Games with Trading of Control

Orna Kupferman

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

Noam Shenwald

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

Abstract

The interaction among components in a system is traditionally modeled by a game. In the turned-based setting, the players in the game jointly move a token along the game graph, with each player deciding where to move the token in vertices she controls. The objectives of the players are modeled by ω-regular winning conditions, and players whose objectives are satisfied get rewards. Thus, the game is non-zero-sum, and we are interested in its stable outcomes. In particular, in the rational-synthesis problem, we seek a strategy for the system player that guarantees the satisfaction of the system's objective in all rational environments. In this paper, we study an extension of the traditional setting by *trading of control*. In our game, the players may pay each other in exchange for directing the token also in vertices they do not control. The utility of each player then combines the reward for the satisfaction of her objective and the profit from the trading. The setting combines challenges from ω-regular graph games with challenges in pricing, bidding, and auctions in classical game theory. We study the theoretical properties of *parity trading games*: best-response dynamics, existence and search for Nash equilibria, and measures for equilibrium inefficiency. We also study the rational-synthesis problem and analyze its tight complexity in various settings.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Logic and verification

Keywords and phrases Parity Games, Rational Synthesis, Game Theory, Auctions

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2023.19

Related Version *Full Version*: https://www.cs.huji.ac.il/~ornak/publications/concur23.pdf

Funding Research is funded by the Israel Science Foundation, Grant 2357/19, and the European Research Council, Advanced Grant ADVANSYNT.

1 Introduction

Synthesis is the automated construction of a system from its specification. A useful way to approach synthesis of *reactive* systems is to consider the situation as a *game* between the system and its environment. Together, they generate a computation, and the system wins if the computation satisfies the specification. Thus, synthesis is reduced to generation of a winning strategy for the system in the game – a strategy that ensures that the system wins against all environments [1, 35].

Nowadays systems have rich structures. More and more systems lack a centralized authority and involve selfish users, giving rise to an extensive study of *multi-agent systems* [2] in which the agents have their own objectives, and thus correspond to *non-zero-sum games* [33]: the outcome of the game may satisfy the objectives of a subset of the agents.

The rich settings in which synthesis is applied have led to more involved definitions of the problem. First, in *rational synthesis* [26, 28, 24, 25, 30], the goal is to construct a system that satisfies the specification in all rational environments, namely environments that are composed of components that have their own objectives and act to achieve their objectives. The system can capitalize on the rationality of the environment, leading to synthesis of specifications that cannot be synthesized in hostile environments. Then, in

34th International Conference on Concurrency Theory (CONCUR 2023).
Editors: Guillermo A. Pérez and Jean-François Raskin; Article No. 19; pp. 19:1–19:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

quantitative synthesis, the satisfaction value of a specification in a computation need not be Boolean. Thus, beyond correctness, specifications may describe *quality*, enabling the specifier to prioritize different satisfaction scenarios. For example, the value of a computation may be a value in $\mathbb{N}$, reflecting costs and rewards to events along the computation. A synthesis algorithm aims to construct systems that satisfy their objectives in the highest possible value [3, 5, 6, 18, 20]. *Quantitative rational synthesis* then combines the two extensions, with systems composed of rational components having quantitative objectives [26, 28, 6, 19].

Viewing synthesis as a game has led to a fruitful exchange of ideas between *formal methods* and *game theory* [17, 27]. The extensions to rational and quantitative synthesis make the connection between the two communities stronger. Indeed, rationality is a prominent notion in game theory, and most studies in game theory involve quantitative utilities for the players. Classical game theory concerns games for economy-driven applications like resource allocation, pricing, bidding, auctions, and more [37, 33]. Many more useful ideas in classical game theory are waiting to be explored and used in the context of synthesis [23]. In this paper, we introduce and study a framework for extending synthesis with *trading of control*. For example, in a communication network in which each company controls a subset of the routers, companies may pay each other in exchange for committing on some routing decisions, and in a system consisting of a server and clients, clients may pay the server for allocating resources in some beneficial way. The decisions of the players in such settings depend on both their behavioral objectives and their desire to maximize the profit from the trade. When a media company decides, for example, how many and which advertisements it broadcasts, its decisions depend not only on the expected revenue but also on its need to limit the volume (and hopefully also content) of commercial content it broadcasts [16, 31]. More examples include *shields* in synthesis, which can alter commands issued by a controller, aiming to guarantee maximal performance with minimal interference [7, 9].

Our framework considers multi-agent systems modeled by a game played on a graph. Since we care about infinite on-going behaviors of the system, we consider infinite paths in the graph, which correspond to computations of the system. We study settings in which each of the players has control in different parts of the system. Formally, if there are n players, then there is a partition $V_1, \ldots, V_n$ of the set of vertices in the game graph among the players, with Player i controlling the vertices in V_i. The game is *turn-based*: starting from an initial vertex, the players jointly move a token along the game graph, with each player deciding where to move the token in vertices she controls. A *strategy* for Player i directs her how to move a token that reaches a vertex in V_i. A *profile* is a vector of strategies, one for each player, and the *outcome* of a profile is the path generated when the players follow their strategies in the profile. The objectives of the players refer to the generated path. In classical *parity games* (PGs, for short), they are given by *parity* winning conditions over the set of vertices of the graph. Thus, each player has a coloring that assigns numbers to vertices in the graph, and her objective is that the minimal color the path visits infinitely often is even. While satisfaction of the parity winning condition is Boolean, the players get quantitative rewards for satisfying their objectives.

In *parity trading games* (PTG, for short), a strategy for Player i is composed of two strategies: a *buying strategy*, which specifies, for each edge $\langle v, u\rangle$ in the game, how much Player i offers to pay the player that controls v in exchange for this player selling $\langle v, u\rangle$; that is, for always choosing u as v's successor; and a *selling strategy*, which specifies, for each vertex $v \in V_i$, which edge from v is sold, as a function of the offers that Player i receives from the other players. Note that Player i need not sell the edge that gets the highest offer. Indeed, her choice also depends on her objective. Also note that selling strategies are similar to memoryless strategies in PGs, in the sense that a sold edge is going to be traversed in all the visits of the token to its source vertex, regardless of the history of the path. Recall that we consider parity winning conditions, which admits memoryless winning strategies. Accordingly, if a player can force the satisfaction of her parity objective in a PG she can also force the satisfaction of her parity objective in the corresponding PTG.

A profile of strategies in a PTG induces a set of sold edges, one from each vertex. Hence, as in PGs, the outcome of each profile is a path in the game. The utility of Player i in the game is the sum of two factors: a *satisfaction profit*, which, as in PGs, is a reward that Player i receives if the outcome satisfies her objective, and a *trading profit*, which is the sum of payments she receives from the other players, minus the sum of payments she gives others, where payments are made only for sold edges.

Related work studies synthesis of systems that combine behavioral and monetary objectives. One direction of work considers systems with *budgets*. The budget can be used for tasks such as sensing of input signals, purchase of library components [22, 15, 4], and, in the context of control – shielding a controller that interacts with a plant [7, 9]. Even closer is work in which the players can use the budget in order to negotiate control. The most relevant work here is on *bidding games* [12]: graph games in which in each turn an auction is held in order to determine which player gets control. That is, whenever the token is on a vertex v, the players submit bids, the player with the highest bid wins, she decides to which successor of v to move the token, and the budgets of the players are updated according to the bids. Variants of the game refer to its duration, the type of objectives, the way the budgets are updated, and more [13, 14, 11]. Trading games are very different from bidding games: in trading games, negotiation about buying and selling of control takes place before the game starts, and no auctions are held during the game. Also, the games include an initial partition of control, as is the natural setting in multi-agent systems. Moreover, control in trading games is not sold to the highest offer. Rather, selling strategies may depend in the objective of the seller. Finally, the games are non-zero-sum, and are studied for arbitrary number of players.

Another direction of related work considers systems with dynamic change of control that do not involve monetary objectives, such as *pawn games* [10]: zero-sum turn-based games in which the vertices are statically partitioned between a set of *pawns*, the pawns are dynamically partitioned between the players, and the player that chooses the successor for a vertex v at a given turn is the player that controls the pawn to which v belongs. At the end of each turn, the partition of the pawns among the players is updated according to a predetermined mechanism.

Since a PTG is non-zero-sum, interesting questions about it concern *stable outcomes*, in particular *Nash equilibria* (NE) [32]. A profile is an NE if no player has a beneficial deviation; thus, no player can increase her utility by changing her strategy in the profile. Note that in PTGs, a change of a strategy amounts to a change in the buying or selling strategies, or in both of them.

We first study *best response* in PTGs – the problem of finding the most beneficial deviation for a player in a given profile. We show that the problem can be reduced to the problem of finding shortest paths in weighted graphs. Essentially, the weights in the graph are induced by the maximal profit that a player can make from selling edges from vertices she owns and the minimal profit she may lose in order to buy edges from vertices she does not own. We conclude that the problem can be solved in polynomial time. We also study *best response dynamics* – a process in which, as long as the profile is not an NE, some player is chosen to perform her best response. We show that trading makes the setting less stable, in the sense that best response dynamics need not converge to an NE, even when convergence is guaranteed in the underlying PG. On the positive side, as is the case in PGs, every PTG has an NE.

We continue and study rational synthesis in PTGs. Two approaches to rational synthesis have been studied. In *cooperative* rational synthesis (CRS) [26], the desired output is an NE profile whose outcome satisfies the objective of the system. In *non-cooperative* rational synthesis (NRS) [28], we seek a strategy for the system such that its objective is satisfied in the outcome of all NE profiles that include this strategy. In settings with quantitative utilities, in particular PTGs, the input to the CRS and NRS problems includes a threshold $t \geq 0$, and we replace the requirement for the system to satisfy her objective by the requirement that her utility is at least t. The two approaches have to do with the technical ability to communicate strategies to the environment players, say due to different architectures, as well as with the willingness of the environment players to follow a suggested strategy. As shown in [6], the two approaches are related to the two stability-inefficiency measures of *price of stability* (PoS) [8] and *price of anarchy* (PoA) [29, 34], and we study these measures in the context of PTG.

Problem	Finding an NE	Cooperative Rational Synthesis	Non-cooperative Rational Synthesis
Parity Games	UP ∩ co-UP fixed n NP-complete unfixed n [37], [Th. 5]	UP ∩ co-UP fixed n NP-complete unfixed n [22], [37]	PSPACE, NP-hard, co-NP-hard fixed n EXPTIME, PSPACE-hard unfixed n [22]
Parity Trading Games		NP-complete [Th. 10]	NP-complete $n = 2$ Σ_2^{P}-complete $n \geq 3$ [Th. 12], [Th. 13]
Büchi Games	PTIME [37], [Th. 5]	PTIME [37]	PTIME fixed n PSPACE-complete unfixed n [22]
Büchi Trading Games		NP-complete [Th. 10]	NP-complete $n = 2$ Σ_2^{P}-complete $n \geq 3$ or unfixed n [Th. 12], [Th. 13]

Figure 1 Complexity of different problems on n-player PGs, PTGs, BGs, and BTGs.

In PGs, the tight complexity of rational synthesis is still open, and depends on whether the number of players is fixed. We show that in PTGs, CRS is NP-complete, and the complexity of NRS depends on the number of players: it is NP-complete for two players and is Σ_2^{P}-complete for three or more (in particular, unfixed number of) players. Our upper bounds are based on reductions to a sequence of shortest-path algorithms in weighted graphs. They hold also for an unfixed number of players, making rational synthesis with an unfixed number of players easier in PTGs than in PGs. Intuitively, it follows from the fact that deviations in the selling or buying strategies of single players in PTGs induce a change in the outcome only if they are matched by the buying and selling strategies, respectively, of players that do not deviate. Our lower bounds involve reductions from SAT and QBF_2, where trade is used to incentive a satisfying assignment, when exists, and to ensure the consistency of suggested assignments. When the number of players in the environment is bigger than 2, we can use trade among the environment players in order to simulate universal quantification, which explains the transition form NP to Σ_2^{P}.

Our complexity results on ω-regular trading games and their comparison to standard ω-regular non-zero-sum games are summarized in the table in Figure 1. Due to the lack of space, examples and some proofs are omitted or given partially, and can be found at the full version.

2 Preliminaries

For $n \geq 1$, let $[n] = \{1, \ldots, n\}$. An *n-player game graph* is a tuple $G = \langle \{V_i\}_{i \in [n]}, v_0, E \rangle$, where $\{V_i\}_{i \in [n]}$ are disjoint sets of vertices, each owned by a different player, and we let $V = \bigcup_{i \in [n]} V_i$. Then, $v_0 \in V_1$ is an initial vertex, which we assume to be owned by Player 1, 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 $\langle v, u \rangle \in E$. The *size* $|G|$ of G is $|E|$, namely the number of edges in it.

For every vertex $v \in V$, we denote by $\mathsf{succ}(v)$ the set of successors of v in G. That is, $\mathsf{succ}(v) = \{u \in V : \langle v, u\rangle \in E\}$. Also, for every $v \in V$, we denote by E_v the set of edges from v. That is, $E_v = \{\langle v, u\rangle : u \in \mathsf{succ}(v)\}$. Then, for every $i \in [n]$, we denote by E_i the set of edges whose source vertex is owned by Player i. That is, $E_i = \bigcup_{v \in V_i} E_v$.

In the beginning of 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* $\rho = v_0, v_1, \ldots$ in G, namely an infinite path that starts in v_0 and respects E: for all $i \geq 0$, we have that $(v_i, v_{i+1}) \in E$.

For a play $\rho = v_0, v_1, \ldots$, we denote by $\mathit{inf}(\rho)$ the set of vertices visited infinitely often along ρ. That is, $\mathit{inf}(\rho) = \{v \in V : \text{there are infinitely many } i \geq 0 \text{ such that } v_i = v\}$. A *parity* objective is given by a coloring function $\alpha : V \to \{0, \ldots, k\}$, for some $k \geq 0$, and requires the minimal color visited infinitely often along ρ to be even. Formally, a play ρ satisfies α iff $\min\{\alpha(v) : v \in \mathit{inf}(\rho)\}$ is even. A *Büchi* objective is a special case of parity. For simplicity, we describe a Büchi objective by a set of vertices $\alpha \subseteq V$. The condition requires that some vertex in α is visited infinitely often along ρ, thus $\mathit{inf}(\rho) \cap \alpha \neq \emptyset$.

A *parity game* (PG, for short) is a tuple $\mathcal{G} = \langle G, \{\alpha_i\}_{i\in[n]}, \{R_i\}_{i\in[n]}\rangle$, where G is a n-player game graph, and for every $i \in [n]$, we have that $\alpha_i : V \to \{0, \ldots, k_i\}$ is a parity objective for Player i. Intuitively, for every $i \in [n]$, Player i aims for a play ρ that satisfies her objective α_i, and $R_i \in \mathbb{N}$ is a reward that Player i gets when α_i is satisfied. Büchi games (BG, for short) are defined similarly, with Büchi objectives. We assume that at least one condition is satisfiable.

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 $(v, f_i(\rho \cdot v)) \in E$. A strategy f_i for Player i is *memoryless* if it only depends on the current vertex. That is, if for every two histories $h, h' \in V^*$ and vertex $v \in V_i$, we have that $f_i(h \cdot v) = f_i(h' \cdot v)$. Note that a memoryless strategy can be viewed as a function $f_i : V_i \to V$.

A *profile* is a tuple $\pi = \langle f_1, \ldots, f_n\rangle$ of strategies, one for each player. The *outcome* of a profile $\pi = \langle f_1, \ldots, f_n\rangle$ is the play obtained when the players follow their strategies. Formally, $\mathsf{Outcome}(\pi) = v_0, v_1, \ldots$ is such that for all $j \geq 0$, we have that $v_{j+1} = f_i(v_0, v_1, \ldots, v_j)$, where $i \in [n]$ is such that $v_j \in V_i$. For every profile π and $i \in [n]$, we say that Player i *wins in* π if $\mathsf{Outcome}(\pi) \models \alpha_i$. Otherwise, Player i *loses in* π. We denote by $\mathsf{Win}(\pi)$ the set of players that win in π. Then, the *satisfaction profit of Player i in* π, denoted $\mathsf{sprofit}_i(\pi)$, is R_i if $i \in \mathsf{Win}(\pi)$, and is 0 otherwise.

As the objectives of the players may overlap, the game is not zero-sum and thus we are interested in *stable* profiles in the game. A profile $\pi = \langle f_1, \ldots, f_n\rangle$ is a *Nash Equilibrium* (NE, for short) [32] if, intuitively, no player can benefit (that is, increase her profit) from unilaterally changing her strategy. Formally, for $i \in [n]$ and some strategy f'_i for Player i, let $\pi[i \leftarrow f'_i] = \langle f_1, \ldots, f_{i-1}, f'_i, f_{i+1}, \ldots, f_n\rangle$ be the profile in which Player i *deviates* to the strategy f'_i. We say that π is an NE if for every $i \in [n]$, we have that $\mathsf{sprofit}_i(\pi) \geq \mathsf{sprofit}_i(\pi[i \leftarrow f'_i])$, for every strategy f'_i for Player i. That is, no player can unilaterally increase her profit.

In *rational synthesis*, we consider a game between a system, modeled by Player 1, and an environment composed of several components, modeled by Players $2 \ldots n$. Then, we seek a strategy for Player 1 with which she wins, assuming rationality of the other players. Note that the system may also be composed of several components, each with its own objective. It is not hard to see, however, that they can be merged to a single player whose objective is the conjunction of the underlying components.

We say that a profile $\pi = \langle f_1, \dots, f_n \rangle$ is a 1-*fixed NE*, if no player $i \in [n] \setminus \{1\}$ has a beneficial deviation. We formalize the intuition behind rational synthesis in two ways, as follows. Consider an n-player game $\mathcal{G} = \langle G, \{\alpha_i\}_{i\in[n]}, \{R_i\}_{i\in[n]} \rangle$, and a threshold $t \geq 0$. The problem of *cooperative rational synthesis* (CRS) is to return a 1-fixed NE π such that $\mathsf{sprofit}_1(\pi) \geq t$. The problem of *non-cooperative rational synthesis* (NRS) is to return a strategy f_1 for Player 1 such that for every 1-fixed NE π that extends f_1, we have that $\mathsf{sprofit}_1(\pi) \geq t$.

As in traditional synthesis, one can also define the corresponding decision problems, of *rational realizability*, where we only need to decide whether the desired strategies exist. In order to avoid additional notations, we refer to CRS and NRS also as decision problems.

3 Parity Trading Games

Parity trading games (PTG, for short, or BTG, when the objectives of the players are Büchi objectives) are similar to parity games, except that now, the movement of the token along the game graph depends on trade among the players, who pay each other in exchange for certain behaviors. Thus, instead of strategies that direct them how to move the token, now the players have strategies that direct the trade.

Consider a PTG $\mathcal{G} = \langle G, \{\alpha_i\}_{i\in[n]}, \{R_i\}_{i\in[n]} \rangle$, defined on top of a game graph $G = \langle \{V_i\}_{i\in[n]}, v_0, E \rangle$. A *buying strategy* for Player i is a function $b_i : E \to \mathbb{N}$ that maps each edge $e = \langle v, u \rangle \in E$ to the price that Player i is willing to pay to the owner of v in exchange for selling e; that is, for always choosing u as v's successor when the token is in v. For edges $e \in E_i$, we require $b_i(e)$ to be 0.

Consider a vector $\beta = \langle b_1, \dots, b_n \rangle$ of buying strategies, one for each player. The vector β determines, for an edge $e \in E$, the collective price that the players are willing to pay for e. Accordingly, we sometime refer to β as a *price list*, namely a function in $\mathbb{N}^E$, where for every $e \in E$, we have that $\beta(e) = \sum_{i\in[n]} b_i(e)$.

A *selling strategy* for Player i determines which edges Player i sells. The strategy is a collection of policies, which determines for each $v \in V_i$, which edge from v to sell, given prices offered for the edges in E_v. Formally, a *selling policy for* $v \in V_i$ is a function $s_v : \mathbb{N}^{E_v} \to E_v$ that maps each price list for the edges in E_v to an edge in E_v. Note that the mapping is arbitrary, thus a player need not sell the edge that gets the highest price. We refer to the selling strategy for Player i, thus the collection $\{s_v : v \in V_i\}$ of selling policies for her vertices, as a function $s_i : \mathbb{N}^E \to 2^{E_i}$ that maps price lists to the set of edges that Player i chooses to sell. Note also that selling strategies in PTGs are similar to memoryless strategies in PGs, in the sense that the choice of the edge that is sold from v is independent of the history of the game.

A *profile* is a tuple $\pi = \langle (b_1, s_1), \dots, (b_n, s_n) \rangle$ of pairs of buying and selling strategies, one for each player. We sometime refer to the pair of buying and selling strategies for Player i as a single strategy, and use the notation $f_i = (b_i, s_i)$. We also use β_π to denote the price list induced by the buying strategies in π. We say that an edge $e \in E_i$ is *sold* in π iff $e \in s_i(\beta_\pi)$. We denote by $\mathsf{S}(\pi)$ the set of edges sold in π. Recall that for every $v \in V$, there exists exactly one edge $e \in E_v$ such that $e \in \mathsf{S}(\pi)$. The *outcome* of a profile π, denoted $\mathsf{Outcome}(\pi)$, is then the path $v_0, v_1, \dots$, where for all $j \geq 0$, we have that $(v_j, v_{j+1}) \in \mathsf{S}(\pi)$.

As in PGs, the satisfaction profit of Player i in π, denoted $\mathsf{sprofit}_i(\pi)$, is R_i if α_i is satisfied in $\mathsf{Outcome}(\pi)$, and is 0 otherwise. In PTGs, however, we consider also the trading profits of the players: For every player $i \in [n]$, the *gain* of Player i in π, denoted $\mathsf{gain}_i(\pi)$, is the sum of payments she receives from other players, and the *loss* of Player i, denoted

$\mathsf{loss}_i(\pi)$, is the sum of payments she pays others. That is, $\mathsf{gain}_i(\pi) = \sum_{e \in \mathsf{S}(\pi) \cap E_i} \beta_\pi(e)$, and $\mathsf{loss}_i(\pi) = \sum_{e \in \mathsf{S}(\pi)} b_i(e)$. Then, the *trading profit* of Player i in π, denoted $\mathsf{tprofit}_i(\pi)$, is her gain minus her loss in π. That is, $\mathsf{tprofit}_i(\pi) = \mathsf{gain}_i(\pi) - \mathsf{loss}_i(\pi)$. Note that while all the edges in $\mathsf{Outcome}(\pi)$ are in $\mathsf{S}(\pi)$, not all edges in $\mathsf{S}(\pi)$ are traversed during the play. Still, payments depend only on $\mathsf{S}(\pi)$, regardless of whether the edges are traversed. Finally, the *utility* of Player i in π, denoted $\mathsf{util}_i(\pi)$, is the sum of her satisfaction and trading profits in π. That is, $\mathsf{util}_i(\pi) = \mathsf{sprofit}_i(\pi) + \mathsf{tprofit}_i(\pi)$. The definitions of beneficial deviations, NEs, and 1-fixed NEs are then defined as in the case of PG.

Note that the definition of a selling strategy s_i as a function from $\mathbb{N}^E$ hides the fact that the selling policy for each vertex $v \in V_i$ depends only on the price list for the edges in E_v. Note also that as there are infinitely many price lists, an enumerative presentation of selling strategies is infinite. As we detail in the full version, we assume that selling strategies are given symbolically. For example, a selling strategy for a vertex v with successors $\{u_1, u_2, u_3\}$, may be "if the price offered for u_2 is at least p, then sell (v, u_2); otherwise, sell (v, u_1)". Specifically, a strategy for Player i is given by a set of pairs of the form $\langle b, T \rangle$, where b is a predicate on $\mathbb{N}^E$ and $T \subseteq E_i$ is the set of edges that Player i sells when then price list satisfies b. The predicates are disjoint, and can be computed in polynomial time. In the full version we also argue that every profile π of strategies can be simplified so that the set of winners and the utilities for the players are preserved, and all prices are of polynomial size. As we argue in the sequel, restricting attention to simple profiles and to strategies that can be represented symbolically does not lose generality, in the sense that whenever we search for a profile of strategies and a desired profile exists, then there is also a profile that consists of strategies that can be represented symbolically.

Describing a profile $\pi = \langle (b_1, s_1), \ldots, (b_n, s_n) \rangle$, we sometimes use a symbolic description, as follows. For players $i, j \in [n]$, an edge $e \in E_j$, and a price $p \in \mathbb{N}$, we say that Player i *offers to buy e for price p* if $b_i(e) = p$, and that Player i *pays p for e* if, in addition, $e \in s_j(\beta_\pi)$. For a vertex $v \in V_i$, and an edge $e = \langle v, u \rangle \in E_v$, we say that Player i *moves from v to u*, if $e \in s_i(\beta_\pi)$, thus Player i sells e in β_π. Then, we say that Player i *always moves from v to u*, if Player i always sells e, thus $e \in s_i(\beta)$ for every price list β. Describing a deviation from π to a profile $\pi' = \langle (b'_1, s'_1), \ldots, (b'_n, s'_n) \rangle$, we sometimes use a symbolic description, as follows. For a player $i \in [n]$ and an edge $e \in E$, we say that Player i *cancels the purchase of* e if $b_i(e) > 0$ and $b'_i(e) = 0$. For an edge $e \in E_i$, we say that Player i *cancels the sale of e* if $e \in s_i(\beta_\pi)$ and $e \notin s_i(\beta_{\pi'})$.

4 Stability in Parity Trading Games

In this section we study the stability of PTGs. We start with the best-response problem, which searches for deviations that are most beneficial for the players, and show that the problem can be solved in polynomial time. On the negative side, a best-response dynamics in PTGs, where players repeatedly perform their most beneficial deviations, need not converge. We then study the existence of NEs in PTGs, show that every PTG has an NE, and relate the stability in a PTG and its underlying PG. Finally, we study the inefficiency that may be caused by instability, and show that the price of stability and price of anarchy in PTGs are unbounded and infinite, respectively.

Throughout this section, we consider an n-player game $\mathcal{G} = \langle G, \{\alpha_i\}_{i \in [n]}, \{R_i\}_{i \in [n]} \rangle$, defined on top of a game graph $G = \langle \{V_i\}_{i \in [n]}, v_0, E \rangle$. We use $\mathcal{G}^P$ and $\mathcal{G}^T$ to denote $\mathcal{G}$ when viewed as a PG and PTG, respectively.

4.1 Best response

The input to the *best response* (BR, for short) problem is a game $\mathcal{G}$, a profile π, and $i \in [n]$. The goal is to find a strategy f'_i for Player i such that $\mathsf{util}_i(\pi[i \leftarrow f'_i])$ is maximal. We describe an algorithm that solves the BR problem in polynomial time. The key idea behind our algorithm is as follows. Consider a profile $\pi = \langle (b_1, s_1), \ldots, (b_n, s_n) \rangle$. Recall that the utility of Player i in π is the sum of her satisfaction and trading profits in π. If Player i ignores her objective and only tries to maximize her trading profit, then her strategy is straightforward: she buys no edge, and in each vertex $v \in V_i$, she sells an edge with the maximal price in β_π. If there is a strategy f^*_i as above such that the outcome of $\pi[i \leftarrow f^*_i]$ satisfies α_i, then clearly f^*_i is a best response for Player i, and we are done. Otherwise, the algorithm searches for a minimal reduction in the trading profit with which Player i can induce an outcome that satisfies α_i. For this, the algorithm labels each edge $e = \langle v, u \rangle$ in G by the cost of ensuring that e is sold. If Player i owns e, then this cost is the difference between $\beta_\pi(e)$ and $\max\{\beta_\pi(e') : e' \in E_v\}$. If Player i does not own e, thus $v \in V_j$, for some player $j \neq i$, then this cost is the minimal price that Player i has to offer for e in order to change β_π to a price list β for which $s_j(\beta) = e$. Once the graph G is labeled by costs as above, the desired strategy is induced by the path with the minimal cost that satisfies α_i. Finally, if the minimal cost of satisfying α_i is higher than her reward R_i, then the best response for Player i is to give up the satisfaction of α_i and follow the strategy f^*_i, in which the maximal trading profit is attained.

We now describe the algorithm in detail. We first label the edges from every vertex $v \in V$ by costs in $\mathbb{N}$. For every vertex $v \in V_i$, we denote by $\mathsf{potential}(\pi, v)$ the maximal price that Player i can get from selling an edge from v. That is, $\mathsf{potential}(\pi, v) = \max\{\beta_\pi(e) : e \in E_v\}$. For every vertex $v \in V_i$ and edge $e \in E_v$, we define $\mathsf{cost}(\pi, e)$ as the cost for Player i of selling e rather then an edge that attains $\mathsf{potential}(\pi, v)$. That is, $\mathsf{cost}(\pi, e) = \mathsf{potential}(\pi, v) - \beta_\pi(e)$.

We continue to vertices $v \notin V_i$. For $j \in [n] \setminus \{i\}$ and an edge $e \in E_j$, we define $\mathsf{cost}(\pi, e)$ as the minimal price that Player i needs to pay to Player j in order for her to sell e. Formally, let B^e_i be the set of buying strategies for Player i that cause Player j to sell e. That is, $B^e_i = \{b'_i : E \to \mathbb{N} : e \in s_j(\beta_\pi[i \leftarrow b'_i])\}$. When Player i uses a strategy $b'_i \in B^e_i$ as her buying strategy, Player j sells e, and Player i pays the price $b'_i(e)$. Hence, the minimal price that Player i needs to pay in order for Player j to sell e is $\mathsf{cost}(\pi, e) = \min\{b'_i(e) : b'_i \in B^e_i\}$. Note that B^e_i may be empty, in which case $\mathsf{cost}(\pi, e) = \infty$.

We define $\mathsf{best}(\pi) \subseteq E$ as the set of edges that minimize the cost of Player i. Formally, $\mathsf{best}(\pi) = \bigcup_{v \in V} \mathsf{best}(\pi, v)$, where for $v \in V_i$, we have that $\mathsf{best}(\pi, v) \subseteq E_v$ is the set of edges from v with which $\mathsf{potential}(\pi, v)$ is attained, thus $\mathsf{best}(\pi, v) = \{e \in E_v : \beta_\pi(e) = \mathsf{potential}(\pi, v)\}$; and for $v \in V_j$, for $j \neq i$, we have that $\mathsf{best}(\pi, v)$ is the set of edges from v that Player i can make Player j sell without paying for e, thus $\mathsf{best}(\pi, v) = \{e \in E_v : \mathsf{cost}(\pi, e) = 0\}$. Note that for every vertex $v \in V$, the set $\mathsf{best}(\pi, v)$ is not empty.

We say that a path ρ in G is *feasible* if $\mathsf{cost}(\pi, e) < \infty$ for every edge e in ρ. In Lemma 1 below, we argue that for every feasible path ρ, Player i can change her strategy in π so that the outcome of the new profile is ρ. We also calculate the cost required for Player i to do so.

▶ **Lemma 1.** *Let ρ be a feasible path in $\mathcal{G}$. Then, there exists a strategy f^ρ_i for Player i such that $\mathsf{Outcome}(\pi[i \leftarrow f^\rho_i]) = \rho$, and $\mathsf{tprofit}_i(\pi[i \leftarrow f^\rho_i]) = \sum_{v \in V_i} \mathsf{potential}(\pi, v) - \sum_{e \in \rho} \mathsf{cost}(\pi, e)$. Also, $\mathsf{tprofit}_i(\pi[i \leftarrow f^\rho_i])$ is the maximal trading profit for Player i when she changes her strategy in π to a strategy that causes the outcome to be ρ.*

For a path ρ in G, let f^ρ_i be a strategy for Player i such that the outcome of $\pi[i \leftarrow f^\rho_i]$ is ρ. Note that f^ρ_i can be described symbolically.

Our algorithms for finding beneficial deviations are based on a search for short *lassos* in weighted variants of the graph G. A lasso is a path of the form $\rho_1 \cdot \rho_2^\omega$, for finite paths $\rho_1 \in V^*$ and $\rho_2 \in V^+$. When G is weighted, the length of the lasso is defined as the sum of the weights in the path $\rho_1 \cdot \rho_2$.

▶ **Theorem 2.** *The BR problem in PTGs can be solved in polynomial time.*

Proof. Given an n-player PTG $\mathcal{G}$, a profile π, and $i \in [n]$, the algorithm for finding a BR for Player i proceeds as follows.

1. Let $G^{\mathsf{best}(\pi)} = \langle V, \mathsf{best}(\pi)\rangle$ be the restriction of G to edges in $\mathsf{best}(\pi)$.
2. If there is a path ρ in $G^{\mathsf{best}(\pi)}$ that satisfies α_i, then return f_i^ρ. Otherwise, let f_i^* be a strategy for Player i that induces some lasso in $G^{\mathsf{best}(\pi)}$.
3. Let $G' = \langle V, E, w\rangle$ be the weighted extension of G, where $w : E \to \mathbb{N}$ is such that for every edge $e \in E$, we have that $w(e) = \mathsf{cost}(\pi, e)$.
4. Let ρ be a shortest (with respect to the weights in w) lasso that satisfies α_i.
5. If $w(\rho) \geq R_i$, then return f_i^*, else return f_i^ρ. ◀

Recall that a *best response dynamic* (BRD) is an iterative process in which as long as the profile is not an NE, some player is chosen to perform a best response. In Theorem 3 below, we demonstrate that a BRD in a PTG (in fact, a BTG) need not converge, even in settings in which every BRD in the corresponding PG does converge.

▶ **Theorem 3.** *There is a game $\mathcal{G}$ such that every BRD in the PG $\mathcal{G}^P$ converges to an NE, yet a BRD in $\mathcal{G}^T$ need not converge.*

Proof. Consider the 2-player Büchi game $\mathcal{G} = \langle G, \{\alpha_1, \alpha_2\}, \{1, 3\}\rangle$, where G is described in Figure 2, $\alpha_1 = \{a, c\}$, and $\alpha_2 = \{b, d\}$.

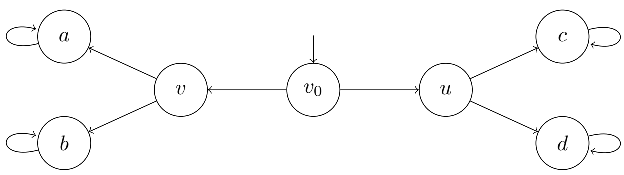

Figure 2 The game graph G. All the vertices are owned by Player 1.

All the vertices in G are owned by Player 1, and the vertices in α_1 are reachable sinks. Hence, once Player 1 is chosen to deviate in $\mathcal{G}^P$, an NE is reached.

In the full version we describe a BRD in $\mathcal{G}^T$ that does not converge. ◀

4.2 Nash equilibria

We continue and show that while a BRD in $\mathcal{G}^T$ need not converge even when every BRD in $\mathcal{G}^P$ does, we can still use NEs in $\mathcal{G}^P$ in order to obtain NEs in $\mathcal{G}^T$. Consider a profile $\pi = \langle f_1, \ldots, f_n\rangle$ of memoryless strategies for the players in $\mathcal{G}^P$. We define the *trivial-trading analogue* of π, denoted $tt(\pi)$ as the a profile in $\mathcal{G}^T$ that is obtained from π by replacing each strategy f_i by the pair (b_i, s_i), for an empty buying strategy b_i (that is, $b_i(e) = 0$ for all $e \in E$), and a selling strategy s_i that mimics f_i (that is, for every price list β, we have that $\langle v, u\rangle \in s_i(\beta)$ iff $f_i(v) = u$). Note that all the strategies in $tt(\pi)$ can be described symbolically.

▶ **Lemma 4.** *If π is an NE in $\mathcal{G}^P$ that consists of memoryless strategies, then $tt(\pi)$ is an NE in $\mathcal{G}^T$.*

Lemma 4 enables us to reduce the search for an NE in an n-player PTG $\mathcal{G}^T$ to a search for an NE in the PG $\mathcal{G}^P$:

▶ **Theorem 5.** *Every PTG has an NE, which can be found in UP ∩ co-UP when the number of players is fixed, and in NP when the number of players is not fixed. For BTGs, an NE can be found in polynomial time.*

Recall that for solving the rational-synthesis problem, we are not interested in arbitrary NEs, but in 1-fixed NEs in which the utility of Player 1 is above some threshold. As we shall see now, the situation here is more complicated: searching for solutions for the rational-synthesis problem in a PTG, we cannot reason about the corresponding PG.

▶ **Theorem 6.** *There is a PTG $\mathcal{G}^T$ and $t \geq 1$ such that there is a 1-fixed NE π^T in $\mathcal{G}^T$ with $\mathsf{util}_1(\pi^T) \geq t$, yet for every 1-fixed NE of memoryless strategies π in $\mathcal{G}^P$, we have that $\mathsf{util}_1(tt(\pi)) < t$.*

Proof. Consider the 2-player BTG $\mathcal{G}^T = \langle G, \{\{a\}, \{b\}\}, \{1, 3\}\rangle$, where G appears in Figure 3. Consider a profile π^T in which the strategy for Player 1 moves from v_0 to b if Player 2 offers to buy $\langle v_0, b\rangle$ for price 2, and moves to a otherwise, and the strategy for Player 2 offers to buy $\langle v_0, b\rangle$ for price 2. In the full version, we prove that π^T is a 1-fixed NE with $\mathsf{util}_1(\pi^T) = 2$, whereas for every 1-fixed NE of memoryless strategies π in $\mathcal{G}^P$, we have that $\mathsf{util}_1(tt(\pi)) < 2$. ◀

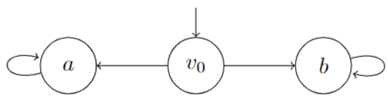

Figure 3 The game graph G. All the vertices are owned by Player 1.

Note that while Theorem 6 considers a 1-fixed NE, and thus corresponds to the setting of CRS, the strategy for Player 1 described there is in fact an NRS solution for the threshold $t = 2$, and the latter cannot be obtained by extending an NRS solution for Player 1 in $\mathcal{G}^P$.

4.3 Equilibrium inefficiency

In this section we study the *price of stability* (PoS) and *price of anarchy* (PoA) measures [33] in PTGs, describing the best-case and worst-case inefficiency of a Nash equilibrium.

Before we define these measures formally, we observe that for every PTG, outcomes that agree on the set of winners also agree in the sum of utilities of the players. Essentially, this follows from the fact that the trading profits for the players sum to 0. Formally, we have the following.

▶ **Lemma 7.** *Let ρ be a path in G, and let $\mathsf{Win}(\rho)$ be the set of players whose objectives are satisfied in ρ. Then, for every profile π with $\mathsf{Outcome}(\pi) = \rho$, we have that the sum of utilities of the players in π is exactly $\sum_{i \in \mathsf{Win}(\rho)} R_i$.*

The *social optimum* in a game $\mathcal{G}$, denoted $\mathsf{SO}(\mathcal{G})$, is the maximal sum of utilities that the players can have in some profile. Thus, $\mathsf{SO}(\mathcal{G})$ is the maximal $\sum_{i \in [n]} \mathsf{util}_i(\pi)$ over all profiles π for $\mathcal{G}$. Since every path ρ in G can be the outcome of some profile, then, by Lemma 7, we have that $\mathsf{SO}(\mathcal{G})$ is the maximal $\sum_{i \in \mathsf{Win}(\rho)} R_i$ over all paths ρ in G.

Let π_B and π_W be NEs with the highest and lowest sum of utilities for the players, respectively. We define $\mathsf{BNE}(\mathcal{G}) = \sum_{i\in[n]} \mathsf{util}_i(\pi_B)$ and $\mathsf{WNE}(\mathcal{G}) = \sum_{i\in[n]} \mathsf{util}_i(\pi_W)$. We then define the price of stability in $\mathcal{G}$ as $\mathsf{PoS}(\mathcal{G}) = \mathsf{SO}(\mathcal{G})/\mathsf{BNE}(\mathcal{G})$, and the price of anarchy in $\mathcal{G}$ as $\mathsf{PoA}(\mathcal{G}) = \mathsf{SO}(\mathcal{G})/\mathsf{WNE}(\mathcal{G})$. Analyzing the prices of stability and anarchy of PTGs, we assume that all rewards in a game $\mathcal{G}$ are positive, thus $R_i > 0$ for all $i \in [n]$. Note that without this assumption, it is easy to define a game $\mathcal{G}$ with $\mathsf{SO}(\mathcal{G}) > 0$ yet $\mathsf{BNE}(\mathcal{G}) = 0$, and hence with $\mathsf{PoS}(\mathcal{G}) = \mathsf{PoA}(\mathcal{G}) = \infty$.

We start with the price of anarchy. It is easy to see that it may be infinite even in simple PTGs in which all rewards are positive:

▶ **Theorem 8.** *There is a 2-player BTG $\mathcal{G}$ with $\mathsf{PoA}(\mathcal{G}) = \infty$.*

Proof. Consider the BTG $\mathcal{G} = \langle G_{PoA}, \{\{a\}, \{a\}\}, \{1, 1\}\rangle$, where the game graph G_{PoA} is described in Figure 4. In the full version we show that $\mathsf{SO}(\mathcal{G}) = 1 + 1 = 2$, whereas $\mathsf{WNE}(\mathcal{G}) = 0$, and so $\mathsf{PoA}(\mathcal{G}) = 2/0 = \infty$. ◀

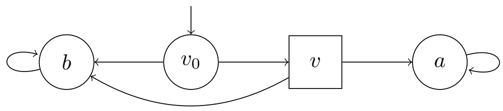

Figure 4 The game graph G_{PoA}. The circles are vertices controlled by Player 1, and the squares are vertices controlled by Player 2.

We continue to the price of stability. It can be shown that every PG has an NE in which all players use memoryless strategies and at least one player satisfies her objective. Essentially, this follows from the fact that either at least one player in the game has a strategy to fulfill her objective from some vertex in all environments (that is, in the zero-sum game played with her objective), or all players do not have such a strategy. In the first case, the outcome of the required NE reaches the winning (in the zero-sum sense) vertex for the player along vertices that are losing (in the zero-sum sense) for the other players. In the second, the outcome traverses a lasso that satisfies the objective of at least one player but consists of vertices that are losing (again, in the zero-sum sense) for all players. By Lemma 4, it then follows that every PTG also has an NE in which at least one player satisfies her objective. Thus, as we assume that all rewards are strictly positive, we conclude that $\mathsf{BNE}(\mathcal{G}) > 0$ for every PTG $\mathcal{G}$. Therefore, we cannot expect $\mathsf{PoS}(\mathcal{G})$ to be ∞, and the strongest result we can prove is that $\mathsf{PoS}(\mathcal{G})$ is unbounded:

▶ **Theorem 9.** *For every $x \in \mathbb{N}$, there exists a two-player BTG $\mathcal{G}$ with $\mathsf{PoS}(\mathcal{G}) = x$.*

Proof. Given x, consider the two-player game graph $G = \langle V_1, V_2, v_1, E\rangle$, where $V_1 = \emptyset$, $V_2 = \{v_1, \ldots, v_{x+2}, u\}$, and $E = \{\langle v_i, v_{i+1}\rangle, \langle v_i, u\rangle : 1 \le i \le x + 1\} \cup \{\langle u, u\rangle, \langle v_{x+2}, v_{x+2}\rangle\}$ (see Figure 5).

Consider the BTG $\mathcal{G} = \langle G, \{\{v_{x+2}\}, \{u\}\}, \{x, 1\}\rangle$. In the full version, we show that $\mathsf{SO}(\mathcal{G}) = x$ whereas $\mathsf{BNE}(\mathcal{G}) = 1$, thus $\mathsf{PoS}(\mathcal{G}) = x$. ◀

5 Cooperative Rational Synthesis in Parity Trading Games

In this section, we study the complexity of the the CRS problem for PTGs and BTGs. Recall that for PGs, the CRS problem can be solved in UP ∩ co-UP when the number of players is fixed, and is in NP when the number of players is not fixed [24]. For BGs, CRS can be solved in polynomial time [36]. We show that trading make the problem harder: CRS in PTGs is NP-complete already for a fixed number of players and for Büchi objectives.

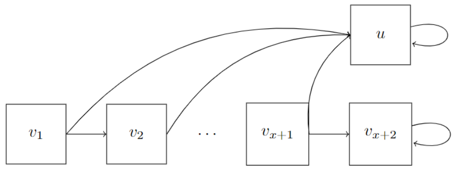

Figure 5 The game graph G. All the vertices are owned by Player 2.

▶ **Theorem 10.** *CRS for PTGs is NP-complete. Hardness in NP holds already for BTGs.*

Proof. We start with membership in NP. Given a threshold $t \geq 0$, an NP algorithm guesses a profile π, checks that $\mathsf{util}_1(\pi) \geq t$, and checks that π is a 1-fixed NE as follows. For every $i \in [n] \setminus \{1\}$, it finds the best response f_i^* for Player i in π, and checks that $\mathsf{util}_i(\pi) \geq \mathsf{util}_i(\pi[i \leftarrow f_i^*])$, thus Player i has no beneficial deviation in π. By Theorem 2, finding the best response for each player in π can be done in polynomial time, hence the check is in polynomial time.

For the lower bound, we describe a reduction from 3-SAT to CRS in BTGs. Let $X = \{x_1, \ldots, x_n\}$, $\overline{X} = \{\overline{x}_1, \ldots, \overline{x}_n\}$, and let φ be a Boolean formula over the variables in X, given in 3CNF. That is, $\varphi = (l_1^1 \vee l_1^2 \vee l_1^3) \wedge \cdots \wedge (l_k^1 \vee l_k^2 \vee l_k^3)$, where for all $1 \leq i \leq k$ and $1 \leq j \leq 3$, we have that $l_i^j \in X \cup \overline{X}$. For every $1 \leq i \leq k$, let $C_i = (l_i^1 \vee l_i^2 \vee l_i^3)$.

Given a formula φ, we construct (see Figure 6) a two-player BG $\mathcal{G} = \langle G_{SAT}, \{\alpha_1, \alpha_2\}, \{R_1, R_2\}\rangle$, where $\alpha_1 = V \setminus \{s\}$, $\alpha_2 = \{s\}$, $R_1 = n + 1$ and $R_2 = 1$, such that φ is satisfiable iff there exists a 1-fixed NE π in $\mathcal{G}$ in which $\mathsf{util}_1(\pi) \geq 1$. The main idea of the reduction is that Player 1 chooses an assignment to the variables in X, and then Player 2 challenges the assignment by choosing a clause of φ. The objective of Player 1 is to not get stuck in a sink, and the objective of Player 2 is to get stuck in the sink. Whenever Player 1 chooses an assignment to a variable, Player 2 has an opportunity to go to the sink, and Player 1 has to buy an edge in order to prevent her from doing so. The reward R_1 for Player 1 is $n+1$, and so Player 1 can buy n edges and still have utility 1. If Player 1 chooses an assignment that satisfies φ, then she can prevent the game from going to the sink by buying only n edges – one for each variable. Otherwise, Player 2 can choose a clause that is not satisfied by the assignment, which forces Player 1 to buy more than n edges or give up the prevention of the sink. ◀

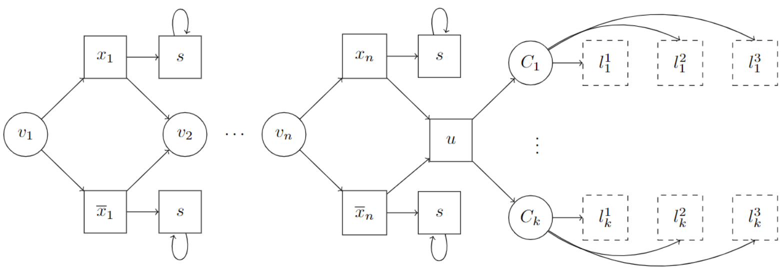

Figure 6 The game graph G_{SAT}. The circles are vertices owned by Player 1, and the squares are vertices owned by Player 2. The dashed vertices are the corresponding literal vertices on the assignment part of the graph.

6 Non-cooperative Rational Synthesis in Parity Trading Games

In this section we study NRS for PTGs. Recall that in PGs, the NRS problem is in PSPACE when the number of players is fixed, and can be solved in exponential time when their number is not fixed [24]. In BGs, NRS can be solved in polynomial time when the number of players is fixed, and the problem is PSPACE-complete when the number of players is not fixed. We show that the NRS problem in PTGs and BTGs is NP-complete for games with two players, and is Σ_2^{P}-complete for games with three or more players.

6.1 Two-player NRS

Consider a game $\mathcal{G} = \langle G, \{\alpha_1, \alpha_2\}, \{R_1, R_2\}\rangle$, a strategy $f_1 = (b_1, s_1)$ for Player 1, and a threshold $t \geq 0$. We describe an algorithm that determines if f_1 is an NRS solution for t in polynomial time. The key idea behind our algorithm is as follows. Let U_2 be the maximal utility for Player 2 in a profile π that extends f_1. Then, as Player 2 can ensure she gets utility of U_2, we have that every profile π in which $\mathsf{util}_2(\pi) = U_2$ is a 1-fixed NE, and every profile π in which $\mathsf{util}_2(\pi) < U_2$ is not a 1-fixed NE. Hence, f_1 is an NRS solution iff for every profile π that extends f_1 with $\mathsf{util}_2(\pi) = U_2$, we have that $\mathsf{util}_1(\pi) \geq t$.

We now describe the algorithm in detail. The algorithm first labels the edges from every vertex $v \in V$ by costs in $\mathbb{N}$. Recall the weights $\mathsf{cost}(\pi, e)$ described in Section 4 in the context of deviations for Player i. Observe that $\mathsf{cost}(\pi, e)$ is independent of the strategy f_i of Player i in π. In particular, when we consider deviations for Player 2, we have that $\mathsf{cost}(\pi, e)$ depends only on the function f_1 of Player 1, and can thus be denoted $\mathsf{cost}(f_1, e)$.

▶ **Lemma 11.** *Checking whether a given strategy for Player 1 is an NRS solution in a PTG can be done in polynomial time.*

Proof. Consider a PTG $\mathcal{G} = \langle G, \{\alpha_1, \alpha_2\}, \{R_1, R_2\}\rangle$, a strategy f_1 for Player 1, and a threshold $t \geq 0$. Let $G = \langle V, E\rangle$.

1. Let $G' = \langle V, E, w\rangle$ be a weighted version of G, where for every edge $e \in E$, we have that $w(e) = \mathsf{cost}(f_1, e)$.
2. For every $W \subseteq \{1, 2\}$, let ρ_W be the shortest lasso in G' such that the set of winners in ρ_W is W. Let f_2^W denote the corresponding strategy for Player 2.
3. Let $U_2 = \max\{\mathsf{util}_2(\langle f_1, f_2^W\rangle) : W \subseteq \{1, 2\}\}$. Note that U_2 is the maximal utility that Player 2 can get when the strategy for Player 1 is f_1.
4. If there exists a set $W \subseteq \{1, 2\}$ such that $\mathsf{util}_2(\langle f_1, f_2^W\rangle) = U_2$ and $\mathsf{util}_1(\langle f_1, f_2^W\rangle) < t$, then f_1 is not a NRS solution. Otherwise, f_1 is an NRS solution. ◀

Lemma 11 implies an NP upper bound for NRS for 2-players PTGs. A matching lower bound is proven by a reduction from 3SAT.

▶ **Theorem 12.** *NRS for 2-players PTGs is NP-complete. Hardness in NP holds already for BTGs.*

6.2 n-player NRS for $n \geq 3$

We continue and study NRS for PTGs with strictly more than two players. As bad news, we show that the polynomial algorithm from the proof of Theorem 12 cannot be generalized for NRS with three or more players. Intuitively, the reason is as follows. In the case of two players, there is a single environment player, and when the strategy for the system player is fixed, we could find the maximal possible utility for the environment player. On the other hand, when there are two or more environment players, the maximal possible utility for each of them depends on both the strategy of the system player and the strategies of the other environment players, which are not fixed. Formally, we prove that NRS for PTGs with strictly more than two players is Σ_2^{P}-complete. As good news, NRS stays Σ_2^{P} also when the number of players in not fixed; thus is is easier than NRS in PGs, where the problem is PSPACE-hard for an unfixed number of players.

▶ **Theorem 13.** *NRS for n-players PTGs with $n \geq 3$ is Σ_2^{P}-complete. Hardness in Σ_2^{P} holds already for BTGs.*

Proof. We start with the upper bound. We say that a profile π is *good* if $\mathsf{util}_1(\pi) \geq t$, or π is not a 1-fixed NE. Checking whether a given profile π is good can be done in polynomial time. Indeed, for checking whether $\mathsf{util}_1(\pi) \geq t$, we can find $\mathsf{S}(\pi)$ and $\mathsf{Outcome}(\pi)$, and then calculate $\mathsf{util}_1(\pi)$ in polynomial time. For checking whether π is not a 1-fixed NE, we can use Theorem 2 and check if some player $i \in [n] \setminus \{1\}$ has a beneficial deviation. Hence, an algorithm in Σ_2^{P} for NRS guesses a strategy f_1 for Player 1 and then checks that for all guessed strategies $f_2, \ldots, f_n$ for Players $2 \ldots n$, the profile $\langle f_1, f_2, \ldots, f_n \rangle$ is good. Note that the complexity is independent of n being fixed.

We continue to the lower bound and show that NRS is Σ_2^{P}-hard already for three players in BTGs. We describe a reduction from QBF_2, the problem of determining the truth of quantified Boolean formulas with one alternation of quantifiers, where the external quantifier is “exists”. Consider a QBF_2 formula $\Phi = \exists x_1, \ldots, x_n \forall y_1, \ldots, y_m \varphi$. We assume that φ is a Boolean propositional formula in 3DNF. Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$. Given Φ, we construct a 3-player Büchi game such that there exists an NRS solution f_1 in $\mathcal{G}$ for $t = 1$ iff $\Phi = \mathbf{true}$.

The main idea of the reduction is to construct a game in which Player 1 chooses an assignment to the variables in X; Player 2 tries to prove that $\Phi = \mathbf{false}$, by showing that there exists an assignment to the variables in Y with which for every clause C_i, there is a literal l_i^j such that $l_i^j = \mathbf{false}$; and Player 3 can point out whenever Player 2's proof is incorrect. The game has a sink s. The objective of Player 1 and Player 3 is to not get stuck in the sink, and the objective of Player 2 is V. That is, Player 2 wins in every path in the game. The reward to Player 1 is $n+1$, and she can pay 1 for each assignment in order to ensure that the play does not reach s. If Player 1 chooses an assignment for the variables in X such that for every assignment to the variables in Y, we have that φ is satisfied, then she and Player 3 can prevent the game from going to s, with Player 1 paying a total price of n. Otherwise, Player 2 can prove that $\Phi = \mathbf{false}$, and by that forces the play to reach s, unless Player 1 pays more than n, which exceeds her reward. ◀

7 Discussion

We introduced trading games, which extend ω-regular graph games with trading of control. Our buying and selling strategies concern edges in the game graph, and the result of the trading is a set of sold edges. In this section we discuss richer settings, classified according to the parameter they extend the setting with.

Buying strategies. We see two interesting ways to enrich buying strategies. The first, which is common in game theory, is to allow *dependencies* between the sold goods, thus let players bid on sets of edges [33]. Indeed, a company may be willing to pay for the rights to direct the traffic in a certain router in a communication network only if it also gets the right to direct traffic in a certain neighbour router. While it is not hard to extend our results to a setting with such dependencies, it makes the description of strategies more complex. The second way concerns the type of control that is traded. Rather than buying edges, a player may buy ownership of vertices. In the case of games with objectives that only require memoryless strategies, the difference boils down to *information*: the new owner is still going to use the same edge in all visits to a vertex she bought, yet unlike in our setting, the seller of the vertex does not known which edge it is. For games in which memoryless strategies are too weak (for example, games with generalized parity objectives, or objectives in LTL [21]), the suggested model allows the buyer to proceed with different edges in different visits to the sold vertex. Moreover, by allowing buying strategies that specify scenarios in which control is wanted, we can let players share control on a vertex. Thus, buying strategies may involve regular expressions that specify conditions on the history of the computation, and the suggested prices depend on these conditions. For example, a user may be willing to pay for an edge that guarantees a certain service only after certain events have happened.

Pricing and deviations. In our setting, payments are made for all the sold edges. It is not hard to see that stability can be increased by charging players only for edges that actually participate in the outcome of the profile. On the other hand, the latter charging policy encourages players to bid for more edges. Also, in our setting, a player can deviate from a profile only if unilaterally changing her buying or selling strategies increases her utility. This deviation rule prevents players from initiating a trade, even if both the seller and buyer benefit from it. This motivates the definition of joined deviations, where, for example, two players can deviate together by offering and accepting an offer, respectively, as long as they both increase their utilities.

Game graphs. The fact our games are turned-based makes the ownership of control simple: Player i controls and may sell the vertices in V_i. It is possible, however, to trade control also in *concurrent* games. There, the movement of the token depends on actions taken by all the players in all the vertices. Two natural ways to trade control in a concurrent setting are transverse – when players buy the right to choose an action for the seller in certain vertices, or longitudinal – when each player has a set of variables she controls, and an action amounts to assigning values to these variables. Then, players may buy variables, namely the right to assign values to these variable throughout the computation. For example, in a system with users that direct robots in warehouse by assigning them a direction and speed, a user may sell the control on her robot in certain locations in the warehouse, or sell the ability to decide its speed throughout the computation. Finally, as in other game-graphs studied in formal methods, it is interesting to study extensions to richer settings, addressing incomplete information, infinite domains, stochastic behavior, and more.

References

1. M. Abadi, L. Lamport, and P. Wolper. Realizable and unrealizable concurrent program specifications. In *Proc. 25th Int. Colloq. on Automata, Languages, and Programming*, volume 372 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 1989.
2. S.V. Albrecht and M.J. Wooldridge. Multi-agent systems research in the united kingdom. *AI Commun.*, 35(4):269–270, 2022.
3. S. Almagor, U. Boker, and O. Kupferman. Formalizing and reasoning about quality. *Journal of the ACM*, 63(3):24:1–24:56, 2016.

4 S. Almagor, D. Kuperberg, and O. Kupferman. Sensing as a complexity measure. *Int. J. Found. Comput. Sci.*, 30(6-7):831–873, 2019.

5 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.

6 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.

7 M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu. Safe reinforcement learning via shielding. In *Proc. of 32nd Conf. on Artificial Intelligence*, pages 2669–2678. AAAI Press, 2018.

8 E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *Proc. 45th IEEE Symp. on Foundations of Computer Science*, pages 295–304. IEEE Computer Society, 2004.

9 G. Avni, R. Bloem, K. Chatterjee, T. A. Henzinger, B. Könighofer, and S. Pranger. Run-time optimization for learned controllers through quantitative games. In *Proc. 31st Int. Conf. on Computer Aided Verification*, volume 11561 of *Lecture Notes in Computer Science*, pages 630–649. Springer, 2019.

10 G. Avni, P. Ghorpade, and S. Guha. A game of pawns, 2023. arXiv:2305.04096.

11 G. Avni and T.A. Henzinger. An updated survey of bidding games on graphs. In *47th Int. Symp. on Mathematical Foundations of Computer Science*, volume 241 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:6. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

12 G. Avni, T.A. Henzinger, and V. Chonev. Infinite-duration bidding games. *Journal of the ACM*, 66(4):31:1–31:29, 2019.

13 G. Avni, T.A. Henzinger, and D. Zikelic. Bidding mechanisms in graph games. *J. Comput. Syst. Sci.*, 119:133–144, 2021.

14 G. Avni, I. Jecker, and D. Zikelic. Infinite-duration all-pay bidding games. In *Symposium on Discrete Algorithms*, pages 617–636. SIAM, 2021.

15 G. Avni and O. Kupferman. Synthesis from component libraries with costs. In *Proc. 25th Int. Conf. on Concurrency Theory*, volume 8704 of *Lecture Notes in Computer Science*, pages 156–172. Springer, 2014.

16 BBC. Should billboard advertising be banned? https://www.bbc.com/news/business-62806697, 2022.

17 R. Bloem, K. Chatterjee, and B. Jobstmann. Graph games and reactive synthesis. In *Handbook of Model Checking.*, pages 921–962. Springer, 2018.

18 P. Bouyer-Decitre, O. Kupferman, N. Markey, B. Maubert, A. Murano, and G. Perelli. Reasoning about quality and fuzziness of strategic behaviours. In *Proc. 28th Int. Joint Conf. on Artificial Intelligence*, pages 1588–1594, 2019.

19 V. Bruyère. Synthesis of equilibria in infinite-duration games on graphs. *ACM SIGLOG News*, 8(2):4–29, 2021.

20 V. Bruyère. *A Game-Theoretic Approach for the Synthesis of Complex Systems*, pages 52–63. Springer International Publishing, 2022.

21 K. Chatterjee, T. A. Henzinger, and N. Piterman. Generalized parity games. In *Proc. 10th Int. Conf. on Foundations of Software Science and Computation Structures*, volume 4423 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2007.

22 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.

23 S. Chaudhuri, S. Kannan, R. Majumdar, and M.J. Wooldridge. Game theory in AI, logic, and algorithms (dagstuhl seminar 17111). *Dagstuhl Reports*, 7(3):27–32, 2017.

24 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.

25 R. Condurache, Y. Oualhadj, and N. Troquard. The Complexity of Rational Synthesis for Concurrent Games. In *Proc. 29th Int. Conf. on Concurrency Theory*, volume 118 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

26 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.

27 O. Kupferman. Examining classical graph-theory problems from the viewpoint of formal-verification methods. In *Proc. 49th ACM Symp. on Theory of Computing*, page 6, 2017.

28 O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. *Annals of Mathematics and Artificial Intelligence*, 78(1):3–20, 2016.

29 O. Kupferman and N. Piterman. Lower bounds on witnesses for nonemptiness of universal co-Büchi automata. In *Proc. 12th Int. Conf. on Foundations of Software Science and Computation Structures*, volume 5504 of *Lecture Notes in Computer Science*, pages 182–196. Springer, 2009.

30 O. Kupferman and N. Shenwald. On the complexity of LTL rational synthesis. In *Proc. 28th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 13243 of *Lecture Notes in Computer Science*, pages 25–45, 2022.

31 Meta. Introduction to the advertising standards. `https://transparency.fb.com/policies/ad-standards/`, 2023.

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 N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

34 C. H. Papadimitriou. Algorithms, games, and the internet. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 749–753, 2001.

35 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.

36 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.

37 J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1953.