi* objectives, and $w : 2^\alpha \to \mathbb{Z}$ is a weight function that maps each subset X of α to the *utility* obtained when exactly all the objectives in X are satisfied. More formally, for a play ρ, let $X \subseteq \alpha$ be the set of objectives in α that ρ satisfies; that is, ρ visits infinitely often exactly all the sets in X. Then, the satisfaction value of the objective $\langle \alpha, w \rangle$ in the play ρ is $w(X)$. We view $\langle \alpha, w \rangle$ as a maximization objective, and refer to games with weighted multiple Büchi objectives as *MaxWB* games. Note that each player has her own objective. Thus, a k-player MaxWB game is $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i \in [k]} \rangle$, where for all $i \in [k] = \{1, \ldots, k\}$, we have that w_i is a weight function for Player i.

Weighted objectives enable the user to conveniently 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_i, Player i can express the utility of different combinations of grants, storage options, locations, and more. Negative weights can be used to specify behaviors that are not desired, or assume-guarantee specifications. As a concrete example, consider robots that patrol a warehouse. The robots are operated by different companies, each having its own objectives. The setting can be modeled by a game graph whose vertices correspond to locations in the warehouse. Each robot is assigned with missions that combine the retrieval of items from appropriate shelves and logistical tasks such as repeatedly visiting or avoiding certain charging stations or areas covered by security cameras. Different sets of locations within the warehouse are associated with varying rewards and costs, reflecting diverse priorities related to requested items, specific shelves for retrieval, balance of the traffic in the different zones, and more logistical considerations for each robot.

In [28], the authors study zero-sum games with weighted multiple objectives. There, each objective also includes a threshold $t \in \mathbb{Z}$, and a player satisfies a *Boolean MaxWB* (BMaxWB) objective $\langle \alpha, w, t \rangle$ if the weight of the satisfied objectives is at least t. The authors solve the problem of deciding the game and analyze its tight complexity and memory requirements for each of the players. General weight functions enable BMaxWB objectives to express all *Muller* objectives [31, 18], which specify the exact set of vertices that repeat infinitely often in a play. By giving a positive weight to such sets and negative ones to the other sets, the translation from BMaxWB objectives to Muller objectives is straightforward, and implies that BMaxWB games can be decided in PSPACE [33, 23]. The connection to Muller games also implies a matching lower bound. The study in [28] then focuses on zero-sum BMaxWB games with restricted weight functions, for which the complexity is lower.

We consider a setting in which the set $[k]$ of players is partitioned to a set S of system players, whose behavior is controllable, and the set $[k] \setminus S$ of environment players, which are rational and aim for maximizing their utility. The main problem we consider, termed *Desired NE* (DNE, for short), is the existence and finding of stable profiles that satisfy some desired properties. Formally, a profile is stable if it is an S-fixed NE: no player in $[k] \setminus S$ has a beneficial deviation. The DNE problem gets as input a k-player MaxWB game $\mathcal{G}$, a set $S \subseteq [k]$ of system players, and a predicate P describing desired utilities for the players. It returns an S-fixed NE π in $\mathcal{G}$ such that the utilities of the players in π satisfy P. The predicate P is given by a Boolean assertion with atoms that refer to the utilities of the players. For example, *cooperative rational synthesis* [20] is a special case of DNE in which P sets lower bounds on the utilities of the system players. The predicate P can also prioritize the players or specify wellness or fairness goals [34].

We start with the case $\mathcal{G}$ has general weight functions and show that the PSPACE complexity of zero-sum BMaxWB games is carried over to non-zero-sum MaxWB games. The ordered objectives studied in [35] are sets in a Muller objective, and so our results apply also to the setting studied there, where the tight complexity of finding an NE (a special case of DNE, with $S = \emptyset$ and $P =$ **true**) was left open. We then consider several restrictions on the weight function and the way they influence the expressive power of the weight functions and the complexity of the DNE problem. We consider the following restrictions: (1) *Positive* weight functions correspond to settings where players are only awarded for satisfying objectives, thus $w(X) \geq 0$ for all $X \subseteq \alpha$. (2) *Non-decreasing* weight functions correspond to settings with *free disposal* [34], thus for all sets $X, X' \subseteq \alpha$, if $X \subseteq X'$, then $w(X) \leq w(X')$. (3) *Additive* weight functions correspond to settings where the objectives are independent of each other, thus the weight of a set is the sum of the weights of its elements. Accordingly, an additive weight function is given by $w : \alpha \to \mathbb{Z}$, and for every $X \subseteq \alpha$, we have that $w(X)$ is the sum of the weights of the objectives in X.

Studying the DNE problem for the various types of weight functions, we show that a key factor in the complexity is the existence of a memoryless strategy for the minimizing player in the corresponding zero-sum BMaxWB game. In particular, games with a non-decreasing weight functions enjoy this property, and we show that the DNE problem for them is NP-complete. Also, while games with an additive weight function are PSPACE-complete, games with an additive weight function that is *almost positive*, namely when only one Büchi objective may not be positive, also satisfy the above property, making the DNE problem for them easier. The result is tight, in the sense that we cannot extend it to games in which a fixed number of Büchi objectives is non-positive. Our study completes the picture known for zero-sum BMaxWB games. In particular, the study in [28] concerns only positive additive weight functions, and we show that, surprisingly, (non-positive) additive weight functions can express all Muller objectives. The best translation between the two formalisms, however, is exponential in both directions. Back to non-zero-sum games, we also show that additive weight functions can be manipulated so that for every partition (S, E) of $[k]$, we can make the game (S, E)-polar, in the sense that all the players in S agree on the polarity (i.e., whether it is positive or negative) of each of the underlying objectives, which is dual to the polarity of the objective for the players in E. Thus, the relative weights of the different objectives play a role that is more significant than their polarity.

We demonstrate the application of MaxWB games by introducing *multi-player games with payments*, where players can incentivize each other to follow strategies that are beneficial for the paying player. Consider for example a supercomputer that gets calculation requests from users. Payments from the users can be used in order to incentivize the supercomputer to perform certain calculations. For simplicity, assume that each player $i \in [k]$ has a single Büchi objective α_i, with a weight $R_i \in \mathbb{N}$, awarded in case α_i is satisfied. Then, a game with payments includes a *payment function* $p : [k] \times [k] \to \mathbb{N}$ that maps each two players $i, j \in [k]$ to the amount Player i commits to pay Player j when α_i is satisfied. It is not hard to see that a game with payments can be reduced to a MaxWB game with almost-positive weight functions. Indeed, the weight for each Büchi objective now takes into account both the reward and the payments to and from the other players. For general MaxWB games with payments, we suggest richer payment functions, and a reduction to MaxWB games is possible too. Beyond the application of MaxWB games for synthesizing strategies in games with payments, we study the *monetary-based repair* of systems, where we synthesize a payment function with which a desired stable outcome exists. Thus, unlike earlier work, where repairs are based on controlling the players [24, 21] or manipulating the objectives [1], our solution adds monetary incentives to the picture, which reflects the way equilibria are often achieved in real life.

2 Preliminaries

2.1 Multi-player games

For $k \geq 1$, let $[k] = \{1, \ldots, k\}$. A *k-player game graph* is $G = \langle \{V_i\}_{i \in [k]}, v_0, E \rangle$, where $\{V_i\}_{i \in [k]}$ are disjoint sets of vertices, each owned by a different player, and we let $V = \bigcup_{i \in [k]} V_i$. 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 at least one $u \in V$ such that $\langle v, u \rangle \in E$.

In the beginning of a play in the game, a token is placed on v_0. The players control the movement of the token in vertices they own: In each turn in the game, the player that owns the vertex with the token chooses a successor vertex and moves the token to it. Together, the players generate a *play* – an infinite path in G. Formally, a *strategy* for Player i is a function $f_i : V^* \cdot V_i \to V$ that directs her how to move the token in vertices she owns. Thus, f_i maps prefixes of plays to possible extensions in a way that respects E: for every $\rho \cdot v$ with $\rho \in V^*$ and $v \in V_i$, we have that $\langle v, f_i(\rho \cdot v)\rangle \in E$. The strategy f_i is *memoryless* if it depends only on the current vertex visited, in which case we describe it by a function $f_i : V_i \to V$, and is *finite-memory* if it is possible to replace the unbounded histories in $V^* \cdot V_i$ by a finite number of memories. A *profile* is a tuple $\pi = \langle f_1, \ldots, f_k \rangle$ of strategies, one for each player. The *outcome* of a profile $\pi = \langle f_1, \ldots, f_k \rangle$ is the play obtained when the players follow their strategies. Thus, $\mathsf{Outcome}(\pi) = v_0, v_1, v_2, \ldots$ is such that for all $j \geq 0$, we have that $v_{j+1} = f_i(v_0, v_1, \ldots, v_j)$, where $i \in [k]$ is such that $v_j \in V_i$.

A *zero-sum two-player game* is $\mathcal{G} = \langle G, \psi \rangle$, where $G = \langle V_1, V_2, v_0, E \rangle$ is a 2-player game graph and $\psi \subseteq V^\omega$ is an objective for Player 1, describing the set of outcomes in which Player 1 wins. The objective of Player 2 complements the one of Player 1, thus Player 2 wins when the outcome is not in ψ. A strategy f_1 is a *winning strategy* for Player 1 if for every strategy f_2 for Player 2, we have that $\mathsf{Outcome}(\langle f_1, f_2 \rangle)$ 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 $\mathsf{Outcome}(\langle f_1, f_2 \rangle)$ does not satisfy ψ. We say that Player i *wins* in $\mathcal{G}$ if she has a winning strategy.

Multi-player games may be *non zero-sum*, thus the objectives of the players may overlap. There, we consider *quantitative objectives* for the players and *stable* profiles. Formally, for $k \geq 1$, a *k-player game* is a pair $\mathcal{G} = \langle G, \{\psi_i\}_{i \in [k]} \rangle$, where G is a k-player game graph, and for every $i \in [k]$, we have that $\psi_i : V^\omega \to \mathbb{Z}$ maps each play ρ in G to the (possibly negative) *utility* of Player i when the outcome is ρ. Formally, the *utility* of Player i in a play ρ, denoted $\mathsf{util}_i(\rho)$, is $\psi_i(\rho)$. Then, for a profile π, the utility of Player i in π, denoted $\mathsf{util}_i(\pi)$, is $\mathsf{util}_i(\mathsf{Outcome}(\pi))$.

A profile $\pi = \langle f_1, \ldots, f_k \rangle$ is a *Nash Equilibrium* (NE, for short) [32] if no player can benefit from unilaterally changing her strategy. Formally, for $i \in [k]$ and a strategy f'_i for Player i, let $\pi[i \leftarrow f'_i] = \langle f_1, \ldots, f_{i-1}, f'_i, f_{i+1}, \ldots, f_k \rangle$ be the profile in which Player i *deviates* to the strategy f'_i. We say that π is an NE if for every $i \in [k]$ and strategy f'_i, we have that $\mathsf{util}_i(\pi[i \leftarrow f'_i]) \leq \mathsf{util}_i(\pi)$.

We say that two profiles π and π' are equivalent iff for all $i \in [k]$, we have that $\mathsf{util}_i(\pi) = \mathsf{util}_i(\pi')$. For two sets of objectives $\{\psi_i\}_{i \in [k]}$ and $\{\psi'_i\}_{i \in [k]}$ over the same set V of vertices, we say that $\{\psi_i\}_{i \in [k]}$ and $\{\psi'_i\}_{i \in [k]}$ are *equivalent* if for every play $\rho \in V^\omega$ and $i \in [k]$, we have that $\psi_i(\rho) = \psi'_i(\rho)$. Then, two games $\mathcal{G} = \langle G, \{\psi'_i\}_{i \in [k]} \rangle$ and $\mathcal{G}' = \langle G, \{\psi'_i\}_{i \in [k]} \rangle$ over the same game graph are equivalent iff $\{\psi_i\}_{i \in [k]}$ and $\{\psi'_i\}_{i \in [k]}$ are equivalent.

2.2 Weighted multiple objectives

In the definitions above, we used $\psi \subseteq V^\omega$ and $\psi_i : V^\omega \to \mathbb{Z}$ to denote Boolean and quantitative objectives. We now define *weighted multiple objectives*, which specify ψ and ψ_i succinctly.

For a play $\rho = v_0, v_1, \ldots$, let $\mathit{reach}(\rho)$ denote the set of vertices visited along ρ and $\mathit{inf}(\rho)$ denote the set of vertices visited infinitely often along ρ. That is, $\mathit{reach}(\rho) = \{v \in V : \text{there is } i \geq 0 \text{ such that } v_i = v\}$, and $\mathit{inf}(\rho) = \{v \in V : \text{there are infinitely many } i \geq 0 \text{ such that } v_i = v\}$. For a set of vertices $\alpha \subseteq V$, a play ρ satisfies the *Büchi* objective α iff $\mathit{inf}(\rho) \cap \alpha \neq \emptyset$.

A *weighted Büchi objective* is a pair $\langle \alpha, w \rangle$, where $\alpha = \{\alpha_1, \ldots, \alpha_m\}$ is a set of m Büchi objectives and $w : 2^\alpha \to \mathbb{Z}$ is a weight function that maps subsets of objectives in α to integer numbers.[1] We assume that $w(\emptyset) = 0$. We say that w is *positive* if for all $X \subseteq \alpha$, we have that $w(X) \geq 0$. We say that w is *non-decreasing* if for every two sets $X, X' \subseteq \alpha$, if $X \subseteq X'$, then $w(X) \leq w(X')$. In the context of game theory, non-decreasing weight functions are very useful, as they correspond to settings with *free disposal*, namely when satisfaction of additional objectives does not decrease the utility [34]. Note that since $w(\emptyset) = 0$, a non-decreasing weight function is positive.

A weight function is *additive* if for every set $X \subseteq \alpha$, the weight of X equals to the sum of weights of the singleton subsets that constitute X. That is, $w(X) = \sum_{\alpha_l \in X} w(\{\alpha_l\})$. An additive weight function is thus given by $w : \alpha \to \mathbb{Z}$, and is extended to sets of objectives in the expected way, thus for every $X \subseteq \alpha$, we have that $w(X) = \sum_{\alpha_l \in X} w(\alpha_l)$.

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 is then the weight of the set of objectives in α that are satisfied in ρ, namely $w(\mathsf{sat}(\rho, \alpha))$. Since we view an objective $\langle \alpha, w \rangle$ as a maximization objective, we refer to games with weighted Büchi objectives as *MaxWB* games. We assume that the objectives of all the players in the game are defined over the same set of underlying objectives. Thus, a MaxWB game is $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i \in [k]} \rangle$, where for all $i \in [k]$, we have that $w_i : 2^\alpha \to \mathbb{Z}$ is a weight function for Player i. Then, the utility of Player i in a play ρ is $w_i(\mathsf{sat}(\rho, \alpha))$. Note that since the utilities for the players depend on the set of vertices that appear infinitely often in a play, if π and π' are profiles such that $\mathit{inf}(\mathsf{Outcome}(\pi)) = \mathit{inf}(\mathsf{Outcome}(\pi'))$, then π and π' are equivalent.

A well-studied special case of MaxWB games is when the players have underlying Büchi objectives. There, each player has a single Büchi objective she wishes to satisfy. Formally, for all $i \in [k]$, there is $j \in [m]$ such that $w_i(\alpha_j) > 0$, and for all $l \in [m] \setminus \{j\}$, we have $w_i(\alpha_l) = 0$. We describe Büchi games by $\mathcal{G} = \langle G, \{\langle \alpha_i, R_i \rangle\}_{i \in [k]} \rangle$, where for all $i \in [k]$, we have that α_i is the objective she aims to satisfy, and $R_i \geq 0$ is the reward for the satisfaction.

[1] All our results can be extended to weight functions over real numbers. Indeed, we only need to consider the relations among the weights of the finitely many subsets of α. Thus, the only challenge with weight functions over real numbers is their representation.

By adding a threshold to the weight function w, we can make the objective Boolean. Formally, a play ρ satisfies a *Boolean MaxWB* (BMaxWB, for short) objective $\langle \alpha, w, t \rangle$, for $t \in \mathbb{Z}$, if $w(\mathsf{sat}(\rho, \alpha)) \geq t$. By [28], two-player zero-sum BMaxWB games are determined, thus in every game, Player 1 or Player 2 wins.

We define the *size* of a game graph G as the size $|E|$ of its edge relation, and define the size of an objective $\psi = \langle \alpha, w \rangle$ as the size of w, defined as $\sum_{X \subseteq \alpha : w(X) \neq 0} w(X)$. Note that w can be encoded in $|w|$ bits. In fact, for our upper bounds, the encoding of $w(X)$ can be in either unary or binary. That is, the bounds stay valid even when the encoding of $w(X)$ adds to the length of the input only $\log w(X)$ bits. When w is additive, our bounds hold also when we define its length by $\sum_{\alpha_l \in \alpha} w(\alpha_l)$.

2.3 Partially-Fixed Nash-Equilibria with Desired Utilities

We consider a setting in which the players model components of a system and its environment. Technically, we assume that $[k]$ is partitioned to a set $S \subseteq [k]$ of system players, whose behavior is controllable, and the set $[k] \setminus S$ of environment players, which are rational and aim for maximizing their utility. The basic problem we consider is the existence and finding of stable profiles that satisfy some desired properties.

We refine the notion of NEs to take into account our ability to control the system players. For a set $S \subseteq [k]$ of system players and a profile π, we say that π is an *S-fixed NE* if no player in $[k] \setminus S$ can benefit from unilaterally changing her strategy. Thus, for every $i \in [k] \setminus S$ and strategy f'_i, we have that $\mathsf{util}_i(\pi[i \leftarrow f'_i]) \leq \mathsf{util}_i(\pi)$. *Desired utilities* of a k-player game are specified by a predicate $P \subseteq \mathbb{Z}^{[k]}$. We describe such predicates by Boolean assertions with atoms of the form $t_1 \leq t_2$, for arithmetic terms t_1 and t_2 defined over $\{u_1, \ldots, u_k\} \cup \mathbb{Z}$, where for all $i \in [k]$, the variable u_i stands for the utility of Player i.

Formally, for a set X of variables, the set of *terms over* X, denoted $\mathcal{T}_X$, is defined inductively as follows.

- x and n, for $x \in X$ and $n \in \mathbb{Z}$.
- $t_1 + t_2$ and $t_1 - t_2$, for $t_1, t_2 \in \mathcal{T}_X$.

The set of *Boolean assertions over* X, denoted $\mathcal{B}_X$, is defined inductively as follows.

- $t_1 \leq t_2$ for $t_1, t_2 \in \mathcal{T}_X$.
- $\neg b_1$ and $b_1 \wedge b_2$ for $b_1, b_2 \in \mathcal{B}_X$.

Consider an assignment $\xi : X \to \mathbb{Z}$ to the variables in X. We extend ξ to terms in the expected way, thus $\xi : \mathcal{T}_X \to \mathbb{Z}$ is such that $\xi(t_1 + t_2) = \xi(t_1) + \xi(t_2)$, and $\xi(t_1 - t_2) = \xi(t_1) - \xi(t_2)$, for all $t_1, t_2 \in \mathcal{T}_X$.

We also extend ξ to Boolean assertions over X, thus $\xi : \mathcal{B}_X \to \{\mathbf{true}, \mathbf{false}\}$ is defined inductively as follows.

– For $t_1, t_2 \in \mathcal{T}_X$, we have that $\xi(t_1 \leq t_2) = \textbf{true}$ iff $\xi(t_1) \leq \xi(t_2)$.
– $\xi(\neg b) = \neg\xi(b)$, for $b \in \mathcal{B}_X$.
– $\xi(b_1 \wedge b_2) = \xi(b_1) \wedge \xi(b_2)$, for $b_1, b_2 \in \mathcal{B}_X$.

Each Boolean assertion $b \in \mathcal{B}_X$ is a predicate on $\mathbb{Z}^X$, thus an assignment $\xi \in \mathbb{Z}^X$ is in b iff ξ satisfies b. A profile π then satisfies a predicate P if the assignment $f : \{u_1, \ldots, u_k\} \to \mathbb{Z}$ with $f(u_i) = \mathsf{util}_i(\pi)$ satisfies P. For example, the predicate $(u_1 \geq 8) \wedge (u_2 \geq u_3) \wedge (u_3 + u_4 \leq 20)$ requires the utility of Player 1 to be at least 8, the utility of Player 2 not to be smaller than that of Player 3, and the combined utilities of Players 3 and 4 to be at most 20.

The problem of *partially-fixed NE with desired utilities* (DNE, for short) gets as input a k-player game $\mathcal{G}$, a set $S \subseteq [k]$ of system players, and a predicate $P \subseteq \mathbb{Z}^{[k]}$ describing desired utilities. Given $\langle \mathcal{G}, S, P \rangle$, the goal is to return an S-fixed NE π in $\mathcal{G}$ such that the utilities of the players in π satisfy P. Below, we describe useful instances of the DNE problem.

– In *cooperative rational synthesis* [20], the predicate P sets lower bounds on the utilities of the system players. For example, solutions to DNE with $\langle \mathcal{G}, \{1\}, u_1 \geq t \rangle$ are 1-fixed NEs in which the utility of the single system player is at least t.
– Different wellness goals like *total* or *fair wellness* [34] can be specified by bounding the differences among the different utilities. For example, the predicate $\bigwedge_{i \in [k]} (2k \cdot avg \leq k \cdot u_i \leq 4k \cdot avg)$, with $avg = \frac{\sum_{i \in [k]} u_i}{k}$, restricts the distance of each player's utility from the average utility, and the predicate $\bigwedge_{i,j \in [k]} (u_i \leq 2u_j)$ restricts the distance between each two players' utilities.
– *Priorities* among players can be specified by predicates like $\bigwedge_{i \in [k-1]} (u_i \geq u_{i+1})$, which order the utilities, or $u_1 \geq (u_2 + \cdots + u_k)$, which relates the utilities of sets of players.

We conclude this section with two useful lemmas. In the first, we consider zero-sum games that are used when a player may deviate from her current strategy and all the other players cooperate in order to make such a deviation non-beneficial. Formally, consider a MaxWB game $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i \in [k]} \rangle$, with $G = \langle \{V_i\}_{i \in [k]}, v_0, E \rangle$. For a player $i \in [k]$, a vertex $v \in V_i$, and a threshold t, we define the *game from v against Player i with objective (α, w_i, t)* as the two-player zero-sum game $\mathcal{G}^v_{i,t}$ defined as follows. The game is played on G with initial vertex v, between Player i (who is Player 1 in $\mathcal{G}^v_{i,t}$) and the players in $[k[\setminus\{i\}$ (who compose Player 2 in $\mathcal{G}^v_{i,t}$). The objective of Player i is the set of plays in which her utility is at least t. Thus, $\mathcal{G}^v_{i,t} = \langle \langle V_i, V \setminus V_i, v \rangle, \psi \rangle$, with $\psi = \{\rho : w_1(\mathsf{sat}(\rho, \alpha)) \geq t\}$. The following lemma offers a useful characterization of NEs [38, 17].

Lemma 1. *Consider a k-player game $\mathcal{G}$, a set $S \subseteq [k]$, and a predicate $P \subseteq \mathbb{Z}^{[k]}$. For every path ρ of G, there is a solution to the DNE problem for $\langle \mathcal{G}, S, P \rangle$ that has outcome ρ iff the ρ satisfies P and for every player $i \in [k] \setminus S$ and vertex $v \in V_i \cap reach(\rho)$, Player i loses in the game $\mathcal{G}^v_{i,t}$ with $t = \mathsf{util}_i(\rho) + 1$.*

Proof. Consider a path ρ of G. Assume first that there exists a solution $\pi = \langle f_1, \ldots, f_k \rangle$ to the DNE problem for $\langle \mathcal{G}, S, P \rangle$ such that $\rho = \mathsf{Outcome}(\pi)$. First, as π is a solution for $\langle \mathcal{G}, S, P \rangle$, then ρ satisfies P. Next, assume by way of contradiction that there exists $i \in [k] \setminus S$ and $v \in V_i \cap \mathit{reach}(\rho)$ such that Player i wins in the game $\mathcal{G}^v_{i,t}$ with $t = \mathsf{util}_i(\pi) + 1$. Let g^v_i be the winning strategy for Player i in $\mathcal{G}^v_{i,t}$. Consider the strategy f'_i that follows f_i until v is visited and then switches to g^v_i. Consider the profile $\pi[f_i \leftarrow f'_i]$. Since the other players follow their strategies in π, the outcome of $\pi[f_i \leftarrow f'_i]$ has a prefix that reaches v, from where Player i switches to g^v_i and wins in $\mathcal{G}^v_{i,t}$. Thus, the utility of Player i is t, which is bigger than $\mathsf{util}_i(\pi)$, making f'_i a beneficial deviation, contradicting the fact π is an S-fixed NE.

For the other direction, let ρ be a path that satisfies P and assume that for every player $i \in [k] \setminus S$ and vertex $v \in V_i \cap \mathit{reach}(\rho)$, Player i loses in the game $\mathcal{G}^v_{i,t}$ with $t = \mathsf{util}_i(\pi)+1$. Consider the profile π in which all the players move the token in a way that generates ρ and for every $i \in [k] \setminus S$, if Player i deviates and moves the token from a vertex $v \in \mathit{reach}(\rho)$ to a successor that does not extend ρ, then all the players in $[k] \setminus i$ play according to their winning strategy in $\mathcal{G}^v_{i,t}$, for $t = \mathsf{util}_i(\rho)+1$. We claim that π is an S-fixed NE with $\mathsf{Outcome}(\pi) = \rho$, and so it is a solution to the DNE problem for $\langle \mathcal{G}, S, P \rangle$. In particular, for every $i \in [k] \setminus S$, Player i does not benefit from deviating from π. Indeed, only deviations that cause the outcome to depart from ρ may influence the utilities of the players, and deviations that depart from ρ are not beneficial.

The second lemma states that when the weight functions are general, the non-zero-sum setting is at least as hard to reason about as the zero-sum setting.

Lemma 2. *Given a zero-sum BMaxWB game $\mathcal{G} = \langle G, (\alpha, w, t) \rangle$, we can construct weight functions w_1 and w_2 such that Player 1 wins in $\mathcal{G}$ iff there is a DNE solution for $\langle \langle G, \alpha, \{w_1, w_2\} \rangle, \{1\}, u_1 \geq t \rangle$.*

Proof. For all $X \subseteq \alpha$, we define $w_1(X) = w(X)$ and $w_2(X) = -w(X)$. Let $\mathcal{G}' = \langle G, \alpha, \{w_1, w_2\} \rangle$. It is not hard to see that f_1 is a winning strategy for Player 1 in $\mathcal{G}$ iff for all strategies f_2 for Player 2, the profile $\langle f_1, f_2 \rangle$ is a 1-fixed NE in which the utility of Player 1 is at least t. Thus, Player 1 wins in $\mathcal{G}$ iff there is a DNE solution for $\langle \mathcal{G}', \{1\}, u_1 \geq t \rangle$.

3 MaxWB Games

In this section, we study the DNE problem in MaxWB games with general weight functions. Thus, $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i \in [k]} \rangle$ is such that $w_i : 2^\alpha \to \mathbb{Z}$, for every $i \in [k]$. The techniques are similar to these used for reasoning about non-zero-sum games with Streett objectives or weighted reachability objectives [12, 38], and we give them for completeness. Essentially, the proof is based on Lemma 1: the DNE solution for $(\mathcal{G}, S, P)$ has an outcome ρ such that for every vertex v in ρ that is owned by a player i not in S, the strategies of the players in $[k] \setminus \{i\}$ make sure that there is no beneficial deviation for Player i from v. We note that the existence of an NE in MaxWB games (that is, Theorem 1) follows also from the study of NEs in the generalized Muller games in [35]. Our proof, however, sets the stage to optimal algorithms for finding an NE.

Theorem 1. *For every k-player MaxWB game $\mathcal{G}$, and set $S \subseteq [k]$ of system players, there exists an DNE solution for $\langle \mathcal{G}, S, \mathbf{true} \rangle$.*

Proof. Consider a k-player MaxWB game $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i \in [k]} \rangle$. For every player $i \in [k]$ and a vertex $v \in V_i$, let t_v be the maximal threshold such that Player i can force her utility from v to be at least t_v. That is, t_v is the maximal t for which Player i wins the MaxWB game $\mathcal{G}^v_{i,t}$ against her.

For every $i \in [k]$ and a vertex $v \in V$, let g^v_i be a winning strategy for Player i in $\mathcal{G}^v_{i,t_v}$. Note that if u and v are vertices with $t_u = t_v$, then we can assume that $g^v_i = g^u_i$. We define a strategy f_i for Player i as follows. The strategy starts by following the strategy $g^{v_0}_i$. Note that by doing so, the generated play is guaranteed to reach only vertices v with $t_v \geq t_{v_0}$. As long as the play visits vertices v with $t_v = t_{v_0}$, the strategy f_i continues to follow $g^{v_0}_i$. When the play reaches a vertex v with $t_v > t_{v_0}$ the strategy f_i switches to follow the strategy g^v_i, and so on. Note that again, the thresholds that induce the strategies can only increase, and eventually they stabilize.

Consider the profile $\pi = \langle f_1, \ldots, f_k \rangle$. Let $\rho = \mathsf{Outcome}(\pi)$. We construct a profile π' such that $\mathsf{Outcome}(\pi') = \rho$, and π' is an S-fixed NE, for all $S \subseteq [k]$. First, note that for every $i \in [k]$ and a vertex $v \in V_i$ that ρ visits, we have that $t_v \leq \mathsf{util}_i(\pi)$. Indeed, f_i is a winning strategy in the game against Player i from v with the MaxWB objective $\langle \alpha, w_i, t_v \rangle$, thus it forces her utility to be at least t_v. Also, the way we have defined t_v implies that for every vertex $v \in V_i$ that ρ visits, the players in $[k] \setminus \{i\}$ have a winning strategy in the game against Player i with the objective $\langle \alpha, w_i, \mathsf{util}_i(\pi) + 1 \rangle$. Consider the profile π' in which the players jointly generate the play ρ, and if a player $i \in [k]$ deviates, the other players proceed to use the winning strategy in the game against Player i with the MaxWB objective $\langle \alpha, w_i, \mathsf{util}_i(\pi)+1 \rangle$. Using the same arguments as in Lemma 1, the profile π' has outcome ρ' and is an NE, and hence also an S-fixed NE, for all $S \subseteq [k]$.

Clearly, once a predicate P is added, a desired S-fixed NE need not exist. Once, however, there is a DNE solution, there is also one with an outcome of polynomial length:

Theorem 2. *Consider a k-player MaxWB game $\mathcal{G}$, a set $S \subseteq [k]$ of system players, and a utility predicate P. If there exists a DNE solution π for $\langle \mathcal{G}, S, P \rangle$, then there also exists a DNE solution π' for $\langle \mathcal{G}, S, P \rangle$ such that all the following hold.*

1. $\mathsf{Outcome}(\pi') = \rho_1 \cdot \rho_2^\omega$, *where ρ_1 and ρ_2 are of polynomial size.*
2. *reach*$(\mathsf{Outcome}(\pi')) \subseteq$ *reach*$(\mathsf{Outcome}(\pi))$.
3. *inf*$(\mathsf{Outcome}(\pi')) =$ *inf*$(\mathsf{Outcome}(\pi))$.

Proof. Let $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i\in[k]}\rangle$, and assume that π is a DNE solution for $\langle \mathcal{G}, S, P\rangle$. Let $\rho = \mathsf{Outcome}(\pi)$.

Let $U = \mathit{inf}(\rho)$ be the set of vertices that ρ visits infinitely often, and consider a vertex $u \in U$. Note that ρ eventually gets stuck in U, and thus there is a cycle ρ_2 through u, of length at most $|U|^2$, that visits exactly all the vertices in U. Also, since $u \in \mathit{reach}(\rho)$, there is a simple path $\rho_1 \subseteq \mathit{reach}(\rho)$ from v_0 to u. Let $\rho' = \rho_1 \cdot \rho_2^\omega$. Note that ρ_1 and ρ_2 are of polynomial size, $\mathit{reach}(\rho') \subseteq \mathit{reach}(\rho)$, and $\mathit{inf}(\rho') = \mathit{inf}(\rho)$. Since the utilities for the players in ρ satisfy P, the latter implies that so do the utilities for the players in ρ'.

By Lemma 1, for every player $i \in [k]\setminus S$ and vertex $v \in V_i \cap \mathit{reach}(\rho)$, Player i loses in the game $\mathcal{G}^v_{i,t}$ with $t = \mathsf{util}_i(\pi)+1$. Since $\mathit{reach}(\rho') \subseteq \mathit{reach}(\rho)$, the above holds also for every player $i \in [k] \setminus S$ and vertex $v \in V_i \cap \mathit{reach}(\rho')$.

Consider a profile π' in which all the players move the token in a way that generates ρ' and for every $i \in [k]\setminus S$, if Player i deviates and moves the token from a vertex $v \in \mathit{reach}(\rho')$ to a successor that does not extend ρ', then all the players in $[k] \setminus \{i\}$ play according to their winning strategy in $\mathcal{G}^v_{i,t}$, for $t = \mathsf{util}(\rho) + 1$. By Lemma 1, the profile π' is a DNE solution for $\langle \mathcal{G}, S, P\rangle$.

We continue to the complexity of the DNE problem. BMaxWB objectives are strongly related to *Muller* objectives. A Muller objective is defined with respect to a finite set $\mathcal{C}$ of colors and is a pair $\psi = \langle \mathcal{F}, \chi\rangle$, where $\mathcal{F} \subseteq 2^{\mathcal{C}}$ specifies desired subsets of colors and $\chi : V \to \mathcal{C}$ colors the vertices in V. A play ρ satisfies ψ iff the set of colors visited infinitely often along ρ is in $\mathcal{F}$. That is, $\{i \in \mathcal{C} : \mathit{inf}(\rho) \cap \chi^{-1}(i) \neq \emptyset\} \in \mathcal{F}$. It is shown in [28] that every BMaxWB objective $\psi = \langle \alpha, w, t\rangle$ has an equivalent Muller objective of size $|\{X \subseteq \alpha : w(X) \geq t\}|$, and every Muller objective $\psi = \langle \mathcal{F}, \chi\rangle$ has an equivalent BMaxWB objective of size $|\psi|$. It follows that Muller games are polynomially reducible to BMaxWB games, and vice versa, and so zero-sum two-player BMaxWB games are PSPACE-complete. As detailed below, this implies a PSPACE completeness also for the DNE problem.

Theorem 3. *Given a k-player MaxWB game $\mathcal{G}$, a set $S \subseteq [k]$ of system players, and a utility predicate P, deciding whether there exists a DNE solution for $\langle \mathcal{G}, S, P\rangle$ is PSPACE-complete. Hardness in PSPACE already applies for $k = 2$.*

Proof. Consider a k-player game $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i\in[k]}\rangle$, a set $S \subseteq [k]$, and a predicate $P \subseteq \mathbb{Z}^{[k]}$. We describe a NPSPACE algorithm that decide whether there exists a DNE solution for $\langle \mathcal{G}, S, P\rangle$. Since NPSPACE = PSPACE, a PSPACE upper bound follows.

The algorithm guesses a polynomial-sized lasso-shaped play ρ in G, and checks that the utilities for the players satisfy P. Then, for every $i \in [k] \setminus S$ and a vertex $v \in V_i$ that ρ visits, the algorithm checks that Player i does not win the game against her with the objective $\langle \alpha, w_i, w_i(\mathsf{sat}(\rho, \alpha)) + 1\rangle$ from v, which can be done in PSPACE [28]. By Lemma 1 and Theorem 2, the algorithm finds a DNE solution iff one exists.

Finally, a PSPACE lower bound follows from Lemma 2 and the fact the problem of deciding zero-sum two-players BMaxWB games is PSPACE-hard [28].

4 MaxWB Games with Non-Decreasing Weight Functions

In this section, we study MaxWB games with non-decreasing weight functions. Thus, $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i\in[k]}\rangle$ is such that for all $i \in [k]$ and $X, X' \subseteq \alpha$ with $X \subseteq X'$, we have that $w_i(X) \leq w_i(X')$. We show that every DNE solution for $\mathcal{G}$ has an equivalent DNE solution of polynomial size. Thus, while Theorem 2 only bounds the length of the outcome of an equivalent solution, the restriction to non-decreasing function also bounds the memory requirements for the players in the solution, making the problem easier.

Theorem 4. *Consider a k-player MaxWB game $\mathcal{G}$ with non-decreasing weight functions, a set $S \subseteq [k]$ of system players, and a utility predicate P. If there exists a DNE solution for $\langle \mathcal{G}, S, P\rangle$, then there also exists a DNE solution for $\langle \mathcal{G}, S, P\rangle$ of size polynomial in $\mathcal{G}$.*

Proof. Let $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i\in[k]}\rangle$, and assume that π is a DNE solution for $\langle \mathcal{G}, S, P\rangle$. By Theorem 2, we can assume that $\rho = \mathsf{Outcome}(\pi)$ is of polynomial size.

Consider a profile π' in which all the players move the token in a way that generates ρ and for every $i \in [k] \setminus S$, if Player i deviates and moves the token from a vertex v to a successor that does not extend ρ, then all the players in $[k] \setminus \{i\}$ play according to their winning strategy in $\mathcal{G}^v_{i,t}$, for $t = \mathsf{util}(\rho) + 1$. By Lemma 1, the profile π' is a DNE solution for $\langle \mathcal{G}, S, P\rangle$. By [28], if Player 2 has a winning strategy in a zero-sum MaxWB game with non-decreasing weight functions, then she also has a memoryless winning strategy. Therefore, the size of winning strategies for Player 2 in games of the form $\mathcal{G}^v_{i,t}$ is polynomial in the size of $\mathcal{G}$. Hence, the profile π' is of polynomial size, and we are done. □

We continue to the complexity of the DNE problem. Note that the reduction in Lemma 2 involves negative weight functions, so we cannot apply it. We can still show a lower bound for $k = 2$, but the proof is more complicated and involves a composition and dualization of BMaxWB games.

Theorem 5. *Given a k-player MaxWB game $\mathcal{G}$ with non-decreasing weight functions, a set $S \subseteq [k]$ of system players, and a utility predicate P, deciding whether there exists a DNE solution for $\langle \mathcal{G}, S, P\rangle$ is NP-complete. Hardness in NP already applies for $k = 2$, positive and additive weight functions, and P that only refers to the utility of Player 1.*

Proof. For the NP upper bound, consider a k-player game $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i\in[k]}\rangle$, a set of players $S \subseteq [k]$ and a predicate P. An NP algorithm guesses a path $\rho =$

$\rho_1 \cdot (\rho_2)^\omega$ such that the length of ρ_1 and ρ_2 are polynomial in $|G|$. Additionally, for every $i \in [k] \setminus S$, the algorithm guesses a memoryless strategy g_i for Player 2 in the game against Player i with the BMaxWB objective $\psi_i = \langle \alpha, w_i, \mathsf{util}_i(\rho)+1 \rangle$. The algorithm checks that $\mathsf{util}_1(\rho), \ldots, \mathsf{util}_k(\rho)$ satisfy P, and checks that for every $i \in [k] \setminus S$, the strategy g_i is indeed winning for Player 2 in the game against Player i with ψ_i, from every vertex v that ρ reaches. Note that indeed g_i can be verified in polynomial time [28]. By Lemma 1 and Theorem 2, there exists such a path and strategies iff there exists a DNE solution for $\langle \mathcal{G}, S, P \rangle$.

For the lower bound, we describe a reduction from the problem of deciding whether Player 2 wins in a zero-sum BMaxWB game, known to be NP-hard already for positive and additive weight functions and threshold $t \geq 1$ [28]. Consider a zero-sum BMaxWB game $\mathcal{G} = \langle G, \langle w, \alpha, t \rangle \rangle$, with $G = \langle V_1, V_2, v_0, E \rangle$. We construct a two-player non-zero-sum MaxWB game $\mathcal{G}' = \langle G', \alpha', \{w_1, w_2\} \rangle$ with positive and additive weight functions such that there exists a DNE solution for $(\mathcal{G}', \{1\}, \{u_1 \geq 1\})$ iff Player 2 wins in $\mathcal{G}$,

Intuitively, we define $\mathcal{G}'$ so that Player 2 chooses to help Player 1 to have utility 1 iff there is no winning strategy for Player 1 in G. Specifically, the game graph G' is as follows. From the initial vertex, Player 2 chooses between moving to a self-looped sink $\perp$, and moving to a copy of G. In the copy of G, Player 2 aims to satisfy the BMaxWB objective of $\mathcal{G}$. Note that the original roles of Player 1 and Player 2 in G are switched in its copy in G'. We define the weight functions so that Player 1 has utility 1 if the play reaches $\perp$ and utility 0 otherwise, and Player 2 has utility $t-1$ if the play reaches $\perp$ and utility according to w and the outcome in G otherwise. Since $t \geq 1$, the weight functions in $\mathcal{G}'$ are indeed positive and additive. Let $S = \{1\}$ and $P = \{u_1 \geq 1\}$. Note that P is satisfied iff the play reaches the sink $\perp$, thus there exists a DNE solution iff Player 1 can make sure Player 2 cannot benefit from moving to the copy of G, which holds iff Player 1 has a strategy in G that forces the satisfaction value of $\langle \alpha, w \rangle$ to be at most $t-1$. Accordingly, there exists a DNE solution in $\mathcal{G}'$ iff Player 2 wins $\mathcal{G}$.

Formally, the game $\mathcal{G}' = \langle G', \alpha', \{w_1, w_2\} \rangle$ is defined as follows.

1. $G' = \langle V_1', V_2', v_0', E' \rangle$ where:
   (a) The vertices are $V_1' = V_2$, and $V_2' = V_1 \cup \{v_0', \perp\}$.
   (b) The edge set is $E' = E \cup \{\langle v_0', \perp \rangle, \langle v_0', v_0 \rangle\}$.
2. The objective set is $\alpha' = \alpha \cup \{\alpha_\perp\}$ where $\alpha_\perp = \{\perp\}$.
3. the weight functions are defined as follows.
   (a) For every $\alpha_l \in \alpha'$, if $\alpha_l = \alpha_\perp$ then $w_1(\alpha_l) = 1$, otherwise $w_1(\alpha_l) = 0$.
   (b) For every $\alpha_l \in \alpha'$, if $\alpha_l = \alpha_\perp$ then $w_2(\alpha_l) = t-1$, otherwise $w_2(\alpha_l) = w(\alpha_l)$.

We prove the correctness of the construction. For the first direction, assume Player 2 wins $\mathcal{G}$. Let π be a profile in $\mathcal{G}'$ in which Player 2 proceeds from the initial vertex to the sink $\perp$, and Player 1 uses the winning strategy of Player 2 from $\mathcal{G}$ in the copy of G. We show that π is a 1-fixed NE with $\mathsf{util}_1(\pi) \geq 1$. First, note that since $\mathsf{Outcome}(\pi)$ reaches $\perp$, the utility for Player 1 is 1. Next, note that π is a 1-fixed NE. Indeed, $\mathsf{util}_2(\pi) = w_2(\alpha_\perp) = t-1$, and for every strategy f_2 for Player 2 that proceeds to the copy of G, we have that $\mathsf{util}_2(\pi[2 \leftarrow f_2]) \leq$

$t-1$, since Player 1 uses a winning strategy of Player 2 from $\mathcal{G}$ in the copy of G, that ensures the BMaxWB objective $\langle \alpha, w, t \rangle$ is not satisfied.

For the second direction, assume that Player 1 wins $\mathcal{G}$. Then, Player 2 can guarantee a utility of at least t by proceeding to the copy of G from the initial vertex and then using a winning strategy for Player 1 from $\mathcal{G}$. Thus, there is no 1-fixed NE in which Player 2 proceeds from the initial vertex to $\bot$ since in such a profile Player 2 only receive a utility of $t-1$ and benefits from deviating. Accordingly, there is no 1-fixed NE in which the utility for Player 1 is at least 1, and so there is no DNE solution for S and P. $\square$

5 MaxWB Games with Additive Weight Functions

In this section, we study MaxWB games with additive weight functions. Thus, $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i\in[k]} \rangle$ is such that for all $i \in [k]$ and $\alpha_j \in \alpha$, we have that $w_i(\alpha_j) \in \mathbb{Z}$, and for $X \subseteq \alpha$, we have that $w_i(X) = \sum_{\alpha_l \in X} w_i(\alpha_l)$. Note that additive weight functions correspond to cases the objectives are independent of each other. In particular, ordered objectives, as in [10], can be specified using additive weight functions.

In [28], the authors study zero-sum BMaxWB games with positive and additive weight functions. In particular, they show that the problem of deciding the winner is co-NP-complete, thus the games are easier than these with general weight functions. We first complete the picture known for the zero-sum case and study zero-sum BMaxWB games with additive (but not necessarily positive) weight functions. Surprisingly, additive weight functions can express all Muller objectives:

Theorem 6. *Consider a Muller objective $\langle \mathcal{F}, \chi \rangle$ defined over a set of colors $[m]$ and vertices in V. There is a set $\alpha \subseteq 2^V$ of underlying objectives and an additive weight function $w : \alpha \to \mathbb{Z}$ such that the objectives $\langle \mathcal{F}, \chi \rangle$ and $\langle \alpha, w, 0 \rangle$ are equivalent.*

Proof. Consider a Muller objective $\langle \mathcal{F}, \chi \rangle$ defined over a set of colors $[m]$, for $m \in \mathbb{N}$. Let $\alpha = \{\alpha_C : C \subseteq [m], C \neq \emptyset\}$, where for every set of colors $C \subseteq [m]$, we have that $\alpha_C = \bigcup_{i \in C} \chi^{-1}(i)$. We show that there exists an additive weight function $w : \alpha \to \mathbb{Z}$ such that $\langle \mathcal{F}, \chi \rangle$ and $\langle \alpha, w, 0 \rangle$ are equivalent. Note that the sets α_C need not be singletons. The important thing is that that weight function w is defined for single objectives in α and induces an additive function.

Note also that a play ρ with $\chi(\mathit{inf}(\rho)) = C$ satisfies exactly all the Büchi objectives $\alpha_{C'} \in \alpha$ such that $C \cap C' \neq \emptyset$. Accordingly, in order for $\langle \alpha, w, 0 \rangle$ to be equivalent to $\langle \mathcal{F}, \chi \rangle$, the sum of weights of objectives $\alpha_{C'}$ such that $C \cap C' \neq \emptyset$ should be at least 0 when $C \in \mathcal{F}$, and smaller than 0 otherwise. Thus, the weight function w must satisfy the following.

1. $\sum_{C' \subseteq [m] : C \cap C' \neq \emptyset} w(\alpha_{C'}) \geq 0$, for every $C \in \mathcal{F}$.
2. $\sum_{C' \subseteq [m] : C \cap C' \neq \emptyset} w(\alpha_{C'}) < 0$, for every $C \in 2^{[m]} \setminus \mathcal{F}$.

In order to prove that such a weight function exists, we prove below that there exists an integral solution for the following system of linear equations, defined over the set of variables $\{x_C : C \subseteq [m], C \neq \emptyset\}$. The solution then induces the weight function w, with $w(\alpha_C) = x_C$:

1. The equation e_C for $C \in \mathcal{F}$ is $\sum_{C' \subseteq [m]: C \cap C' \neq \emptyset} x_{C'} = 0$.
2. The equation e_C for $C \in 2^{[m]} \setminus (\mathcal{F} \cup \{\emptyset\})$ is $\sum_{C' \subseteq [m]: C \cap C' \neq \emptyset} x_{C'} = -1$.

Let $n = 2^m - 1$, and let $C_1, \ldots, C_n$ be the sets $C \in 2^{[m]} \setminus \{\emptyset\}$, ordered according to $|C|$, where sets with the same size appear in a lexicographical order. Let A be the matrix that correspond to the linear system of equations, where the i-th column corresponds to the variable x_{C_i}, and the i-th row corresponds to the equation e_{C_i}. Note that for every pair $i, j \in [n]$ we have that $A[i,j] = 1$ iff $C_i \cap C_j \neq \emptyset$, and $A[i,j] = 0$ otherwise. We show that A satisfies the following properties.

1. $A[i, n-i] = 0$, for every $1 \leq i < n$.
2. $A[i,j] = 1$, for every $1 \leq i \leq n$ and $n - i + 1 \leq j \leq n$.

Intuitively, $C_{n-i} = [m] \setminus C_i$, for every $1 \leq i < n$. Then, $A[i, n-i] = 0$, since $C_i \cap ([m] \setminus C_i) = \emptyset$. Also, since $|[m] \setminus C_i| \leq |C_j|$ for every $j > n - i$, we have that C_j contains elements from C_i. Thus, $C_i \cap C_j \neq \emptyset$ and $A[i,j] = 1$, for every $j > n - i$. So, the n-th row is all 1s, the $(n-1)$-th row has 0 in the first column, and then it is all 1s, the $(n-2)$-th row has 0 in the second column and then all 1s, the $(n-3)$-th row has 0 in the third column and then all 1s, and so on (see Example 1).

Thus, to prove A has the above properties, it is sufficient to show that $C_{n-i} = [m] \setminus C_i$. Consider a set C_i of size k, and assume C_i is the j-th smallest subset of $[m]$ of size k. Thus, $i = \sum_{l=1}^{k-1} \binom{m}{l} + j$. Since C_i is the j-th smallest subset of $[m]$ of size k, the set $[m] \setminus C_i$ is the j-th biggest subset of $[m]$ of size $m - k$. Indeed, when the lexicographic index of C_i among sets of size k gets smaller, the lexicographic index of $[m] \setminus C_i$ gets bigger. Hence, $[m] \setminus C_i = C_{i'}$ such that $i' = n - (\sum_{l=m-k+1}^{m} \binom{m}{l} + j - 1)$. Since $\binom{m}{l} = \binom{m}{m-l}$, we have that $\sum_{l=m-k+1}^{m} \binom{m}{l} = \sum_{l=m-k+1}^{m} \binom{m}{m-l} = \sum_{l=k-1}^{0} \binom{m}{l} = \sum_{l=1}^{k-1} \binom{m}{l} + 1 = i + 1 - j$. Thus, $i' = n - (\sum_{l=m-k+1}^{m} \binom{m}{l} + j - 1) = n - (i + 1 - j + j - 1) = n - i$.

We now show that, due to the above properties, there exists a sequence of row subtractions in the matrix that reaches a matrix with a diagonal of 1s, and 0s in all the other entries. That is, we describe an algorithm to reduce A to the identity matrix. Note that it implies that the system of equations has a solution. Also note that the only row operations we use to reduce A to the identity matrix (up to changing the order of the rows) is subtraction, implying the solution is integral.

The sequence of row subtractions is defined as follows, when R_i is used to denote the i-th row.

1. For every i from n to 2:
   (a) Subtract $R_i \leftarrow R_i - R_{i-1}$.

(b) Subtract $R_j \leftarrow R_j - R_i$, for every $j < i$ such that $A[j, n-i+1] = 1$.

Note that the first subtraction leaves $A[i, n-i+1] = 1$, since $A[i-1, n-i+1] = 0$, and $A[i,j] = 0$ for every $j > n-i+1$. Then, since R_i has a single 1 entry $A[i, n-i+1]$, the following subtractions make the rest of the entries in the $(n-i+1)$-th column to be 0.

Example 1. Let $m = 3$, and consider the Muller objective $\langle \mathcal{F}, \chi \rangle$ with $\mathcal{F} = \{\{1\}, \{2\}, \{3\}, \{1,2,3\}\}$. The corresponding system of equations $A \cdot \overline{x} = \overline{b}$ appears in Table 1. Table 2 shows the system of equations after subtracting $R_7 \leftarrow R_7 - R_6$, and Table 3 shows the system of equations after subtracting $R_i \leftarrow R_i - R_7$, for every $i < 7$ such that $A[i,1] = 1$. The solution is then $x_{\{1\}} = x_{\{2\}} = x_{\{3\}} = 1$, $x_{\{1,2\}} = x_{\{1,3\}} = x_{\{2,3\}} = -2$, and $x_{\{1,2,3\}} = 3$. □

	$x_{\{1\}}$	$x_{\{2\}}$	$x_{\{3\}}$	$x_{\{1,2\}}$	$x_{\{1,3\}}$	$x_{\{2,3\}}$	$x_{\{1,2,3\}}$	
$e_{\{1\}}$	1	0	0	1	1	0	1	0
$e_{\{2\}}$	0	1	0	1	0	1	1	0
$e_{\{3\}}$	0	0	1	0	1	1	1	0
$e_{\{1,2\}}$	1	1	0	1	1	1	1	-1
$e_{\{1,3\}}$	1	0	1	1	1	1	1	-1
$e_{\{2,3\}}$	0	1	1	1	1	1	1	-1
$e_{\{1,2,3\}}$	1	1	1	1	1	1	1	0

Table 1. The system of equations $A \cdot \overline{x} = \overline{b}$, for $m = 3$ and $\mathcal{F} = \{\{1\}, \{2\}, \{3\}, \{1,2,3\}\}$.

	$x_{\{1\}}$	$x_{\{2\}}$	$x_{\{3\}}$	$x_{\{1,2\}}$	$x_{\{1,3\}}$	$x_{\{2,3\}}$	$x_{\{1,2,3\}}$	
$e_{\{1\}}$	1	0	0	1	1	0	1	0
$e_{\{2\}}$	0	1	0	1	0	1	1	0
$e_{\{3\}}$	0	0	1	0	1	1	1	0
$e_{\{1,2\}}$	1	1	0	1	1	1	1	-1
$e_{\{1,3\}}$	1	0	1	1	1	1	1	-1
$e_{\{2,3\}}$	0	1	1	1	1	1	1	-1
$e_{\{1,2,3\}}$	1	0	0	0	0	0	0	1

Table 2. The system of equations after subtracting $R_7 \leftarrow R_7 - R_6$.

Thus, the MaxWB objective $\langle \alpha, w, 0 \rangle$, with $w(\alpha_{\{1\}}) = w(\alpha_{\{2\}}) = -1$ and $w(\alpha_{\{1,2\}}) = 1$, is equivalent to $\langle \mathcal{F}, \chi \rangle$. □

The number of underlying Büchi objectives in the equivalent MaxWB objective generated in Theorem 6 is exponential in the number of colors. Also, while Muller objectives can refer to all subsets of objectives that are satisfied, such a reference is succinct in additive weight functions, and so the translation of

	$x_{\{1\}}$	$x_{\{2\}}$	$x_{\{3\}}$	$x_{\{1,2\}}$	$x_{\{1,3\}}$	$x_{\{2,3\}}$	$x_{\{1,2,3\}}$	
$e_{\{1\}}$	0	0	0	1	1	0	1	-1
$e_{\{2\}}$	0	1	0	1	0	1	1	0
$e_{\{3\}}$	0	0	1	0	1	1	1	0
$e_{\{1,2\}}$	0	1	0	1	1	1	1	-2
$e_{\{1,3\}}$	0	0	1	1	1	1	1	-2
$e_{\{2,3\}}$	0	1	1	1	1	1	1	-1
$e_{\{1,2,3\}}$	1	0	0	0	0	0	0	1

Table 3. The system of equations after subtracting $R_i \leftarrow R_i - R_7$, for every $i < 7$ such that $A[i, 1] = 1$.

BMaxWB objectives with additive weight functions to Muller objectives is also exponential. Thus, while the two formalisms are as expressive, the best translation between them is exponential in both directions, and we cannot easily lift known results about Muller games to games with MaxWB objectives with additive weight functions. We show that the complexity of MaxWB games with additive weight functions still coincides with the one of Muller games. Thus, the advantage of positive additive weight functions is lost. We start with zero-sum games. Essentially, the upper bound follows from the fact the PSPACE algorithm in [31] does not need an explicit representation of the Muller objective, and the lower bound follows from a careful examination of the PSPACE-hardness proof for Muller games [23], showing that the Muller objective used there can be specified as a BMaxWB objective with an additive weight function of polynomial size.

Theorem 7. *Deciding whether Player 1 wins a zero-sum two-player BMaxWB game with an additive weight function is PSPACE-complete.*

Proof. We start with the upper bound. As discussed above, the translation from a BMaxWB objective with additive weight function to a Muller objective, may involve an exponential blow up, and thus we cannot simply use the known PSPACE upper bound for zero-sum two-player Muller games. The PSPACE algorithm in [31], however, does not need an explicit representation of the Muller objective. Rather, it examines outcomes of the game and checks whether they satisfy the objective. A similar check can be done in PSPACE also when the objectives are BMaxWB objectives given by additive weight functions.

For the lower bound, we describe a reduction from QBF. That is, given a QBF formula Φ, we construct a zero-sum BMaxWB game $\mathcal{G}_\Phi$ with an additive weight function such that $\Phi = \mathbf{true}$ iff Player 1 wins $\mathcal{G}_\Phi$.

Consider a set $X = \{x_1, \ldots, x_n\}$ of variables, and let $\overline{X} = \{\overline{x_1}, \ldots, \overline{x_n}\}$. We assume that QBF formulas are of the form $\Phi = \exists x_1 \forall x_2 \ldots \exists x_{n-1} \forall x_n \varphi$ and φ is a propositional formula over $X \cup \overline{X}$ given in 3DNF. That is, $\varphi = C_1 \vee \cdots \vee C_k$ with $C_i = (l_i^1 \wedge l_i^2 \wedge l_i^3)$ and $l_i^1, l_i^2, l_i^3 \in X \cup \overline{X}$, for every $i \in [k]$. The QBF problem is to decide whether $\Phi = \mathbf{true}$. For every $i \in [n]$, we say that the index of the literals x_i and $\overline{x_i}$ is i. Note that all the existentially-quantified variables have odd indices, and all universally-quantified literals have even indices.

Our reduction is similar to the reduction from QBF to deciding Muller games [23], showing that the Muller objectives used there can be specified as BMaxWB objectives with additive weight functions of polynomial size. Essentially, the reduction in [23] constructs from Φ a game graph G_Φ in which Player 1 chooses a clause C_i of φ, and then Player 2 chooses a literal l_i^j of C_i. Choosing a literal involves proceeding to a vertex that corresponds to that literal. Then, the play proceeds to traverse a string of vertices that correspond to all the literals with indices bigger than the index of l_i^j, and returns to the initial vertex. The Muller objective $\psi = \langle \mathcal{F}, \chi \rangle$ is defined over the set of colors $\{\bot\} \cup X \cup \overline{X}$, and χ is defined so every vertex that corresponds to a literal l has color l, and the initial and clause vertices have color $\bot$. Then, $\mathcal{F}$ contains the following sets of colors.

1. $F_i = \{\bot\} \cup \{x_j, \overline{x_j} : i \leq j \leq n\}$, for every even $i \in [n]$.
2. $\{x_i\} \cup F_{i+1}$, for every odd $i \in [n]$.
3. $\{\overline{x_i}\} \cup F_{i+1}$, for every odd $i \in [n]$.

Note that the Muller objective ψ is satisfied if for every existentially-quantified variable x_i, the play traverses both vertices that correspond to the literal x_i and the literal $\overline{x_i}$ only if there exists a variable with smaller index j such that the play traverses both vertices that correspond to the literal x_j and the literal $\overline{x_j}$.

As proven in [23], Player 1 has a winning strategy for ψ in G_Φ iff $\Phi = \mathbf{true}$. Thus, it is sufficient to construct a MaxWB objective $\psi' = \langle \alpha, w, t \rangle$ such that for every play ρ in G_Φ we have that ψ is satisfied in ρ iff ψ' is satisfied in ρ.

For every literal $l \in X \cup \overline{X}$, we define the Büchi objective $\alpha_l = \chi^{-1}(l)$. That is, α_l is satisfied iff the play visits vertices that correspond to l infinitely often. Note that by the definition of the game, if an objective α_l is satisfied, then also every objective α_l' is satisfied, where l' is a literal with an index bigger than the index of l. Then, we define the weight function w and the threshold t so objectives that correspond to literals of existentially-quantified variables have negative weights, objectives that correspond to literals of universally-quantified variables have positive weights, and a sum of weights of a set of objectives is above t iff the corresponding set of literals is in $\mathcal{F}$.

Formally, $\psi' = \langle \alpha, w, t \rangle$ is defined as follows.

1. The set of objectives is $\alpha = \{\alpha_l : l \in X \cup \overline{X}\}$, where $\alpha_l = \chi^{-1}(l)$. That is, α_l consists of all the vertices that correspond to the literal l.
2. The weight function $w : \alpha \to \mathbb{Z}$ is defined as follows.
   (a) For every odd $i \in [n]$ and literal $l \in \{x_i, \overline{x_i}\}$, we define $w(\alpha_l) = -i$.
   (b) For every even $i \in [n]$ and literal $l \in \{x_i, \overline{x_i}\}$, we define $w(\alpha_l) = i$.
3. The threshold is $t = n + 1$.

We prove the correctness of the construction. Consider a play ρ in G_Φ, and let i be the minimal index of a literal $l \in X \cup \overline{X}$ such that α_l is satisfied in ρ. That is, the play satisfies α_{x_i}, $\alpha_{\overline{x_i}}$, or both, and does not satisfy objectives that correspond to literals with indices smaller than i. Note that all the objectives in $\{\alpha_{x_j}, \alpha_{\overline{x_j}} : i + 1 \leq j \leq n\}$ are satisfied in ρ, and they contribute the sum of weights $2\sum_{j=i+1}^{n} w(\alpha_{x_j})$. We calculate the sum of weights for the case i is even, and the case i is odd. For $i \in n$, let $up(i) = \{j : j \geq i \text{ and } j \text{ is odd}\}$.

1. If i is even, then $2\sum_{j=i+1}^{n} w(\alpha_{x_j}) = 2\sum_{j=i+1}^{n}(-1)^j \cdot j = 2\sum_{j\in up(i+1)}(-j+j+1) = 2\sum_{j\in up(i+1)} 1 = 2\cdot\dfrac{n-(i+1)+1}{2} = n-i$.
2. If i is odd, then $2\sum_{j=i+1}^{n} w(\alpha_{x_j}) = 2(i+1) + 2\sum_{j\in up(i+2)}(-j+j+1) = 2(i+1)+2\sum_{j\in up(i+2)} 1 = 2(i+1)+2\cdot\dfrac{n-(i+2)+1}{2} = 2(i+1)+n-(i+1) = n+i+1$.

We prove that the Muller objective ψ and the BMaxWB objective ψ' are equivalent. For the first direction, assume that ψ is satisfied in ρ. If i is even, then both α_{x_i} and $\alpha_{\overline{x_i}}$ are satisfied in ρ, and thus $w(\mathsf{sat}(\rho,\alpha)) = 2i + (n-i) = n+i \geq n+2 > t$. If i is odd, then α_{x_i} or $\alpha_{\overline{x_i}}$ are not satisfied in ρ, and thus $w(\mathsf{sat}(\rho,\alpha)) = -i + (n+i+1) = n+1 = t$. Therefore, ψ' is satisfied in ρ.

For the second direction, assume ψ is not satisfied in ρ. If i is even, then α_{x_i} or $\alpha_{\overline{x_i}}$ are not satisfied in ρ, and thus $w(\mathsf{sat}(\rho,\alpha)) = i+(n-i) = n < t$. If i is odd, both α_{x_i} and $\alpha_{\overline{x_i}}$ are satisfied in ρ, and thus $w(\mathsf{sat}(\rho,\alpha)) = -2i + (n+i+1) = n-i+1 \leq n-1+1 = n < t$. Therefore, ψ' is not satisfied in ρ. □

We continue to non-zero-sum games. The upper bound is similar to the one in Theorem 3, using Theorem 7 for the involved zero-sum BMaxWB games. The lower bound follows from Theorem 7 and Lemma 2.

Theorem 8. *Given a k-player MaxWB game $\mathcal{G}$ with additive weight functions, a set $S \subseteq [k]$ of system players, and a utility predicate P, deciding whether there exists a DNE solution for $\langle \mathcal{G}, S, P\rangle$ is PSPACE-complete. Hardness in PSPACE already applies for $k = 2$.*

6 MaxWB Games with Restricted Additive Weight Functions

In Section 5, we saw that allowing negative weights increases the complexity of the DNE problem in MaxWB games with additive weight functions. Specifically, the complexity jumps from NP to PSPACE. In this section, we examine whether we can allow some restricted use of negative weights in additive weight functions in a way that does not increase the complexity. We consider two cases. The first is when at most one objective is allowed to get a negative weight, and the second is when the weights enjoy some polarity, making the setting closer to zero-sum games.

6.1 When at most one objective may be negative

An additive weight function $w : \alpha \to \mathbb{Z}$ is *almost positive* if there exists $l \in [m]$ such that for every $j \in [m] \setminus \{l\}$, we have that $w(\alpha_j) \geq 0$. That is, at most one objective can have a negative weight. Note that different players may have different negative objectives.

In [28], it is shown that a BMaxWB objective with a positive weight function can be translated to an equivalent *generalized Büchi* objective. Here, we show that a BMaxWB objective with an almost positive weight function can be translated to an equivalent *Streett* objective. A Streett objective is a set $\psi = \{\langle L_l, R_l\rangle\}_{l\in[m]} \subseteq 2^V \times 2^V$ of pairs of sets of vertices. A play ρ satisfies ψ iff for every $l \in [m]$, if ρ visits L_l infinitely often, then it also visits R_l infinitely often. That is, $\mathit{inf}(\rho) \cap L_l = \emptyset$ or $\mathit{inf}(\rho) \cap R_l \neq \emptyset$. A generalized Büchi objective is a special case of Streett, with $L_l = V$ for all $l \in [m]$. The important point for us is that, as is the case with generalized Büchi objectives, zero-sum two-player Streett games are such that if Player 2 wins, then she has a memoryless winning strategy. Thus, handling of the single objective with negative weight involves a transition from generalized Büchi to Streett objectives, but still leaves us in the terrain of games in which Player 2 has a memoryless winning strategy.

Theorem 9. *Consider a zero-sum MaxWB game $\mathcal{G}$ with an almost-positive weight function. Player 2 wins $\mathcal{G}$ iff she has a memoryless winning strategy.*

Proof. Consider a zero-sum MaxWB game $\mathcal{G} = \langle G, \psi\rangle$ with $\psi = \langle \alpha, w, t\rangle$ and $\alpha = \{\alpha_1, ..., \alpha_m\}$ such that w is almost positive. WLOG, assume that $w(\alpha_m) < 0$.

Since α_m is the only objective with a negative weight, the objective ψ is satisfied in a play if α_m is not satisfied and the sum of weights of satisfied objectives from $\alpha \setminus \{\alpha_m\}$ is above t; or if α_m is satisfied and the sum of weights of satisfied objectives from $\alpha \setminus \{\alpha_m\}$ is above $t - w(\alpha_m)$. That is, ψ is satisfied iff α_m is not satisfied and the BMaxWB objectives $\psi_1 = \langle \alpha \setminus \{\alpha_m\}, w, t\rangle$ is satisfied, or the BMaxWB objective $\psi_2 = \langle \alpha \setminus \{\alpha_m\}, w, t - w(\alpha_m)\rangle$ is satisfied.

Below we show that the above characterization can be captured by a Streett objective, thus there is a Streett objective equivalent to ψ. Since Player 2 wins a Streett game iff she has a memoryless winning strategy [19, 25], the result for $\mathcal{G}$ follows.

Let α' and α'' be the generalized Büchi objectives equivalent to ψ_1 and ψ_2, respectively.

We define the Streett objective $\psi' = \{\langle V, \alpha'_l\rangle : \alpha'_l \in \alpha'\} \cup \{\langle \alpha_m, \alpha''_l\rangle : \alpha''_l \in \alpha''\}$. Note that the Streett objective $\{\langle V, \alpha'_l\rangle : \alpha'_l \in \alpha'\}$ is equivalent to the generalized Büchi objective α', and that the Streett objective $\{\langle \alpha_m, \alpha''_l\rangle : \alpha''_l \in \alpha''\}$ is satisfied iff α_m is not satisfied or the generalized Büchi objective α'' is satisfied.

We prove that the objectives ψ and ψ' are equivalent. Consider a play ρ in G, and assume first that ψ is satisfied in ρ. Note that the generalized Büchi objective α' is satisfied in ρ, and thus also the Streett objective $\{\langle V, \alpha'_l\rangle : \alpha'_l \in \alpha'\}$. If α_m is not satisfied in ρ, then the Streett objective $\{\langle \alpha_m, \alpha''_l\rangle : \alpha''_l \in \alpha''\}$ is satisfied since vertices in α_m are visited only finitely often. Otherwise, α'' is satisfied in ρ, and thus also the Streett objective $\{\langle \alpha_m, \alpha''_l\rangle : \alpha''_l \in \alpha''\}$ is satisfied. Therefore, ψ' is satisfied in ρ.

For the second direction, assume ψ' is satisfied in ρ. Note that this implies α' is satisfied in ρ as well. If α_m is not satisfied, then the satisfaction of α' implies the satisfaction of ψ. If α_m is satisfied in ρ, we have that α'' is satisfied in ρ. Therefore, the sum of weights of satisfied objectives is at least $t - w(\alpha_m) + w(\alpha_m) = t$. □

One may be tempted to generalize the result in Theorem 9 to a fixed number of objectives that may have a negative weight. With more than one negative objective, however, a winning strategy of Player 2 may need to direct the outcome of an interaction into different successors of a vertex that repeats in the outcome. For a specific example, see Lemma 3.

Lemma 3. *There is a zero-sum BMaxWB game $\mathcal{G}$ with an additive weight function in which exactly two objectives have negative weights and such that Player 2 wins $\mathcal{G}$, but Player 2 does not have a memoryless winning strategy in $\mathcal{G}$.*

Proof. We define a zero-sum BMaxWB game $\mathcal{G} = \langle G, \{\alpha_1, \alpha_2\}, w, t\rangle$ with $w(\alpha_1), w(\alpha_2) < 0$ such that Player 2 wins $\mathcal{G}$, but Player 2 does not have a memoryless winning strategy in $\mathcal{G}$.

Intuitively, the game is defined so Player 2 needs to satisfy both objectives α_1 and α_2 in order to win, and satisfying both objectives requires memory.

The game graph G is defined as follows. From the initial vertex v_0, Player 2 chooses between proceeding to a vertex a, and proceeding to a vertex b. From both a and b, the game returns to the initial vertex. Then, we define $\alpha_1 = \{a\}$, $\alpha_2 = \{b\}$, $w(\alpha_1) = w(\alpha_2) = -1$, and $t = -1$. It is easy to see that Player 2 has a winning strategy that satisfies both objectives α_1 and α_2, and that every strategy for Player 2 that satisfies both objectives require memory.

Theorem 9 enables us to lift the "polynomial DNE solution" property proved in Theorem 4 for non-decreasing weight functions to almost-positive weight functions. In order to also lift the NP solution in Theorem 5, we first need to find a solution in NP for zero-sum games. Guessing a memoryless strategy f_2 is a first step for that, but one also needs to check f_2 in polynomial time. The fact only one objective $\alpha_l \in \alpha$ may get a negative weight makes it possible. Indeed, we only need to reason about the *maximal* strongly connected sets of the graph induced by f_2, and its restrictions to vertices not in α_l, which can be done in linear time [37].

Theorem 10. *Deciding whether Player 2 wins a zero-sum MaxWB game with an almost-positive weight function is NP-complete.*

Proof. We start with the upper bound. Consider a zero-sum MaxWB game $\mathcal{G} = \langle G, \psi\rangle$ with $\psi = \langle \alpha, w, t\rangle$, $\alpha = \{\alpha_1, \ldots, \alpha_m\}$, and w is almost positive. Let l be such that $w(\alpha_l) < 0$. An NP algorithm guesses a memoryless strategy f_2 for Player 2, calculates the sub-graph G_{f_2} of G such that edges from vertices in V_2 agree with f_2, and checks that there does not exist a play in G_{f_2} that satisfies ψ. The check proceeds as follows.

First, we remove from G_{f_2} the vertices in α_l and check whether the obtained graph has a maximal strongly connected set U with $w(\{\alpha_j \in \alpha : U \cap \alpha_j \neq \emptyset\}) \geq t$. If there is such a set, we know that there is a play – one that reaches U and then traverses all the vertices in U infinitely often, that satisfies ψ. Otherwise, we check whether the graph G_{f_2} has a maximal strongly connected set U with $w(\{\alpha_j \in \alpha : U \cap \alpha_j \neq \emptyset\}) \geq t$. That is, we repeat the same check, now with G_{f_2} itself. If there is such a component, we again know that there is a play that satisfies ψ. Also, if no set U is found in the two checks, we can conclude that no a play in G_{f_2} satisfies ψ. Indeed, since all the objectives in $\alpha \setminus \alpha_l$ have positive weights, restricting attention to *maximal* strongly connected sets is sound, and leads to a linear algorithm [37].

The lower bound follows from the NP-hardness of deciding whether Player 2 wins a zero-sum BMaxWB game with positive and additive weight functions [28]. ☐

The considerations in the proof of Theorem 5 can now be applied for games with almost-positive weight functions. In particular, note that the weight function used in the lower bound there is positive and additive, and hence also almost-positive.

Theorem 11. *Given a k-player MaxWB game $\mathcal{G}$ with almost-positive weight functions, a set $S \subseteq [k]$ of system players, and a utility predicate $P \subseteq \mathbb{Z}^{[k]}$, deciding whether there exists a DNE solution for $\langle \mathcal{G}, S, P \rangle$ is NP-complete. Hardness in NP already applies for $k = 2$.*

6.2 Additive Weight Functions with Polarities

Consider a MaxWB game $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i \in [k]} \rangle$ with an additive weight function. Let $\alpha = \{\alpha_1, \ldots, \alpha_m\}$. For an objective $\alpha_l \in \alpha$ and a set $A \subseteq [k]$ of players, we say that α_l is *good for* A if for all $i \in A$, we have that $w_i(\alpha_l) > 0$. Likewise, α_l is *bad for* A if $w_i(\alpha_l) < 0$ for all $i \in A$. Finally, α_l is *polar for* A if α_l is good or bad for A.

Consider a partition $A, B \subseteq [k]$ of the players in $[k]$; thus $A \cup B = [k]$ and $A \cap B = \emptyset$. We say that (A, B) is a *natural partition* (to coalitions) of $[k]$ if every objective $\alpha_l \in \alpha$ is polar for A and for B, with dual polarity. That is, either α_l is good for A and bad for B, or α_l is bad for A and good for B

Games with a natural partition of $[k]$ to coalitions may seem weaker than general games. Indeed, if (A, B) is a natural partition, it is tempting to merge all the players in A to a single player, aiming for the satisfaction of objectives with a positive polarity for them, and merge all the players in B to a single player, aiming for the satisfaction of the complementary set of objectives. Thus, games with a natural partition to coalitions seem related to zero-sum games.

As we show below, however, the relative weights of the different objectives play a role that is more significant than their polarity. Using these relative weights, we can turn each MaxWB game $\mathcal{G}$ to an equivalent game that has a natural partition to coalitions. Moreover, given *any* partition (A, B) of the players, we can turn $\mathcal{G}$ to an equivalent games in which (A, B) is a natural partition to coalitions. Thus, games with natural partitions are by no means easier than general ones.

Theorem 12. *Consider a MaxWB k-player game* $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i\in[k]}\rangle$ *with an additive weight function. For every partition* (A, B) *of* $[k]$*, there is a game* $\mathcal{G}'$ *equivalent to* $\mathcal{G}$ *such that* (A, B) *is a natural partition to coalitions in* $\mathcal{G}'$.

Proof. Consider a MaxWB k-player game $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i\in[k]}\rangle$, and a partition (A, B) of $[k]$. Let $m = |\alpha|$. We construct a MaxWB game $\mathcal{G}'$ equivalent to $\mathcal{G}$ such that (A, B) is a natural partition in $\mathcal{G}'$. The game $\mathcal{G}' = \langle G, \alpha', \{w'_i\}_{i\in[k]}\rangle$ is defined as follows (see Example 2).

Example 2. Consider the MaxWB 3-player game $\mathcal{G} = \langle G, \{\alpha_1, \alpha_2, \alpha_3\}, \{w_1, w_2, w_3\}\rangle$, where the weights functions are described in the table below (left).

	α_1	α_2	α_3
w_1	$+1$	$+1$	-1
w_2	$+1$	-1	$+1$
w_3	-1	$+1$	-1

	α_1^1	α_1^2	α_2^2	α_2^2	α_3^1	α_3^2
w'_1	$+2$	-1	$+2$	-1	$+1$	-2
w'_2	-1	$+2$	-2	$+1$	-1	$+2$
w'_3	-2	$+1$	-1	$+2$	-2	$+1$

The partition $(\{1\}, \{2, 3\})$ is not a natural partition of $\mathcal{G}$. On the table (right), we describe the weight functions w'_1, w'_2, and w'_3, on the two copies of α and with which $(\{1\}, \{2, 3\})$ is a natural partition. Note that for every $i, l \in \{1, 2, 3\}$, we have that $w_i(\alpha_l) = w'_i(\alpha_l^2) + w'_i(\alpha_l^2)$. □

First, α' consists of two copies[2] of α. For $l \in [m]$, let α_l^1 and α_l^2 denote the two copies of α_l in α'. Then, for every player $i \in A$, we define the weight function $w'_i : \alpha' \to \mathbb{Z}$ as follows. Consider an objective $\alpha_l \in \alpha$.

1. If $w_i(\alpha_l) \leq 0$, then $w'_i(\alpha_l^1) = 1$ and $w'_i(\alpha_l^2) = w_i(\alpha_l) - 1$.
2. If $w_i(\alpha_l) > 0$, then $w'_1(\alpha_l^1) = w_i(\alpha_l) + 1$ and $w'_i(\alpha_l^2) = -1$.

Finally, for every player $i \in B$, we define the weight function $w'_i : \alpha' \to \mathbb{Z}$ as follows. Consider an objective $\alpha_l \in \alpha$.

1. If $w_i(\alpha_l) \leq 0$, then $w'_i(\alpha_l^1) = w_i(\alpha_l) - 1$ and $w'_i(\alpha_l^2) = 1$.
2. If $w_i(\alpha_l) > 0$, then $w'_i(\alpha_l^1) = -1$ and $w'_i(\alpha_l^2) = w_i(\alpha_l) + 1$.

Note that for all $l \in [m]$, the objective α_l^1 is good for A and bad for B, and the objective α_l^2 is bad for A and good for B. Also, $w_i(\alpha_l) = w'_i(\alpha_l^1) + w'_i(\alpha_l^2)$, and so we can express with w' the same utilities in $\mathcal{G}'$ expressed with w in $\mathcal{G}$. □

7 Non-zero-Sum Games with Payments

In this section, we study non-zero-sum games in which each player $i \in [k]$ may commit to pay the other players a certain amount for each of the underlying objectives that are satisfied. Clearly, such payments may incentive the other players to choose strategies that are preferable for Player i.

[2] That is, α' as a multi-set. Avoiding a multi-set is possible, but involves a duplication of vertices.

We first study the setting for Büchi objectives, and then extend to MaxWB objectives. Consider a k-player Büchi game $\mathcal{G} = \langle G, \{\alpha_i\}_{i\in[k]}, \{R_i\}_{i\in[k]}\rangle$. A *payment function* $p : [k] \times [k] \to \mathbb{N}$ maps each two players $i, j \in [k]$ to the amount Player i commits to pay Player j when α_i is satisfied. We require that $p(i,i) = 0$. When Player i does not commit to pay more than her reward, thus $R_i \geq \sum_{j\in[k]} p(i,j)$, we say that p is *positive*. A k-player Büchi game with payments is a pair $\langle G, p\rangle$, for a Büchi game $\mathcal{G}$ and a payment function p.

The utility of a player in a game with payments $\langle \mathcal{G}, p\rangle$ combines her reward in $\mathcal{G}$ with the payments she pays and receives from the other players. Formally, for a profile π, let $W(\pi) = \{j \in [k] : \alpha_j \in \mathsf{sat}(\mathsf{Outcome}(\pi), \alpha)\}$. Thus, $W(\pi)$ is the set of objectives satisfied in π, which is also the set of players that has to fulfill the payment commitments to the other players. Then, the utility of Player i in π is

$$\mathsf{util}_i(\pi) = \begin{cases} R_i - \sum_{j\in[k]} p(i,j) + \sum_{j\in W(\pi)} p(j,i) & \text{if } i \in W(\pi), \\ \sum_{j\in W(\pi)} p(j,i) & \text{if } i \notin W(\pi). \end{cases}$$

Stability in games with payments is defined with respect to the utilities that take payments into account. Note that the payment function is fixed and payments are not part of the player's strategies. The DNE problem for games with payments gets as input a game $\mathcal{G}$, a payment function p, a set of players $S \subseteq [k]$, and a utility predicate $P \subseteq \mathbb{Z}^{[k]}$, and has to return an S-fixed NE in $\langle \mathcal{G}, p\rangle$ that satisfies P.

Theorem 13. *For every k-player Büchi game $\mathcal{G}$ and payment function $p : [k] \times [k] \to \mathbb{N}$, there is an equivalent k-player MaxWB game $\mathcal{G}'$ with almost-positive additive weight functions. If p is positive, then $\mathcal{G}'$ has positive additive weight functions.*

Proof. Let $\mathcal{G} = \langle G, \alpha, \{R_i\}_{i\in[k]}\rangle$. We define $\mathcal{G}' = \langle G, \alpha, \{w_i\}_{i\in[k]}\rangle$, where for every $i \in [k]$, we have that $w_i(\alpha_i) = R_i - \sum_{j\in[k]} p(i,j)$, and for all $l \in [k] \setminus \{i\}$, we have that $w_i(\alpha_l) = p(l,i)$.

The correctness of the construction follows immediately from the definition of the utilities in games with payments. Indeed, for every profile π in G, the utilities of Player i in π coincide in $\langle \mathcal{G}, p\rangle$ and $\mathcal{G}'$. ☐

Remark 1. **[MaxWB games with payments]** The definition of games with payments as well as the reduction in Theorem 13 can be easily extended to MaxWB games. The same way we study different classes of weight functions, one can define different classes of payment functions in MaxWB games. Consider a game $\mathcal{G} = \langle G, \alpha, \{w_i\}_{i\in[k]}\rangle$. The most general payment function is $p : [k] \times [k] \times 2^{\alpha} \to \mathbb{N}$, mapping each two players $i, j \in [k]$ and set $X \subseteq \alpha$ of objectives to the amount Player i pays Player j when exactly all the objectives in X are satisfied. A payment function is additive if it induced by a function $p : [k] \times [k] \times \alpha \to \mathbb{N}$, which describe payments for single objectives.

We show how the construction in Theorem 13 can be extended to general MaxWB games. There, a payment function for a k-player MaxWB game with m underlying objectives and weight functions $\{w_i\}_{i\in[k]}$ is $p : [k] \times [k] \times [m] \to \mathbb{N}$, where $p(i,j,l)$ describes how much Player i commits to pay Player j in case α_l is satisfied. We say that p is positive if $\sum_{j\in[k]}\sum_{\alpha_l\in X} p(i,j,l) \leq w_i(X)$, for every $i \in [k]$ and $X \subseteq \alpha$.

The weight functions in the proof of Theorem 13 are then $\{w'_i\}_{i\in[k]}$ with $w'_i(X) = w_i(X) - \sum_{\alpha_l\in X}\sum_{j\in[k]} p(i,j,l)$, for every $i \in [k]$ and $X \subseteq \{\alpha_1,\ldots,\alpha_m\}$. Note that if the weight functions in $\mathcal{G}$ are additive, then the weight functions in $\mathcal{G}'$ are additive and positive if p is positive, and if the weight functions in $\mathcal{G}$ are non-decreasing and p is positive, the weight functions in $\mathcal{G}'$ are positive and non-decreasing.

It follows that the complexity results in Theorems 4 and 5 apply also for the case of MaxWB games with payments for non-decreasing weight functions and a positive payment function. Also, the complexity results for general and non-positive additive weight functions apply also for the case of MaxWB games with payments with the same type of weight functions and general payment functions.

Note however that when the payment function is non-positive, MaxWB games with payments and positive additive weight functions are equivalent to MaxWB games with non-positive additive weight functions, and games with non-decreasing weight functions are equivalent to MaxWB games with general weight function. In particular, recall the zero-sum game $\mathcal{G}$ from the lower bound proof of Theorem 7, and note that there exists a MaxWB game with payments and a positive additive weight functions $\langle \mathcal{G}', p\rangle$, such that there exists a DNE solution for $\langle \mathcal{G}, \{1\}, \{\mathsf{util}_1 \geq t\}\rangle$ in $\langle \mathcal{G}', p\rangle$ iff Player 1 wins $\mathcal{G}$. Specifically, the weight functions for the players are empty, thus $w_1(\alpha_l) = w_2(\alpha_l) = 0$, for every objective $\alpha_l \in \alpha$, and the payment function p mimics the weight function of $\mathcal{G}$, thus $p(1,2,l) = i$, for every existentially-quantified variable x_i and a literal l with index i, and $p(2,1,l) = i$, for every universally-quantified variable x_i and a literal l with index i. $\square$

The DNE problem for Büchi games without payments can be solved in polynomial time [38][3]. Adding payments involves a transition to the DNE problem for MaxWB games with almost-positive weight functions, which has a computations cost. In Theorem 14 below, we show that this cost is unavoidable, in fact, already for the special case of cooperative rational synthesis.

Theorem 14. *Given a k-player Büchi game with payments $\langle \mathcal{G}, p\rangle$, a set $S \subseteq [k]$ of system players, and a utility predicate $P \subseteq \mathbb{Z}^{[k]}$, deciding whether there exists a DNE solution for $\langle \mathcal{G}, p, S, P\rangle$ is NP-complete. NP-hardness applies already when p is positive, $S = \{1\}$, and $P = \{x_1 \geq 1\}$.*

Proof. The upper bound follows from Theorem 13 and Theorem 5.

For the lower bound, we describe a reduction from 3SAT. Given a 3CNF formula φ, we construct a Büchi game $\mathcal{G}$ and a positive payment function p such that there exists a DNE solution for $\langle \mathcal{G}, p, \{1\}, u_1 \geq 1\rangle$ iff φ is satisfiable.

[3] The study in [38] is for cooperative rational synthesis with Boolean objectives, but can be easily extended to the DNE problem.

Consider a set of variables $X = \{x_1, \ldots, x_n\}$, and let $\overline{X} = \{\overline{x_1}, \ldots, \overline{x_n}\}$. Also consider a formula $\varphi = C_1 \wedge \ldots \wedge C_k$, with $C_i = (l_i^1 \vee l_i^2 \vee l_i^3)$ and $l_i^1, l_i^2, l_i^3 \in X \cup \overline{X}$, for every $i \in [k]$.

Intuitively, we define the game with payments $\langle \mathcal{G}, p \rangle$ as a $(2n+2)$-player game with Player 1, Player 2, and literal players Player l for every $l \in X \cup \overline{X}$, so only Player 1 and Player 2 control vertices in the game graph, and Player 2 chooses to help Player 1 to have utility 1 iff at most n literal players pay Player 2 when she does not help Player 1. When Player 2 does not help Player 1, Player 1 is required to choose an assignment to the variables in X, and then prove that the chosen assignment satisfies φ. There, for every literal $l \in X \cup \overline{X}$, the objective of Player l is satisfied iff Player 1 states that l is evaluated to **true**, either as part of the chosen assignment or as part of her proof. Each literal player offers to pay Player 2 when her objective is satisfied, and accordingly, Player 1 can ensure that at most n literal players pay Player 2 iff there exists an assignment that satisfies φ.

The game graph G proceeds as follows. From the initial vertex, Player 2 chooses between proceeding to a self-looped sink $\bot$ in which the objective of Player 1 is satisfied and accordingly Player 1 gets utility of 1, and proceeding to an *assignment sub-graph*. In the assignment sub-graph, Player 1 chooses an assignment to every variable in X, and then chooses for every clause of φ one of its literals, essentially stating that the literal is evaluated to **true** in the chosen assignment, and the process repeats infinitely often. Choosing an assignment to a variable x_i involves choosing between proceeding to a vertex that corresponds to the literal x_i, and a vertex that corresponds to the literal $\overline{x_i}$. Choosing a literal l_i^j for the clause C_i involves proceeding to a vertex that correspond to the literal l_i^j. Note that only Player 1 and Player 2 own vertices in G.

The objective of Player 1 and Player 2 is to reach $\bot$, and the objective of Player l is to visit infinitely often vertices that correspond to the literal l in the assignment sub-graph, for every $l \in X \cup \overline{X}$. Then, Player 1 gets reward of 1, Player 2 gets reward of n, and Player l gets reward for 2, for every $l \in X \cup \overline{X}$. Also, every Player l offers to pay 1 to Player 2 when her objective is satisfied. That is, Player l pays 1 to Player 2 when the play proceeds to the assignment sub-graph, and visit vertices that correspond to l infinitely often.

Note that by definition, every play in the assignment sub-graph satisfies at least n objectives of literal players, and there exists a play that satisfies exactly n objectives of literal players iff φ is satisfiable. Indeed, for a satisfying assignment, Player 1 can generate a play that only visits vertices that correspond to literals that are evaluated to **true** in the satisfying assignment, and thus satisfy exactly n objectives of literal players. Specifically, satisfy the objectives of Player l for every literal l that is evaluated to **true** in the assignment. And, when there does not exist a satisfying assignment, when choosing literals for every clause, Player 1 is forced to visit both a vertex that correspond to l and a vertex that correspond to $\overline{l}$, for some literal $l \in X \cup \overline{X}$. Therefore, when φ is not satisfiable, every play in the assignment sub-graph satisfies the objectives of at least $n + 1$ literal players.

Since every literal player offers to pay 1 to Player 2 when her objective is satisfied, we have that Player 2 benefits from proceeding to the assignment sub-graph more than proceeding to $\bot$ iff at least $n+1$ objectives of literal players are satisfied there, and Player 1 can force that at most n objectives of literal players are satisfied iff φ is satisfiable. Accordingly, there exists a 1-fixed NE in which Player 2 proceed to $\bot$, and thus the utility for Player 1 is 1 iff φ is satisfiable.

Formally, $\langle \mathcal{G}, p \rangle$ is defined as follows. First, the $(2n+2)$-player Büchi game $\mathcal{G} = \langle G, \{\alpha_1, \alpha_2\} \cup \{\alpha_l\}_{l \in X \cup \overline{X}}, \{R_1, R_2\} \cup \{R_l\}_{l \in X \cup \overline{X}} \rangle$ contains the following components.

1. The game graph $G = \langle V_1, V_2, v_0, E \rangle$ contains the following components.
   (a) $V_1 = \{\bot\} \cup \{v_i : i \in [n]\} \cup X \cup \overline{X} \cup \{C_i : i \in [k]\} \cup \{l_i^1, l_i^2, l_i^3 : i \in [k]\}$. The vertices in $\{v_i : i \in [n]\}$,$X \cup \overline{X}$,$\{C_i : i \in [k]\}$,$\{l_i^1, l_i^2, l_i^3 : i \in [k]\}$ are *variable*, *literal*, *clause* and *clause-literal* vertices, respectively.
   (b) $V_2 = \{v_0\}$.
   (c) The set of edges E contains the following edges.
      i. $\langle v_0, v_1 \rangle$, $\langle v_0, \bot \rangle$, and $\langle \bot, \bot \rangle$. That is, from the initial vertex, Player 2 chooses between proceeding to the assignment sub-graph, and to $\bot$, in which case the play stays in $\bot$ indefinitely.
      ii. $\langle v_i, x_i \rangle$ and $\langle v_i, \overline{x_i} \rangle$, for every $i \in [n]$. That is, Player 1 assigns **true** to x_i by proceeding to the literal vertex x_i, and assigns **true** by proceeding to the literal vertex $\overline{x_i}$.
      iii. $\langle x_i, v_{i+1} \rangle$ and $\langle \overline{x_i}, v_{i+1} \rangle$, for every $1 \leq i < n$.
      iv. $\langle x_n, C_1 \rangle$ and $\langle \overline{x_n}, C_1 \rangle$.
      v. $\langle C_i, l_i^j \rangle$, for every $i \in [k]$ and $j \in \{1, 2, 3\}$. That is, Player 1 states that the literal l_i^j is evaluated to **true** in the chosen assignment by proceeding from C_i to the clause-literal vertex l_i^j.
      vi. $\langle l_i^j, C_{i+1} \rangle$, for every $1 \leq i < k$ and $j \in \{1, 2, 3\}$.
      vii. $\langle l_k^j, v_1 \rangle$, for every $j \in \{1, 2, 3\}$.
2. The objectives for the players are defined as follows.
   (a) $\alpha_1 = \alpha_2 = \{\bot\}$.
   (b) $\alpha_l = \{l\} \cup \{l_i^j : i \in [k], j \in \{1, 2, 3\}, \text{ and } l_i^j = l\}$, for every $l \in X \cup \overline{X}$. That is, the objective of Player l is to visit vertices that correspond to the literal l infinitely often.
3. The rewards for the players are defined as follows.
   (a) $R_1 = 1$.
   (b) $R_2 = n$.
   (c) $R_l = 2$, for every $l \in X \cup \overline{X}$.

The payment function p has $p(l, 2) = 1$, for every $l \in X \cup \overline{X}$, and $p(i, j) = 0$ for all $i \notin X \cup \overline{X}$ or $j \neq 2$. That is, every literal player pays 1 to Player 2 when α_l is satisfied, and these are the only payments.

We prove the correctness of the construction.

Note that the utility for Player 1 is 1 only in plays that reach $\bot$. Thus, it is sufficient to show that φ is satisfiable iff there exists a 1-fixed NE whose outcome reaches $\bot$.

For the first direction, assume φ is satisfiable. Consider a profile π whose outcome reaches $\bot$, and the strategy for Player 1 in the assignment sub-graph visits infinitely often literal vertices only if they correspond to literals evaluated to **true** in the satisfying assignment. Since there exactly n such literals, there are n literal players that their objectives are satisfied in the play in the assignment sub-graph, thus Player 2 does not benefit from deviating. Indeed, her utility in π is n from her reward, and by deviating she would get the same utility from the payments from the other players.

For the second direction, assume φ is not satisfiable. Note that it implies that every play in the assignment sub-graph satisfies at least $n+1$ different objectives from $\{\alpha_l : l \in X \cup \overline{X}\}$. Thus, there does not exist a 1-fixed NE in which Player 2 proceeds from the initial vertex to $\bot$. Indeed, by deviating, she increases her utility to at least $n+1$. □

7.1 Monetary-based repair of Büchi games

Consider a k-player Büchi game $\mathcal{G} = \langle G, \{\alpha_i\}_{i\in[k]}, \{R_i\}_{i\in[k]}\rangle$. For a payment function $p : [k] \times [k] \to \mathbb{N}$ and a set $S \subseteq [k]$ of system players, we say that p incentivizes $[k]\setminus S$ if $p(i,j) = 0$ for all $i \in [k]\setminus S$ and $j \in [k]$. That is, only system players may suggest payments.

In the *monetary-based system-repair* problem, we are given a k-player game $\mathcal{G}$, a set of system players $S \subseteq [k]$, and a utility predicate $P \subseteq \mathbb{Z}^{[k]}$, and we seek a payment function that incentivizes $[k] \setminus S$ to follow strategies with which P is satisfied. Formally, a *repair solution* for $(\mathcal{G}, S, P)$ is a pair (p, π), where p is a payment function p that incentivizes $[k] \setminus S$ and π is a DNE solution for $\langle\langle\mathcal{G}, p\rangle, S, P\rangle$. Note that since the system players are under our control (formally, π is an S-fixed NE), they need not be incentivized. Still, p may include transfers within the system players in order to satisfy P.

Theorem 15. *Given a k-player game $\mathcal{G}$, a set $S \subseteq [k]$ of system players, and a utility predicate $P \subseteq \mathbb{Z}^{[k]}$, deciding whether there exists a repair solution for $\langle\mathcal{G}, S, P\rangle$ is NP-complete. Hardness in NP already applies when $S = 1$ and $P = \{u_1 \geq 1\}$.*

Proof. For the upper bound, an NP algorithm guesses a payment function p that incentivizes $[k] \setminus S$, and then finds a DNE solution for $\langle\mathcal{G}, p, S, P\rangle$, as described in Theorem 14. Note that the sum of payments that a given player offers is bounded by the sum of rewards of all the other players in the game, thus it is sufficient to guess a payment function that is polynomial in the input.

We continue to the lower bound. We describe a reduction from 3SAT. That is, given a formula $\varphi = C_1 \wedge \ldots \wedge C_k$ with $C_i = (l_i^1 \vee l_i^2 \vee l_i^3)$, for every $i \in [k]$, over a set of variables $X = \{x_1, \ldots, x_n\}$, we construct a Büchi game $\mathcal{G}$ such that there exists a repair solution for $\langle\mathcal{G}, \{1\}, u_1 \geq 1\rangle$ iff φ is satisfiable.

Intuitively, we define a $(2n+1)$-player game $\mathcal{G}$ with Player 1 as the single system player and literal players Player l, for every $l \in X \cup \overline{X}$, so the literal players choose to help Player 1 to satisfy her objective and have utility of at least 1 iff Player 1 can pay n literal players such that only those players can satisfy their objectives by not helping her. Similar to the previous reduction, when the literal players do not help Player 1, Player 1 is required to choose an assignment to the variables in X and prove that the chosen assignment satisfies φ, and Player l wins there iff Player 1 states the literal l is evaluated to **true**. We then define the reward for Player 1 so she can afford to pay at most n literal players and still have a utility of at least 1. Accordingly, Player 1 can ensures her utility is at least 1 iff φ is satisfiable.

The game graph G proceeds as follows. Let G_φ be the assignment sub-graph from the proof of Theorem 14, and recall that this is a game graph in which all the vertices are controlled by Player 1, that chooses assignments to every variable in X, then chooses a literal for every clause in φ, and repeats the process infinitely often.

From the initial vertex, the play traverses a string of vertices $\bot_l$, controlled by Player l, for every literal $l \in X \cup \overline{X}$. From $\bot_l$, Player l chooses between proceeding to the assignment sub-graph G_φ, or proceeding to the next vertex in the string. At the last $\bot_l$ vertex, Player l chooses between proceeding to G_φ, and proceeding to a self-looped sink $\bot$.

The objective of Player 1 is to reach $\bot$, and the objective of the literal player Player l is α_l from the proof of Theorem 14. That is, α_l consists of all the vertices that correspond to the literal l in G_φ. Then, the rewards for the players are $n+1$ for Player 1, and 1 for every literal player. Note that the utility for Player 1 is at least 1 only in plays that reach $\bot$. Intuitively, there exists a payment function p and a 1-fixed NE in $\langle \mathcal{G}, p\rangle$ such that the utility for Player 1 can be at least 1 iff the play in the assignment sub-graph satisfies the objectives of exactly n players. Since Player 1 can generate a play in the assignment sub-graph that satisfies the objectives of exactly n players iff φ is satisfiable, it would then follow that there exists a repair solution for $\langle \mathcal{G}, \{1\}, \{u_1 \geq 1\}\rangle$ iff φ is satisfiable.

Formally, $\mathcal{G} = \langle G, \{\alpha_1\} \cup \{\alpha_l\}_{l \in X \cup \overline{X}}, \{R_1, R_2\} \cup \{R_l\}_{l \in X \cup \overline{X}}\rangle$ contains the following components.

1. Recall the assignment game graph $G_\varphi = \langle V_1, \emptyset, v_1, E\rangle$ from the proof of Theorem 14. The game graph $G = \langle \{V_1 \cup \{v_0, \bot\}\} \cup \{V_l\}_{l \in X \cup \overline{X}}, v_0, E'\rangle$ contains the following components.
   (a) The set of vertices owned by Player 1 is $V_1 \cup \{v_0, \bot\}$, and $V_l = \{\bot_l\}$, for every $l \in X \cup \overline{X}$.
   (b) The set of edges E' contains E, and the following edges.
      i. $\langle v_0, \bot_{x_1}\rangle$. That is, from the initial vertex, the play proceed to traverse the $\bot_l$ vertices.
      ii. $\langle \bot_{x_i}, \bot_{\overline{x_i}}\rangle$, for every $i \in [n]$.
      iii. $\langle \bot_{\overline{x_i}}, \bot_{x_{i+1}}\rangle$, for every $1 \leq i < n$. $\langle \bot_{\overline{x_n}}, \bot\rangle$.
      iv. $\langle \bot_l, v_1\rangle$, for every $l \in X \cup \overline{X}$. That is, from $\bot_l$, Player l chooses between proceeding to the next $\bot$ vertex, and proceeding to the assignment sub-graph.
      v. $\langle \bot, \bot\rangle$.
2. The objectives for the players are defined as follows.

(a) $\alpha_1 = \{\bot\}$.
   (b) $\alpha_l = \{l\} \cup \{l_i^j : i \in [k], j \in \{1,2,3\},$ and $l_i^j = l\}$, for every $l \in X \cup \overline{X}$. That is, the objective of Player l is to visit vertices that correspond to the literal l infinitely often.
3. The rewards for the players are defined as follows.
   (a) $R_1 = n + 1$.
   (b) $R_l = 1$, for every $l \in X \cup \overline{X}$.

We prove the correctness of the construction.

Note that the utility for Player 1 is at least 1 only in plays that reach $\bot$. Also note that for every payment function p, a profile π in $\langle \mathcal{G}, p \rangle$ that reaches $\bot$ is a 1-fixed NE iff Player 1 pays every player that her objective is satisfied in the play in the assignment sub-graph. Thus, there exist such payment function p and 1-fixed NE iff Player 1 can generate a play in the assignment sub-graph that satisfies the objectives of (at most) n players. Since there exists such a play iff φ is satisfiable, the correctness of the construction follows. □

8 Discussion

general	non-decreasing	additive	almost positive	additive + polarities
PSPACE (Th. 3)	NP (Th. 5)	PSPACE (Th. 8)	NP (Th. 10)	PSPACE (Th. 12)

Table 4. The complexity of deciding if there exists a DNE solution for $\langle \mathcal{G}, S, P \rangle$ for a game $\mathcal{G}$ with some restriction on the weight function.

A major challenge in automated synthesis is the design of *high-quality* systems. The specification of such systems combines different aspects of the behavior of the system. Multiple weighted objectives enable a rich and convenient way to specify these combinations. We showed that the different classes of weight functions offer a hierarchy of expressiveness and complexity that is different from the one induced by the different classes of ω-regular objectives, and that is very relevant in the context of high-quality synthesis. In particular, it brings the analysis closer to the one used in classic game theory (c.f., [34] Chapter 11). We showed that the different classes of weight functions offer a hierarchy of expressiveness and complexity that is different from the one induced by the different classes of ω-regular objectives, and that is very relevant in the context of high-quality synthesis. In particular, it brings the analysis closer to the one used in classic game theory (c.f., [34] Chapter 11).

From a practical point of view, multiple weighted objectives also make it possible to calibrate expenses on different resources. In particular, it enables reasoning about settings in which *money* can be used as a resource. Such an approach is used, for example, when the system is composed from a library of priced components [6, 30], or when agents need to pay in order to sense a signal

[15] or to take an action, as in bidding games [5] or networks with tolls [29]. Here, we suggested to use payments in order to repair systems, by incentivizing the environment to follow a strategy in which the objective of the system is satisfied. It is interesting to study additional types of monetary-based repair in non-zero-sum games, for example when payments are used in order to buy control [27] or generate coalitions [11].

References

1. S. Almagor, G. Avni, and O. Kupferman. Repairing multi-player games. In *Proc. 26th Int. Conf. on Concurrency Theory*, volume 42 of *LIPIcs*, pages 325–339, 2015.
2. S. Almagor, U. Boker, and O. Kupferman. Formalizing and reasoning about quality. In *Proc. 40th Int. Colloq. on Automata, Languages, and Programming*, volume 7966 of *Lecture Notes in Computer Science*, pages 15 – 27. Springer, 2013.
3. S. Almagor and O. Kupferman. High-quality synthesis against stochastic environments. In *Proc. 25th Annual Conf. of the European Association for Computer Science Logic*, volume 62 of *LIPIcs*, pages 28:1–28:17, 2016.
4. S. Almagor, O. Kupferman, and G. Perelli. Synthesis of controllable Nash equilibria in quantitative objective game. In *Proc. 27th Int. Joint Conf. on Artificial Intelligence*, pages 35–41, 2018.
5. G. Avni, T. A. Henzinger, and V. Chonev. Infinite-duration bidding games. In *Proc. 28th Int. Conf. on Concurrency Theory*, volume 85 of *LIPIcs*, pages 21:1–21:18, 2017.
6. G. Avni and O. Kupferman. Synthesis from component libraries with costs. *Theor. Comput. Sci.*, 712:50–72, 2018.
7. R. Bloem, K. Chatterjee, T. Henzinger, and B. Jobstmann. Better quality in synthesis through quantitative objectives. In *Proc. 21st Int. Conf. on Computer Aided Verification*, volume 5643 of *Lecture Notes in Computer Science*, pages 140–156. Springer, 2009.
8. R. Bloem, K. Chatterjee, and B. Jobstmann. Graph games and reactive synthesis. In *Handbook of Model Checking.*, pages 921–962. Springer, 2018.
9. 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.
10. 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.
11. R. Brenguier. Robust equilibria in mean-payoff games. In *Proc. 19th Int. Conf. on Foundations of Software Science and Computation Structures*, volume 9634 of *Lecture Notes in Computer Science*, pages 217–233. Springer, 2016.
12. V. Bruyère, C. Grandmont, and J-F Raskin. As soon as possible but rationally. In *Proc. 35th Int. Conf. on Concurrency Theory*, volume 311 of *LIPIcs*, pages 14:1–14:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
13. V. Bruyère, Q. Hautem, and J-F Raskin. Parameterized complexity of games with monotonically ordered ω-regular objectives. In *Proc. 29th Int. Conf. on Concurrency Theory*, volume 118 of *LIPIcs*, pages 29:1–29:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

14. K. Chatterjee, A.K. Goharshady, and Y. Velner. Quantitative analysis of smart contracts. In *27th European Symposium on Programming Languages and Systems*, volume 10801 of *Lecture Notes in Computer Science*, pages 739–767. Springer, 2018.
15. 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.
16. K. Chatterjee, R. Majumdar, and M. Jurdzinski. On Nash equilibria in stochastic games. In *Proc. 13th Annual Conf. of the European Association for Computer Science Logic*, volume 3210 of *Lecture Notes in Computer Science*, pages 26–40. Springer, 2004.
17. R. Condurache, E. Filiot, R. Gentilini, and J.-F. Raskin. The complexity of rational synthesis. In *Proc. 43th Int. Colloq. on Automata, Languages, and Programming*, volume 55 of *LIPIcs*, pages 121:1–121:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.
18. S. Dziembowski, M. Jurdzinski, and I. Walukiewicz. How much memory is needed to win infinite games. In *Proc. 12th ACM/IEEE Symp. on Logic in Computer Science*, pages 99–110, 1997.
19. E.A. Emerson and C. Jutla. Tree automata, μ-calculus and determinacy. In *Proc. 32nd IEEE Symp. on Foundations of Computer Science*, pages 368–377, 1991.
20. D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In *Proc. 16th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2010.
21. D. Harel, G. Katz, A. Marron, and G. Weiss. Non-intrusive repair of reactive programs. In *ICECCS*, pages 3–12, 2012.
22. T.A. Henzinger. From Boolean to quantitative notions of correctness. In *Proc. 37th ACM Symp. on Principles of Programming Languages*, pages 157–158, 2010.
23. 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.
24. B. Jobstmann, A. Griesmayer, and R. Bloem. Program repair as a game. In *Proc. 17th Int. Conf. on Computer Aided Verification*, volume 3576 of *Lecture Notes in Computer Science*, pages 226–238, 2005.
25. N. Klarlund. Progress measures, immediate determinacy, and a subset construction for tree automata. *Annals of Pure and Applied Logic*, 69(2):243–268, 1994.
26. O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. *Annals of Mathematics and Artificial Intelligence*, 78(1):3–20, 2016.
27. 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.
28. O. Kupferman and N. Shenwald. Games with weighted multiple objectives. In *22nd Int. Symp. on Automated Technology for Verification and Analysis*, Lecture Notes in Computer Science. Springer, 2024.
29. 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.
30. Y. Lustig and M.Y. Vardi. Synthesis from component libraries. *Software Tools for Technology Transfer*, 15(5-6):603–618, 2013.

31. R. McNaughton. Infinite games played on finite graphs. *Annals of Pure and Applied Logic*, 65:149–184, 1993.
32. J.F. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*, 1950.
33. J. Neumann, A. Szepietowski, and I. Walukiewicz. Complexity of weak acceptance conditions in tree automata. *Inf. Process. Lett.*, 84(4):181–187, 2002.
34. N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.
35. S. Paul and S-E. Simon. Nash equilibrium in generalised muller games. In *Proc. 29th Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 4 of *LIPIcs*, pages 335–346. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2009.
36. 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.
37. R.E. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal of Computing*, 1(2):146–160, 1972.
38. M. Ummels. The complexity of Nash equilibria in infinite multiplayer games. In *Proc. 11th Int. Conf. on Foundations of Software Science and Computation Structures*, pages 20–34, 2008.
39. M. Ummels and D. Wojtczak. The complexity of nash equilibria in limit-average games. In *Proc. 22nd Int. Conf. on Concurrency Theory*, volume 6901 of *Lecture Notes in Computer Science*, pages 482–496. Springer, 2011.
40. 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.
41. J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1953.