The Complexity of Infinite-Horizon General-Sum Stochastic Games

Yujia Jin
Stanford University, CA, USA

Vidya Muthukumar
Georgia Institute of Technology, Atlanta, GA, USA

Aaron Sidford
Stanford University, CA, USA

Abstract

We study the complexity of computing stationary Nash equilibrium (NE) in n-player infinite-horizon general-sum stochastic games. We focus on the problem of computing NE in such stochastic games when each player is restricted to choosing a stationary policy and rewards are discounted. First, we prove that computing such NE is in PPAD (in addition to clearly being PPAD-hard). Second, we consider turn-based specializations of such games where at each state there is at most a single player that can take actions and show that these (seemingly-simpler) games remain PPAD-hard. Third, we show that under further structural assumptions on the rewards computing NE in such turn-based games is possible in polynomial time. Towards achieving these results we establish structural facts about stochastic games of broader utility, including monotonicity of utilities under single-state single-action changes and reductions to settings where each player controls a single state.

2012 ACM Subject Classification Theory of computation → Complexity classes

Keywords and phrases complexity, stochastic games, general-sum games, Nash equilibrium

Digital Object Identifier 10.4230/LIPIcs.ITCS.2023.76

Related Version *Full Version*: `https://arxiv.org/abs/2204.04186`

Funding Researchers are supported in part by an Adobe Data Science Research Award, a Danzig-Lieberman Graduate Fellowship, a Google Research Colabs Award, a Microsoft Research Faculty Fellowship, NSF CAREER Award CCF-1844855, NSF Grant CCF-1955039, NSF Grant IIS-2212182, a PayPal research award, a Sloan Research Fellowship and a Stanford Graduate Fellowship.

Acknowledgements The authors thank Constantinos Daskalakis, Noah Golowich, and Kaiqing Zhang for kindly coordinating on uploads to arXiv. The authors also thank Aviad Rubinstein and anonymous reviewers for helpful feedback. Part of this work was conducted while the authors were visiting the Simons Institute for the Theory of Computing.

1 Introduction

Stochastic games [24, 4] are a fundamental mathematical model for dynamic, non-cooperative interaction between multiple players. Multi-player dynamic interaction arises naturally in a diverse set of contexts including natural resource competition [42], monetary interaction in markets [38], packet routing [1], and computer games [64, 63]. Such games have also been of increased study in reinforcement learning (RL); there have been a number of successes in transferring results from single-player RL to multiplayer RL under zero-sum and cooperative interaction, but comparatively less success for general-sum interaction (see e.g. [72] for a survey).

We consider the broad class of general-sum, simultaneous, tabular, n-player stochastic games [61, 27, 67], which we henceforth refer to as SimSGs.[1] SimSGs are parameterized by a (finite) state space $\mathcal{S}$ and disjoint (finite) action sets $\mathcal{A}_{i,s}$ for each player i and state s. The players choose a joint strategy $\boldsymbol{\pi}$, consisting of distributions $\boldsymbol{\pi}^t_{i,s}$ over the actions $\mathcal{A}_{i,s}$ for each player $i \in [n]$ at each state $s \in \mathcal{S}$ at time-step $t \geq 0$. The game then proceeds in time-steps, where in each time-step $t \geq 0$, the game is at a state $s^t \in \mathcal{S}$ and each player $i \in [n]$ samples independently from $\mathcal{A}_{i,s^t}$ according to $\boldsymbol{\pi}^t_{i,s^t}$. The set of actions $\mathbf{a}^t$ chosen at time-step t then yields an immediate reward $\mathbf{r}_{i,s,\mathbf{a}^t}$ to each player i, and causes the next state s^{t+1} to be sampled from a distribution $\mathbf{p}_{s,\mathbf{a}^t}$. Each player i aims to maximize her own long-term value as a function of the rewards they receive, i.e. $\mathbf{r}_{i,s,\mathbf{a}^t}$. For any fixed strategy the states in a SimSG evolve as a Markov chain[2] and the single-player specialization of SimSGs, i.e. when $n = 1$, is a Markov decision processes (MDP) [5, 57].

Our focus in this paper is on computing (*approximate*) *Nash equilibrium* in the multiplayer *general-sum setting* of SimSGs. The term *general-sum* emphasizes that we do not impose any shared structure on the immediate reward functions across players (in contrast to the special case of zero-sum games where $n = 2$ and $\mathbf{r}_2 = -\mathbf{r}_1$). A *Nash equilibrium (NE)* [51] is defined as a joint strategy $\boldsymbol{\pi}$ such that no player can gain in reward by deviating (keeping the other players' strategies fixed). NE is a solution concept of fundamental interest and importance in both static [50] and dynamic games [24, 4]. While NE are known to always exist in SimSG [61, 27, 67], they are challenging to compute efficiently; current provably efficient algorithms from computing (approximate) NE for SimSGs make strong assumptions on the rewards [33, 43].

One setting for which the complexity of computing general-sum NE is relatively well-understood is the *finite-horizon* model, where all players play up to a horizon of finite and known length H and wish to optimize their total reward. Even with just two players, and a single state (or multiple states but a horizon length $H = 1$), the problem of NE computation in SimSG is PPAD-hard as it generalizes computing NE for a two-player normal-form game which is known to be PPAD-complete [9, 16]. On the other hand, by leveraging stochastic dynamic programming techniques [24], one can show that the complexity of NE computation in finite-horizon SimSG and normal-form games is polynomial-time equivalent: in particular, NE-computation remains PPAD-complete. This dynamic programming technique is also broadly applicable to solution concepts that are comparatively tractable, such as correlated equilibrium (CE) [54]. Exploiting this property, recent work [34, 65, 46] has shown that simple decentralized RL algorithms can provably learn and converge to the set of CE's in a finite-horizon SimSG.

The central goal of this work is to broaden our understanding of the complexity of SimSGs. We ask, "*how brittle is the property of* PPAD*-completeness of finite-horizon* SimSG*s?*", specifically to:

- ***Infinite time horizon:*** *what if players optimize rewards over an infinite time horizon?*
- ***Turn-based games:*** *what if each state is controlled only by a single player?*
- ***Localized rewards:*** *what if rewards are only received for a player at states they control?*

[1] See Section 2 for the more formal definition and description of our notational conventions.

[2] This Markov structure is commonly assumed across the stochastic games literature, particularly when stationary strategies are considered [61, 24, 12] and some recent literature [3, 34, 65] refers to SimSGs as *Markov games*. There are studied generalizations of SimSGs that allow non-stationary or non-Markovian dynamics [4], but are outside the scope of this paper.

In this paper we systematically address these questions and provide theoretical foundations for understanding the complexity of infinite-horizon stochastic games. Our key results include complexity-class characterizations, algorithms, equivalences and structural results regarding such games. For a brief summary of our main complexity characterizations, see Table 1.

Infinite time horizon. First, we consider *infinite-horizon* SimSGs in which each player seeks to maximize rewards over an infinite time horizon while following a *stationary strategy*. A stationary strategy is one in which action distributions are independent of the time-step (i.e. $\boldsymbol{\pi}_{i,s}^{t_1} = \boldsymbol{\pi}_{i,s}^{t_2}$ for all $i \in [n]$, $s \in \mathcal{S}$, and $t_1, t_2 \geq 0$). We focus on the discounted-reward model, and defer discussion of the alternative average-reward model to Appendix C in full version. (The single-player version of such games is known as a discounted Markov decision process (DMDP) and has been the subject of extensive study in optimization [70], operations research [57], and machine learning [66].) Stationary strategies are especially attractive to study owing to their succinctness in representation compared to non-stationary strategies and the fact that stationary *policies* (the 1-player analog of strategies) can attain the optimal value in single-player DMDPs [5, 57].

Despite the fact that stationary NE are always known to exist in infinite-horizon SimSGs [27, 67], existence does not appear to directly follow from the straightforward proof of existence in finite-horizon SimSG. In particular, the dynamic programming technique for finite-horizon SimSGs breaks down for infinite-horizon SimSGs [73] and does not directly imply membership in PPAD. Nevertheless, as described in Section 3.2, we show that the stationary NE-computation problem for SimSG remains in PPAD. To prove this result we establish a number of key properties of discounted SimSGs (and, thereby, an alternative NE existence proof) that are crucial for several of the results in this paper and may be independently useful for future SimSG algorithm design (see Section 3.1).

Turn-based games. We then consider *turn-based* variants of SimSGs, which we henceforth refer to as TBSG. Formally, TBSGs are the specialization of SimSGs where for each state there is at most one-player that has a non-trivial set of distinct actions to choose from. TBSGs are common in the literature and encompass the popular instantiations of game-play for which large-scale RL has yielded empirical success [64, 63]. Additionally, they have been extensively studied in the case of two players and zero-sum rewards [61, 12, 23, 31, 62].

Whereas it was natural to suspect that discounted SimSGs would be PPAD-complete, the computational complexity of computing NE for TBSGs seems less clear. The trivial proof of PPAD-hardness for SimSGs breaks down even for the case of multiplayer TBSG – specializing to a single-state game reduces the problem to trivial independent reward maximization by each player, rather than a simultaneous normal-form game. More generally, TBSGs seem to have more special structure than SimSGs owing to the restriction of a single player controlling each state. As a quick illustration of this structure, note that non-stationary NE for general-sum, *finite-horizon* TBSGs can be computed in polynomial time by a careful application of the multi-agent dynamic programming technique. Further, in Section 7.1 of full version, we extend this technique to show that *non-stationary* NE for TBSGs can be computed in polynomial time for a polynomially bounded discount factor.

Despite this seemingly special structure of TBSGs, one of the main contributions of our work (described in Section 3.3) is to show that computing a multiplayer *stationary* NE for TBSG is PPAD-hard even for a constant discount factor $\gamma \in (0, 1)$. This shows a surprising and non-standard divergence between the non-stationary and stationary solution concepts in infinite-horizon stochastic games. Moreover, it even implies the hardness of stationary

coarse-correlated equilibrium (CCE) computation in SimSGs (owing to a stationary NE in TBSGs being a special case), which is a relaxed notation of equilibrium that allows for more computationally-efficient methods in two-player normal-form games (in contrast to SimSGs). Our hardness results hold even for TBSGs for which each player controls a different state, and each player receives a non-zero reward (allowed to be either positive or negative) at at most 4 states, including her own.

Localized rewards. Finally, with the hardness of discounted general-sum SimSGs and TBSGs established, we ask *"under what further conditions on reward functions are there polynomial-time algorithms for* TBSG*s?"* As described in Section 3.4, we show that further localizing the reward structure such that each player receives a reward of the same sign *only* at a single state which she controls changes the complexity picture and leads to a polynomial-time algorithm. We show that for these specially structured TBSGs, a pure NE always exists and is polynomial-time computable via approximate best-response dynamics (also called *strategy iteration* in the stochastic games literature [31]). These results are derived via a connection to potential games [49] modulo a monotonic transformation of the utilities. While the connection to potential game theory yields *approximate* NE, we also design a more combinatorial, graph-theoretic algorithm that computes *exact* NE in polynomial-time if, additionally, the transitions in TBSG are deterministic.

Summary and additional implications. In summary, we show that (a) stationary NE computation for infinite-horizon SimSG$'s$ is in PPAD, (b) stationary NE computation for infinite-horizon TBSGs is PPAD-hard, and (c) stationary pure NE computation for infinite-horizon TBSGs when each player receives a consistently-signed reward at one controlled state is polynomial-time solvable.

Beyond shedding light on the complexity of infinite horizon general-sum stochastic games, our work yields several insights and implications of additional interest. On the one hand, our hardness result for stationary NE in infinite-horizon TBSG implies the hardness of slightly more complex solution concepts such as stationary coarse-correlated equilibrium (CCE) in SimSG (as the former is a special case of the latter). On the other hand, our PPAD membership result for SimSG (which includes TBSG as a special case) is interesting as in SimSGs the utility that a player receives is a non-convex function of her actions, and general-sum non-convex games lie in a complexity class suspected to be harder than PPAD [60]; indeed, even the zero-sum case is PPAD-hard [18]. Further, many of the results in this paper crucially utilize special structure that we prove (in Lemma 1 of full version) of a monotonic change with upper and lower-bounded slope (which we refer to as *pseudo-linear*) on each player's value function when she changes her policy at only *one* state. This observation has powerful consequences for many of our results and allows us to leverage several algorithmic techniques that are normally applied only to linear and piecewise-linear utilities. As one example, it yields a particularly simple existence proof of stationary NE in SimSG compared to past literature [27, 67]. We hope these results facilitate the further study of infinite-horizon stochastic games.

Paper Organization. We cover notation and fundamental definitions in Section 2, an overview of our results and techniques in Section 3, and related work in Section 4. Main results and technical details are provided in more detail in the full version.

Table 1 Summary of SimSG complexity characterization. Characterizations are for computing non-stationary NE in the finite-horizon case, and for computing stationary NE in the infinite-horizon case.

Setting	SimSG	TBSG	TBSG (localized rewards)
Finite-horizon	PPAD	Polynomial	Polynomial
Infinite-horizon	PPAD	PPAD	Polynomial

2 Preliminaries

Here we introduce notation and basic concepts for SimSGs and TBSGs we use throughout the paper.

Simultaneous stochastic games (SimSGs). This paper focuses on computing NE of *multi-agent general-sum simultaneous stochastic games (*SimSG*s)* in infinite-horizon settings. Unless stated otherwise, we consider discounted infinite-horizon SimSGs and denote an instance by tuple $\mathcal{G} = (n, \mathcal{S}, \mathcal{A}, \mathbf{p}, \mathbf{r}, \gamma)$. n denotes the number of players (agents), $\mathcal{S}$ denotes a finite state space, and $\mathcal{A}$ denotes the finite set of actions available to the players where for player $i \in [n]$ and $s \in \mathcal{S}$ the possible actions of player i at states s are $\mathcal{A}_{i,s}$. We say player $i \in [n]$ *controls state* $s \in \mathcal{S}$ if $\mathcal{A}_{i,s} \neq \emptyset$. We use $\mathcal{I}_s = \{i \in [n] | \mathcal{A}_{i,s} \neq \emptyset\} \subseteq [n]$ to denote the players controlling state s, and $\mathcal{A}_s$ to denote the joint action space of all players controlling state s, i.e. for any $\mathbf{a}_s \in \mathcal{A}_s$, $\mathbf{a}_s = (a_{i,s})_{i \in \mathcal{I}_s}$ where $a_{i,s} \in \mathcal{A}_{i,s}$. We denote the action space size for player i by $A_{\mathsf{tot},i} := \sum_{s \in \mathcal{S}} |\mathcal{A}_{i,s}|$ and the joint action space size by $A_{\mathsf{tot}} := \sum_{i \in [n]} A_{\mathsf{tot},i}$. We let $\mathbf{p}$ denote the transition probabilities, where $\mathbf{p}_{s,\mathbf{a}_s} \in \Delta^{\mathcal{S}} := \{\mathbf{x} \in \mathbb{R}^{\mathcal{S}}_{\geq 0} | \sum_{s \in \mathcal{S}} x_s = 1\}$ is a distribution over states for all $s \in \mathcal{S}$ and $\mathbf{a}_s \in \mathcal{A}_s$. $\mathbf{r}$ denotes the instantaneous rewards, where $r_{i,s,\mathbf{a}_s}$ with $|r_{i,s,\mathbf{a}_s}| \leq 1$ is the reward of player i at state s if the players controlling it play $\mathbf{a}_s \in \mathcal{A}_s$. $\gamma \in (0, 1)$ denotes a discount factor.

SimSG notation and simplifications. Recall that we use $\mathcal{I}_s = \{i \in [n] | \mathcal{A}_{i,s} \neq \emptyset\} \subseteq [n]$ to denote the players controlling state s. Additionally, we use $\mathcal{S}_i = \{s \in \mathcal{S} | \mathcal{A}_{i,s} \neq \emptyset\} \subseteq \mathcal{S}$ to denote states that are controlled by player i. Without loss of generality, we assume that for each player i there exists at least one state $s \in \mathcal{S}$ where $\mathcal{A}_{i,s} \neq \emptyset$ (i.e. $|\mathcal{S}_i| \geq 1$), since otherwise we can remove the corresponding player i from the game. Also, we assume for each state $s \in \mathcal{S}$ there is at least a player i such that $\mathcal{A}_{i,s} \neq \emptyset$ (i.e. $|\mathcal{I}_s| \geq 1$). This is because for any $s \in \mathcal{S}$, if $\mathcal{A}_{i,s} = \emptyset$ for all $i \in [n]$ and the transition from the state is $\mathbf{p}_s \in \Delta^{\mathcal{S}}$, this is equivalent to setting $\mathcal{A}_{1,s} = \{a_s\}$ and $\mathbf{p}_{s,a_s} = \mathbf{p}_s$.

SimSG model and objectives. A SimSG proceeds as follows. It starts from time step $t = 0$ and initial state $s^0 \in \mathcal{S}$ drawn from initial distribution $\mathbf{q}$. In each turn $t \geq 0$ the game is at a state s^t. At state s^t, each player $i \in \mathcal{I}_{s^t}$ plays an action $a_i^t \in \mathcal{A}_{i,s^t}$. The joint action $\mathbf{a}^t = (a_i^t)_{i \in \mathcal{I}_{s^t}} \in \mathcal{A}_{s^t}$ then yields reward $r_{i,s^t,\mathbf{a}^t}$ for each player $i \in [n]$.The next state s^{t+1} is then sampled (independently) by $\mathbf{p}_{s^t,\mathbf{a}^t} \in \Delta^{\mathcal{S}}$. The goal of each player $i \in [n]$ is to maximize their expected infinite-horizon discounted reward, or known as value *of the game for player* i, defined as $v_i = \mathbb{E}[\sum_{t \geq 0} \gamma^t r_{i,s^t,\mathbf{a}^t}]$.

SimSG policies and strategies. Unless stated otherwise, for each player $i \in [n]$ we restrict to considering randomized stationary policies, i.e. $\boldsymbol{\pi}_i = (\boldsymbol{\pi}_{i,s})_{s \in \mathcal{S}_i}$ where $\boldsymbol{\pi}_{i,s} \in \Delta^{\mathcal{A}_{i,s}}$, and use $\pi_{i,s}(a)$ to denote the probability of player i playing action $a \in \mathcal{A}_{i,s}$ at state s. We call a

collection of policies for all players, i.e. $\boldsymbol{\pi} = (\boldsymbol{\pi}_i)_{i\in[n]}$, a *strategy*. For a strategy $\boldsymbol{\pi}$ we use $\boldsymbol{\pi}_{-i}$ to denote the collection of policies of all players other than player i, i.e. $\boldsymbol{\pi}_{-i} := (\boldsymbol{\pi}_j)_{j\in[n]\setminus\{i\}}$; we do not distinguish between orders of $\boldsymbol{\pi}_{i,s}$ in the set $\boldsymbol{\pi}$ when clear from context (e.g. see definition of NE in (4)). Further, we use $\mathbf{P}^{\boldsymbol{\pi}} \in \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ and $\mathbf{r}^{\boldsymbol{\pi}}$ to denote the probability transition kernel and instantaneous reward, respectively, under strategy $\boldsymbol{\pi}$, where

$$\mathbf{P}^{\boldsymbol{\pi}}(s,\cdot) := \sum_{\mathbf{a}_s\in\mathcal{A}_s}\left(\prod_{i\in\mathcal{I}_s}\pi_{i,s}(a_{i,s})\right)\mathbf{p}_{s,\mathbf{a}_s} \in \Delta^{\mathcal{S}} \text{ and } \mathbf{r}_i^{\boldsymbol{\pi}}(s) := \sum_{\mathbf{a}_s\in\mathcal{A}_s}\left(\prod_{i\in\mathcal{I}_s}\pi_{i,s}(a_{i,s})\right)r_{i,s,\mathbf{a}_s}. \tag{1}$$

Under strategy $\boldsymbol{\pi}$, we define the value function of each player $i \in [n]$ at state $s \in \mathcal{S}$ to be

$$V_i^{\boldsymbol{\pi}}(s) := \mathbb{E}\left[\sum_{t\geq 0}\gamma^t r_{i,s^t,\mathbf{a}^t} | s_0 = s, a_{j,s^t}^t \sim \boldsymbol{\pi}_{j,s^t} \text{ for all } j,t\right] = \mathbf{e}_s^\top(\mathbf{I}-\gamma\mathbf{P}^{\boldsymbol{\pi}})^{-1}\mathbf{r}^{\boldsymbol{\pi}}\ . \tag{2}$$

The value of a strategy to player i starting at initial distribution $\mathbf{q} \in \Delta^{\mathcal{S}}$ is defined as

$$v_i^{\boldsymbol{\pi},\mathbf{q}} := \mathbb{E}_{\mathbf{q}}^{\boldsymbol{\pi}}\left[\sum_{t\geq 0}\gamma^t r_{i,s^t,\mathbf{a}^t}\right] = \mathbb{E}\left[\sum_{t\geq 0}\gamma^t r_{i,s^t,\mathbf{a}^t} | s_0 \sim \mathbf{q}, a_{j,s^t}^t \sim \boldsymbol{\pi}_{j,s^t} \text{ for all } j,t\right] = \langle \mathbf{q}, \mathbf{V}_i^{\pi}\rangle. \tag{3}$$

Nash equilibrium (NE) in SimSGs. Given any $\epsilon \geq 0$, we call a strategy $\boldsymbol{\pi}$ an *ϵ-approximate Nash Equilibrium (NE) (ϵ-NE)* if for each player $i \in [n]$

$$u_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) \geq u_i(\boldsymbol{\pi}_i', \boldsymbol{\pi}_{-i}) - \epsilon, \quad \text{for any} \quad \boldsymbol{\pi}_{i,s}' \in \Delta^{\mathcal{A}_{i,s}}. \tag{4}$$

where $u_i(\boldsymbol{\pi})$ for all $i \in [n]$ is a real value (as a function of $\boldsymbol{\pi}$) referred to as *utility of player i under strategy $\boldsymbol{\pi}$*. For general SimSGs, unless specified otherwise, we let $u_i(\boldsymbol{\pi}) := u_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) = \langle \mathbf{q}, \mathbf{V}_i^{\boldsymbol{\pi}}\rangle$, i.e. the value function with initial distribution $\mathbf{q} = \frac{1}{|\mathcal{S}|}\mathbf{e}_{\mathcal{S}}$. Further, we call any 0-approximate NE an *exact NE* and when we refer to a NE we typically mean an ϵ-NE for inverse-polynomially small ϵ. We use the term *approximate NE* to refer to an ϵ-NE for constant ϵ.

Turn-based stochastic games (TBSGs). TBSGs are the class of SimSGs where each state is controlled by at most one player (i.e. $|\mathcal{I}_s| \leq 1$), or equivalently, the states controlled by each of the players are disjoint (i.e $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for any $i \neq j$, $i,j \in [n]$). Equivalently (by earlier assumptions), a TBSG is a SimSG with $|\mathcal{I}_s| = 1$ for all $s \in \mathcal{S}$; accordingly, we use $\mathcal{I}_s = \{i_s\}$ to denote the single player that is controlling state s in a TBSG. Since $\mathcal{S} = \cup_{i\in[n]}\mathcal{S}_i$ in a TBSG we denote an instance by $\mathcal{G} = (n, \mathcal{S} = \cup_{i\in[n]}\mathcal{S}_i, \mathcal{A}, \mathbf{p}, \mathbf{r}, \gamma)$. Following SimSG notation, we have $\mathcal{A} = (\mathcal{A}_{i,s})_{i\in[n],s\in\mathcal{S}_i}$ and $\mathcal{A}_s = \mathcal{A}_{i,s}$ if and only if $s \in \mathcal{S}_i$ as well as $\mathbf{p} = (\mathbf{p}_{s,a})_{s\in\mathcal{S},a_s\in\mathcal{A}_s}$ and $\mathbf{r} = (r_{i,s,a_s})_{i\in[n],s\in\mathcal{S},a_s\in\mathcal{A}_s}$. When clear from context, we also use $\boldsymbol{\pi}_s := \boldsymbol{\pi}_{i_s,s}$ for all $s \in \mathcal{S}$. Using this notation, the probability transition kernel $\mathbf{P}^{\boldsymbol{\pi}} \in \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ and instantaneous reward $\mathbf{r}^{\boldsymbol{\pi}} \in \mathbb{R}^{\mathcal{S}}$ under strategy $\boldsymbol{\pi}$ are

$$\mathbf{P}^{\boldsymbol{\pi}}(s,\cdot) = \sum_{a_s\in\mathcal{A}_s}\pi_{i_s,s}(a_s)\mathbf{p}_{s,a_s} \in \Delta^{\mathcal{S}} \text{ and } \mathbf{r}_i^{\boldsymbol{\pi}}(s) = \sum_{a_s\in\mathcal{A}_s}\pi_{i_s,s}(a_s)r_{i,s,a_s}\ . \tag{5}$$

Game variations. Here we briefly discuss variants of discounted SimSGs we consider.

- **Number of players**: We focus on n-player games and our hardness results use that n can scale with the problem size. Establishing the complexity of computing general-sum NE for TBSGs with a constant number of players, e.g. $n = 2$, remains open.
- **Number of states each player controls**: We use O-SimSG (and O-TBSG) to denote the class of SimSGs (and TBSGs) where each player only controls one state $|\mathcal{S}_i| = 1$ (note that it is possible that $|\mathcal{I}_s| > 1$, for some $s \in \mathcal{S}$ in an O-SimSG). For simplicity, in O-TBSG instances we denote the state space by $\mathcal{S}_i = \{s_i\}$ for each $i \in [n]$ and thus $\mathcal{S} = \cup_{i\in[n]}\{s_i\}$ and let $\mathcal{A}_i := \mathcal{A}_{s_i} = \mathcal{A}_{i,s_i}$. For O-SimSGs and O-TBSGs, we use $v_i(\boldsymbol{\pi}) := V_i^{\boldsymbol{\pi}}(s_i)$ to denote the value of player i under strategy π with initial distribution $\mathbf{e}_{s_i}$. Unless specified otherwise, we use $v(\cdot)$ as the utility function in the definition of NE for O-SimSGs and O-TBSGs; in Appendix B of full version we prove that these two notions of approximate NE are equivalent up to polynomial factors.
- **Different types of strategies (and policies).** We focus on stationary strategies in the majority of this paper, but at times we consider non-stationary strategies where the distribution over actions chosen at each time-step is allowed to depend on t. Further, we call a policy $\boldsymbol{\pi}_i$ a *pure (or deterministic) policy* if it maps a state to a single action for that player, i.e. if $\boldsymbol{\pi}_{i,s} = \mathbf{e}_{a_{i,s}}$ for some $a_{i,s} \in \mathcal{A}_{i,s}$ for each $s \in \mathcal{S}_i$ and call a strategy $\boldsymbol{\pi} = (\boldsymbol{\pi}_i)_{i\in[n]}$ a *pure strategy* if all policies $\boldsymbol{\pi}_i$ are pure. Some of the results in paper restrict to consider pure strategies and we extend the definitions of NE to these cases by restricting to such strategies in (4).

3 Overview of results and techniques

Here we provide an overview of our main results and techniques for establishing the complexity of computing stationary NEs in discounted infinite-horizon general-sum SimSGs. First, in Section 3.1 we cover foundational structural results regarding such SimSGs that we use throughout the paper. In Section 3.2 we discuss how we show that the problem of computing stationary NE in such SimSGs is in PPAD. We then consider the TBSG specialization of this problem and discuss how we show that computing stationary NE in such TBSGs is PPAD-hard (Section 3.3), but polynomial-time solvable under additional assumptions on rewards (Section 3.4).

Although we focus on discounted SimSGs and TBSGs in the body of the paper, in Appendix C of full version we extend our results to the *average-reward model* where the rewards are not discounted, but instead amortized over time. We show that results analogous to our main results hold for under the assumption of bounded mixing times. These extensions are achieved by building upon tools established in [35] for related discounted and average-reward MDPs.

3.1 Foundational properties

Here we introduce two types of foundational structure we demonstrate for infinite-horizon SimSGs, which both our positive and negative complexity characterizations crucially rely on. These structures use the fact that when fixing the strategies of all but one of the players in a SimSG, the problem reduces to a single-agent DMDP.

The first property we observe is that when changing the action of a player at any single state in a SimSG from one distribution to another, the utility for that player changes in a *monotonic* manner, with slope that is both upper and lower-bounded. We refer to this type

of change as *pseudo-linear*. This property is equivalent to showing the following theorem that the utilities are pseudo-linear in the special class of SimSGs where each player controls only one state, i.e. O-SimSGs.

▶ **Theorem 1** (Pseudo-linear utilities in O-SimSGs, restating Corollary 1 in full version)**.** *Consider any* O-SimSG *instance* $\mathcal{G} = (n, \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{R}, \gamma)$ *any initial distribution* $\mathbf{q}$*, and some player* $i \in [n]$*. Her utility function* $u_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) = v_i^{\boldsymbol{\pi},\mathbf{q}}$*, when fixing other players' strategy* $\boldsymbol{\pi}_{-i}$*, is* pseudo-linear *in* $\boldsymbol{\pi}_i$*, i.e. for any* $\boldsymbol{\pi}_i$*,* $\boldsymbol{\pi}_i' \in \Delta^{\mathcal{A}_i}$ *ordered such that* $u_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) \leq u_i(\boldsymbol{\pi}_i', \boldsymbol{\pi}_{-i})$ *and any* $\theta \in [0, 1]$*, we have*

$$(1-\gamma)\theta(u_i(\boldsymbol{\pi}_i', \boldsymbol{\pi}_{-i}) - u_i(\boldsymbol{\pi})) \leq u_i(\theta\boldsymbol{\pi}_i' + (1-\theta)\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) - u_i(\boldsymbol{\pi}) \leq \frac{1}{1-\gamma}\theta(u_i(\boldsymbol{\pi}_i', \boldsymbol{\pi}_{-i}) - u_i(\boldsymbol{\pi})). \tag{6}$$

First, to see why O-SimSGs have pseudo-linear utilities, we note that when we consider a linear combination of policies $\boldsymbol{\pi}_i'$ and $\boldsymbol{\pi}_i$ for player i and fix the other players' strategy $\boldsymbol{\pi}_{-i}$, it is equivalent to considering a DMDP in which a single player linearly changes her policy on a single state s between two actions a and a'. In this case, the difference in transition matrices is of rank-1 and we can use the Sherman-Morrison formula to exactly characterize the change in utility as

$$u_i(\theta\boldsymbol{\pi}_i' + (1-\theta)\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) = u_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) + \theta \frac{\mathbf{q}^\top \mathbf{Q}\mathbf{e}_s \cdot \left[(r_{s,a'} - r_{s,a}) + \gamma(\mathbf{p}_{s,a'} - \mathbf{p}_{s,a})^\top \mathbf{Q}\mathbf{r}^{\boldsymbol{\pi}}\right]}{1 - \gamma\theta(\mathbf{p}_{s,a'} - \mathbf{p}_{s,a})^\top \mathbf{Q}\mathbf{e}_s} \tag{7}$$

where $\mathbf{Q} := \left(\mathbf{I} - \gamma\mathbf{P}^{(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i})}\right)^{-1}$. We then bound the difference in utilities arising from changing the transitions; we show that $(\mathbf{p}_{s,a'} - \mathbf{p}_{s,a})^\top \mathbf{Q}\mathbf{e}_s \in [-1/(1-\gamma), 1]$ by utilizing a specific Markov chain interpretation of the utilities. This implies the more fine-grained property in (6) that the utility function is *pseudo-linear* with bounded slope.

Theorem 1 describes powerful structure on the utility functions of each player that we leverage for our membership *and* hardness results. Although utilities for SimSGs may be non-linear and non-convex (in fact, even under a single-state policy change, (7) may be either convex or concave in θ depending on the sign of D and $[u_i(\boldsymbol{\pi}_i', \boldsymbol{\pi}_{-i}) - u_i(\boldsymbol{\pi})]$) with complex global correlations, for any fixed player Theorem 1 shows that utilities are not too far from linear.

This pseudo-linear structure is key to many of our subsequent proofs. For example, the pseudo-linear property in (6) implies a distinct proof of existence of NE for O-SimSG that is considerably simpler than the classic existence proofs for SimSGs [27, 67]. We describe how pseudo-linearity is used in each of our proofs of membership of SimSG (Section 3.2), hardness of TBSG (Section 3.3) and polynomial-time algorithms for pure NE in special cases (Section 3.4).

While pseudo-linearity is useful for several of our results, it appears to tie closely with NE in O-SimSGs[3]. To leverage the pseudo-linearity property more broadly for SimSGs, we make the following important structural observation of SimSGs, which implied that computing an approximate NE of general SimSG (TBSG) instances is polynomial-time reducible to computing an approximate NE of some corresponding O-SimSG (O-TBSG) instances.

[3] Monotonicity structure in stochastic games has been studied previously, and [45] claimed that a version of this structure holds for all TBSGs, including ones in which one player can control multiple states. However, it appears that the restriction to O-SimSG or (equivalently) considering the change in actions only at a single state is key and we prove in Appendix A of full version that without this, monotonicity may not hold.

▶ **Theorem 2** (Approximate-NE equivalences for SimSGs and O-SimSGs, restating Theorem 6 in full version)**.** *There exists a linear-time-computable mapping between the original* SimSG *and a linear-time-computable corresponding* O-SimSG *instance, such that for any $\epsilon \geq 0$ a strategy $\boldsymbol{\pi}$ is an ϵ-approximate mixed NE of the original* SimSG *if its induced policy $\boldsymbol{\pi}'$ is a $((1-\gamma)\epsilon/|\mathcal{S}|)$-approximate mixed NE in the corresponding* O-SimSG *(Definition 2 in full version).*

To prove Theorem 2, we leverage a key property of the induced single-player MDP for player i when the other players' policies $\boldsymbol{\pi}_{-i}$ are fixed: the policy improvement property of coordinate-wise (i.e. *asynchronous*) policy iteration [5, 57]. In general, Theorem 2 implies that an algorithm applicable to all O-SimSG instances can also be adapted to solve SimSG instances. This allows us to transfer the benefits of pseudo-linearity in the more specialized O-SimSG classes to all infinite-horizon SimSGs, despite the absence of monotonicity structure in the latter.

3.2 Complexity of NE in SimSGs

Here we describe how we leverage our structural results on infinite horizons SimSGs to show that computing NE of SimSGs is in PPAD and thereby obtain a full complexity characterization of such games (they are PPAD-complete). Our main complexity result for SimSGs is Theorem 3.

▶ **Theorem 3** (Complexity of NE in SimSG, restating Theorem 7 in full version)**.** *The problem of computing an ϵ-approximate NE for infinite-horizon* SimSG *class is* PPAD*-complete for a polynomially-bounded discount factor $\frac{1}{1-\gamma} = \text{poly}(A_{\text{tot}})$ and accuracy $\epsilon = \Omega(1/\text{poly}(A_{\text{tot}}))$.*

Showing hardness in Theorem 7 is relatively trivial: it follows immediately by considering $\gamma \to 0$ and noting that choosing the optimal stationary policy for one step involves computing a NE for an arbitrary multiplayer normal-form game, which is known to be PPAD-hard [16].

The more interesting component of the proof of Theorem 7 is the proof of PPAD membership. This proof is provided in Section 6.1 of full paper and leverages the foundational structure of SimSG discussed in Section 3.1 and additional properties of DMDPs. In particular, making use of the Brouwer fixed point argument (see, e.g. [14]) that shows the existence of NE, we construct two different types of Brouwer functions on strategies as below:

$$
\begin{aligned}
&f_{\mathsf{value}} : \boldsymbol{\pi} \to \mathbf{y} \\
&\text{such that} \quad y_{i,a}(\boldsymbol{\pi}) = \frac{\pi_i(a) + \max(u_i(\mathbf{e}_a, \boldsymbol{\pi}_{-i}) - u_i(\boldsymbol{\pi}), 0)}{1 + \sum_{a' \in \mathcal{A}_{i,s}} \max(u_i(\mathbf{e}_{a'}, \boldsymbol{\pi}_{-i}) - u_i(\boldsymbol{\pi}), 0)} \text{ for O-SimSG}, \qquad (8) \\
&f_{\mathsf{os-Bellman}} : \boldsymbol{\pi} \to \mathbf{y} \\
&\text{such that} \quad y_{i,s,a}(\boldsymbol{\pi}) = \frac{\pi_{i,s}(a) + \max([r_{i,s,a} + \gamma \mathbf{p}_{s,a}^\top \mathbf{V}_i^{\boldsymbol{\pi}}] - V_i^{\boldsymbol{\pi}}(s), 0)}{1 + \sum_{a' \in \mathcal{A}_{i,s}} \max([r_{i,s,a} + \gamma \mathbf{p}_{s,a}^\top \mathbf{V}_i^{\boldsymbol{\pi}}] - V_i^{\boldsymbol{\pi}}(s), 0)} \text{ for SimSG}\ .
\end{aligned}
$$

Both of these functions satisfy the property that $f(\boldsymbol{\pi}) = \boldsymbol{\pi}$ if and only if $\boldsymbol{\pi}$ is a NE, and are reminiscent of the Brouwer functions used in original PPAD-membership arguments that are tailored to linear utilities [16]. Each function leads to a different PPAD-membership proof and we include both due to the interesting distinct properties of SimSGs that they utilize.

Our proof based on f_{value} uses both the linear-time equivalence between O-SimSG and SimSG provided in Theorem 2, and the pseudo-linear structure of O-SimSG utilities in Theorem 1. The most non-trivial step involves showing that approximate Brouwer fixed points correspond to approximate NE (Lemma 5 of full paper), for which we critically use our

established property of pseudo-linearity. This proof has two key technical steps: showing the Brouwer function is L-Lipschitz-continuous utilizing properties of single-player DMDPs [37], and showing that the approximate fixed points of the Brouwer function $f(\cdot)$ correspond to approximate NE. We also show that this proof strategy generalizes to show PPAD-membership of any n-player-k-action game with pseudo-linear utilities (see Theorem 8 in full paper, under other mild conditions), which we think may be of independent interest.

Our alternative proof based on $f_{\mathsf{os-Bellman}}$ builds upon the structural fact that small Bellman errors suffice to argue about approximation of NE in Equation (18) of Appendix B in full version. Here the crucial observation is that fixing all other players' policies, the Bellman errors are linear in policy-space for a single player. As a consequence we can apply the more standard analysis [16] to argue that when $\boldsymbol{\pi}$ is an approximate fixed point of $f_{\mathsf{os-Bellman}}$, the Bellman update error $\max([r_{i,s,a} + \gamma \mathbf{p}_{s,a} V_i^{\boldsymbol{\pi}}] - V_i^{\boldsymbol{\pi}}(s)$ is close to 0. This in turn maps back to an approximate NE using the sufficient conditions on Bellman-error for NE (Appendix B in full paper).

Clarification and contextualization with recent prior work [20]: After initial drafting of this manuscript, we were pointed to the recent work of [20], which claims to have already shown the PPAD-membership of general SimSGs. However, we were unable to verify their proof; in particular, we do not know how to derive the 6-th line from the 5-th line in proving Case 2 of Lemma 4 in [20] (analogous to our Lemma 5 in the full version of our paper). Like us, the authors of [20] also use the Brouwer function f_{value} (more commonly known as Nash's Brouwer function and originally designed for linear utilities); however, unlike us, they do not establish or use any special pseudo-linear structure on the value functions. In our proof of Lemma 5 in full paper, this structure is key to establishing PPAD-membership and used for the most non-trivial part of the proof – that the approximate fixed points of the Brouwer function f_{value} are equivalent to approximate NE.

3.3 Complexity of NE in TBSGs

Here we consider the specialization of infinite-horizon SimSGs to TBSGs. Recall that in a TBSG, each state is controlled by only one player and, thus, players take turns in controlling the Markov process. We ask the fundamental question, *how hard is it to compute stationary NE in* TBSG*s*?

Unlike their non-turn-based counterparts, it is no longer clear that this problem is PPAD-hard: the aforementioned direct encoding of NE of arbitrary two-player normal-form games no longer applies when $|\mathcal{S}| = 1$ or $\gamma \to 0$. Moreover, in Sections 7.1 and 7.2 of full paper we show that approximate NE computation for TBSG is in polynomial-time if: (a) non-stationary NE are allowed, *or* (b) the number of states $|\mathcal{S}|$ is held to a constant; note that equilibrium computation for SimSGs remains PPAD-hard even under these simplifications.

Though prior work on general-sum TBSGs is limited, the special case of 2-player zero-sum TBSGs has been well studied [12, 61, 31, 62] and are known to possess additional structure beyond SimSGs. For example, [61] showed that a pure NE always exists for zero-sum TBSGs and [31] showed that NE is computable in strongly polynomial time when the discount factor is constant. However, this structure does not carry over to the general-sum case and [73] shows that there are TBSGs with only mixed NE (which hints at possible hardness). In Section 7.3 of full paper, we prove the following theorem and establish PPAD-hardness of computing NEs of TBSGs.

Table 2 Instantaneous rewards of player aux, out, informal: Constant γ is the given problem discount factor, $\gamma^L \leq \epsilon^2$ is tiny.

	$s_{\mathsf{in}}, a^1_{\mathsf{in}}$	$s_{\mathsf{in}}, a^2_{\mathsf{in}}$	$s_{\mathsf{aux}}, a^1_{\mathsf{aux}}$
aux	$1/2$	0	0
out	$-1/4$	$-1/4$	0

▶ **Theorem 4** (Complexity of TBSG NEs, restating Theorem 9 in full version, informal)**.** *Approximate NE-computation in infinite-horizon γ-discounted* TBSG*s with any $\gamma \in [1/2, 1)$ is* PPAD*-complete.*

We prove Theorem 9 by reducing the problem of generalized approximate circuit satisfiability (ϵ-GCircuit, formally defined in Definition 7 in full version) to O-TBSGs; ϵ-GCircuit is known to be PPAD-hard for even sufficiently small constant $\epsilon > 0$ [59]. This reduction is, at a high-level, the approach taken in the first proofs of PPAD-hardness of normal-form games [9, 16] as well as more recent literature (e.g. hardness for public goods games [52]); though it has been predominantly applied to games with linear or piecewise linear utilities. The key ingredients of our reduction are the implementation of certain circuit gates, i.e. $G_=$ (equal), G_α (set to constant α), $G_\times$ (multiply), G_+ (sum), G_- (subtraction), $G_>$ (comparison), $G_\wedge$ (logic AND), $G_\vee$ (logic OR), $G_\neg$ (logic NOT), through O-TBSG *game gadgets* which carefully encode these gates in an O-TBSG.

As an illustration, here we show how to implement an approximate equal gate $G_=$ between input and output players (corresponding to input and output states), i.e. $p_{\mathsf{out}} \in [p_{\mathsf{in}} - \epsilon, p_{\mathsf{in}} + \epsilon]$ at *any* approximate NE. This gadget includes 3 players (states): in (s_{in}), out (s_{out}) and aux (s_{aux}). Figure 1 illustrates the transitions in the TBSG instance and Table 2 partially specifies the instantaneous rewards. Here, the reader should think of p_{in} as the probability of player in choosing action a^1_{in} and p_{out} as the probability of player out choosing action a^1_{out}. Our game gadgets are crucially *multiplayer* in that they allow flexible choice of instant rewards for different players (e.g. in, aux and out).

We consider the case of exact NE as a warmup; in particular, we hope to show that exact NE necessitates $\pi_{\mathsf{in}}(a^1_{\mathsf{in}}) = \pi_{\mathsf{out}}(a^1_{\mathsf{out}})$. Just as in the typically implemented graphical game gadgets [16, 9], our hope is to enforce this equality constraint through a proof-by-contradiction argument that goes through two steps. As an illustration of the contradiction argument, suppose that $\pi_{\mathsf{in}}(a^1_{\mathsf{in}}) > \pi_{\mathsf{out}}(a^1_{\mathsf{out}})$. Our optimistic hope would be to choose the rewards and transitions so that the value function of player aux at his own state under choice of $\boldsymbol{\pi}_{\mathsf{out}}, \boldsymbol{\pi}_{\mathsf{in}}$ satisfies

$$V_{\mathsf{aux}}^{(\mathbf{e}_{a^1_{\mathsf{aux}}}, \boldsymbol{\pi}_{\mathsf{in}}, \boldsymbol{\pi}_{\mathsf{out}})} = \gamma \boldsymbol{\pi}_{\mathsf{in}}(a^1_{\mathsf{in}}) \quad \text{and} \quad V_{\mathsf{aux}}^{(\mathbf{e}_{a^2_{\mathsf{aux}}}, \boldsymbol{\pi}_{\mathsf{in}}, \boldsymbol{\pi}_{\mathsf{out}})} = \gamma \boldsymbol{\pi}_{\mathsf{out}}(a^1_{\mathsf{out}}). \tag{9}$$

If (9) were satisfied, aux player would have to take pure strategy a^1_{aux} at exact NE, which would transit to state s_{in}. As reflected in the reward table (Table 2) this would be a bad event for player out due to the negative reward she accrues at state s_{in}. Consequently, she would prefer to take action a^1_{out} as much as possible, i.e. $\pi_{\mathsf{out}}(a^1_{\mathsf{out}}) \approx 1$, which would lead to the desired contradiction. (A symmetric contradictory argument would work for the case $\pi_{\mathsf{in}}(a^1_{\mathsf{in}}) < \pi_{\mathsf{out}}(a^1_{\mathsf{out}})$, ensuring that the system balances and necessitates $\pi_{\mathsf{in}} = \pi_{\mathsf{out}}$ at an exact NE.)

However, creating an equal gadget through O-TBSG is much more intricate than a corresponding graphical game gadget due to the twin challenges of *nonlinearity* and *common structure* in players' utilities. For one, the pseudo-linear structure described in Section 3.1 only ensures approximate linearity up to multiplicative constants; the more fine-grained equality

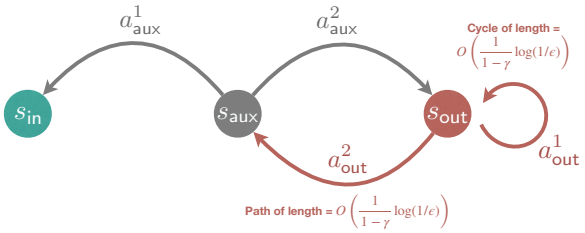

Figure 1 Illustration of states and transitions for "equal gadget" to implement $G_=$. The transitions in red encode a cycle or path of length L, where $L = \lceil \frac{4}{1-\gamma}\log(1/\epsilon) \rceil$ for constants γ, ϵ.

required in (9) is far more difficult to achieve (and unclear whether possible). Moreover, unlike the definitional local structure between players in graphical games [40], TBSGs have significant *global* structure between players (as players represent states that transit to one another). In other words the players' utility functions depend on *all* of the other players and not just their immediate neighbors. As a consequence of this global structure, a naive combination of individual gadgets could sizably change the value functions and break the local circuit operations.

We work around these two issues by creating long cycles and paths with "dummy states" for the actions the out player takes such that the only non-zero rewards are collected outside these dummy states. We show that this elongation of paths simultaneously induces *approximate linearity* and *localization to neighbors* in the O-TBSG instance. When the path has length $L = O(\frac{1}{1-\gamma}\log(\frac{1}{\epsilon}))$, we satisfy (9) in an approximate sense up to tiny poly(ϵ) errors. Further, this almost-linear structure turns out to be robust to transitions that are "further away" from s_{out}. This ensures that the out player can then be used as an input for subsequent gadgets connected in series, and enables a successful combination of the gadgets without changing the NE conditions at each state.

It remains to translate these ideas from an exact-NE argument to an approximate-NE argument. For this, the pseudo-linearity property that we established in Section 5.1 of full version proves to be especially useful. In particular, when $\boldsymbol{\pi}_{\text{in}}(a^1_{\text{in}}) > \boldsymbol{\pi}_{\text{out}}(a^1_{\text{out}}) + \epsilon$, we can adapt the bounded "slope" argument in (6) and observe that

$$\begin{aligned} V_{\text{aux}}^{(\mathbf{e}_{a^1_{\text{aux}}}, \boldsymbol{\pi}_{\text{in}}, \boldsymbol{\pi}_{\text{out}})} - V_{\text{aux}}^{(\theta\mathbf{e}_{a^1_{\text{aux}}} + (1-\theta)\mathbf{e}_{a^2_{\text{aux}}}, \boldsymbol{\pi}_{\text{in}}, \boldsymbol{\pi}_{\text{out}})} &\geq (1-\theta)(1-\gamma)\left[V_{\text{aux}}^{(\mathbf{e}_{a^1_{\text{aux}}}, \boldsymbol{\pi}_{\text{in}}, \boldsymbol{\pi}_{\text{out}})} - V_{\text{aux}}^{(\mathbf{e}_{a^2_{\text{aux}}}, \boldsymbol{\pi}_{\text{in}}, \boldsymbol{\pi}_{\text{out}})}\right] \\ &\geq (1-\theta)\gamma(1-\gamma)\epsilon, \end{aligned}$$

which ensures that θ, i.e. the probability that player aux takes action a^1_{aux}, must be close enough to 1 at any approximate NE. We use this pseudo-linearity multiple times to formally relax the exact NE argument under approximation, in order to implement $G_=$ gate for ϵ-GCircuit.

Ultimately, our proof of this theorem sheds further light on the problem's structure and shows that hardness is fairly resilient in general-sum stochastic games. Even in the special case where each player controls a single state and receives non-zero reward at at most 4 states (*or* alternatively, all players have non-negative but dense reward structure), the problem is still PPAD-hard.

Contextualization with independent concurrent work. In independent and concurrent work, the authors of [17] were additionally able to prove that the computation of NE of even 2-player TBSGs is PPAD-hard. We note that beyond claims of hardness in TBSGs, each of [17] and this work contain disjoint results of independent interest. For instance, [17] provides a polynomial-time algorithm for finding *non-stationary* Markov CCEs for SimSGs. On the other hand, this work focuses exclusively on stationary equilibrium concepts. In addition to hardness of TBSGs, we show the PPAD-membership of general games with pseudo-linear utilities including SimSGs (see Section 3.2) and provide polynomial-time algorithms for finding stationary NEs for TBSGs under extra assumptions on the reward structure (see Section 3.4).

3.4 Efficient algorithms for TBSGs under localized rewards

As shown above, the problem of finding an approximate NE for infinite-horizon TBSGs is PPAD-complete even under a variety of additional structural assumptions. For example we show that even when each player only controls one state, all transitions are deterministic, and all players receive non-negative rewards (possibly in many states) computing a NE in a TBSG is PPAD-hard.

Towards characterizing what features are critical to the hardness of the problem, we specialize further and ask what happens if we further restrict each player to receive reward only at the *single state* that they control. We call this class of games LocReward and consider the class of *fixed-sign* LocReward O-TBSG, i.e. $r_{i,s_i,\cdot} \geq 0$ (or $r_{i,s_i,\cdot} \leq 0$) for all $i \in [n]$ and $r_{i,s',\cdot} = 0$ for any $s' \neq s_i$.

The intuitive reason for why this special structure is helpful is that it creates a qualitative symmetry in the players' incentives: all of them wish to either reach (in the case of non-negative rewards) or avoid (in the case of negative rewards) their own controlling state. Mathematically, we observe that given a strategy $\boldsymbol{\pi}$, the utility function has the following structure

$$u_i(\boldsymbol{\pi}) = V_i^{\boldsymbol{\pi}} = \mathbf{e}_{s_i}^\top (\mathbf{I} - \gamma \mathbf{P}^{\boldsymbol{\pi}})^{-1} \mathbf{r}^{\boldsymbol{\pi}} = \frac{E_i(\boldsymbol{\pi}_{-i})}{\det(\boldsymbol{\pi})} r_i^{\boldsymbol{\pi}_i}(s_i) \text{ where } \det(\boldsymbol{\pi}) := \det(\mathbf{I} - \gamma \mathbf{P}^{\boldsymbol{\pi}}) > 0 \tag{10}$$

and $E_i(\boldsymbol{\pi}_{-i})$ is the determinant of the $(i,i)^{th}$ minor of the matrix $\mathbf{I} - \gamma \mathbf{P}^{\boldsymbol{\pi}}$. The last equality for $u_i(\boldsymbol{\pi})$ in (10) used the matrix inversion formula and the fact that $r_i^{\boldsymbol{\pi}_j}(s_j) = 0$ for any $j \neq i$. Consequently, the numerator of $u_i(\boldsymbol{\pi})$ is separable in $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_{-i}$ and the denominator is common to all players $i \in [n]$. This implies that, following a logarithmic transformation, a fixed-sign LocReward O-TBSG game is equivalent to a *potential game* [49], with potential function $\Phi(\boldsymbol{\pi}) := \log(\det(\boldsymbol{\pi})^{-1} \prod_{i \in [n]} r_i^{\boldsymbol{\pi}_i}(s_i))$. That is, for any $i \in [n]$ and $\boldsymbol{\pi}_i$, $\boldsymbol{\pi}_i'$ we have $\Phi(\boldsymbol{\pi}_i', \boldsymbol{\pi}_{-i}) - \Phi(\boldsymbol{\pi}) = \log V_i^{(\boldsymbol{\pi}_i', \boldsymbol{\pi}_{-i})} - \log V_i^{(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i})}$.

The potential game structure of LocReward O-TBSG automatically implies the existence of a pure NE for all fixed-sign LocReward games. Further, it follows [49] that (approximate) best response dynamics (also known as *strategy iteration* in the stochastic games literature [31]) provably decrease this potential by a polynomial factor, until it achieves a pure-strategy approximate NE. This yields a polynomial-time algorithm for computing approximate NE as stated below.

▶ **Theorem 5** (Restating Lemmas 11 and 12 and Propositions 3 and 4 in full version)**.** *Consider a* LocReward O-TBSG *instance* $\mathcal{G} = (n, \mathcal{S} = \cup_{i \in [n]} \{s_i\}, \mathcal{A}, \mathbf{p}, \mathbf{r}, \gamma)$ *where all rewards are non-negative (or non-positive). Then the game has a pure NE, and given some accuracy* ϵ*, approximate best-response dynamics find an* ϵ*-approximate pure NE in time* $\mathrm{poly}(A_{\mathsf{tot}}, \frac{1}{1-\gamma}, \frac{1}{\epsilon})$.

Table 3 Summary of complexity characterization for O-TBSG under various reward assumptions.

Setting (O-TBSG)	Localized rewards (LocReward)	General rewards
Fixed-sign rewards	Polynomial (pure NE)	PPAD-complete
Mixed-sign rewards	NP-hard (pure NE), open problem (mixed NE)	PPAD-complete

We also show that under further assumptions it is possible to compute an exact NE through a different set of algorithms inspired by graph problems. Specifically, we consider a special sub-class of fixed-sign LocReward O-TBSG where we impose two additional structural assumptions: (a) all transitions are deterministic, and (b) all rewards on each player's own state are independent of actions. Under these refinements, all players in an non-negative LocReward instance are incentivized to go through a shortest-cycle to maximize its utility, while all players in an non-positive LocReward instance are incentivized to go through a cycle that is as long as possible (or, most ideally follow a path to a cycle that doesn't return to the player's controlled state). Accordingly, we design graph algorithms (Algorithm 5 and Algorithm 7 in full version) that locally, iteratively find the cycle and path structure that corresponds to an exact NE. Our results show that best-response dynamics (i.e. strategy iteration) and graph-based algorithms can work in general-sum TBSGs beyond zero-sum setting [31].

Finally, note that our positive results really require *both* of the assumptions of (a) reward only at a single state (b) rewards of the same sign (see Table 3 for a summary). From Section 3.3 we already know when relaxing the first condition, finding an approximate mixed NE is PPAD-hard. The second condition is also important, as relaxing it (i.e. allowing both positive and negative rewards in the LocReward O-TBSG model) may preclude even the existence of pure NE [73]. In fact, we show in Section 8.2 of full version that even determining whether or not a pure NE exists is NP-hard via a reduction to the Hamiltonian path problem (but whether *mixed* NE are polynomial-time computable under this modification remains open). Ultimately, this gives a more complete picture of what transformations change the problem from being PPAD-complete to being polynomial time solvable.

4 Related work

Here we highlight prior work that is most closely related to our results.

General-sum stochastic game theory. Central questions in stochastic game theory research involve (a) the existence of equilibria and (b) the convergence and complexity of algorithms that compute these equilibria. Existence of equilibria is known in significantly more general formulations of stochastic games than the tabular SimSGs that are studied in our paper (see, e.g. the classic textbooks [24, 4]). Relevant to our study, the first existence proofs of general-sum tabular stochastic games appeared in [27, 67]. They are based on Kakutani's fixed point theorem, and so *non-constructive* in that they do not immediately yield an algorithm. This is a departure from the zero-sum case, where Shapley's proof of existence [61] is constructive and directly leverages the convergence of infinite-horizon dynamic-programming.

Indeed, the recent survey paper on multi-agent RL [72] mentions the search for computationally tractable and provably convergent (to NE) algorithms for SimSG as an open problem. Algorithms that are known to converge to NE in SimSG require strong assumptions on the heterogeneous rewards – such as requiring the one-step equilibrium to be unique at each iteration [33, 29], or requiring the players to satisfy a "friend-or-foe" relationship [43].

An important negative result in the literature was the shown failure of convergence of infinite-horizon dynamic-programming algorithms for general-sum SimSGs [73]. More generally, they uncover a fundamental identifiability issue by showing that more than one equilibrium value (and, thereby, more than one NE) can realize identical action-value functions. This identifiability issue suggests that any iterative algorithm that uses action-value functions in its update (including policy-based methods like policy iteration and two-timescale actor-critic [41]) will fail to converge for similar reasons.

Since then, alternative algorithms that successfully *asymptotically* converge to NE have been developed for general-sum SimSGs based on two-timescale approaches [56] and homotopy methods [7, 32]. However, these algorithms are intricately coupled across players and states in a more intricate way and, at the very least, suffer a high complexity per iteration. Finite-time guarantees for these algorithms do not exist in the literature. A distinct approach that uses linear programming is also proposed [21], but this algorithm also suffers from exponential iteration complexity. Algorithms that are used for general-sum SimSG in practice are largely heuristic and directly minimize the Bellman error of the strategy [55] (which we defined in Appendix B of full version).

Interestingly, this picture does not significantly change for TBSGs despite their significant structure over and above SimSGs. The counterexamples of [73] are in fact 2-player, 2-state, and 2-action-per-state TBSGs. Our PPAD-hardness results for TBSG resolve an open question that was posed by [73], who asked whether alternative methods (using Q-values and equilibrium-value functions) could be used to derive stationary NE in TBSG instead. In particular, we show that the stationary NE is not only difficult to approach via popular dynamics, but is fundamentally hard.

Very recently, a number of positive results for finite-horizon non-stationary CCE in SimSGs were provided [65, 34, 46]. These results even allow for independent learning by players. A natural question is whether an infinite-horizon *stationary* CCE could be extracted from these results. Since TBSG NE is a special case of SimSG CCE, our PPAD-hardness result answers this question in the negative. In general, tools that are designed for computing and approaching non-stationary equilibria cannot be easily leveraged to compute or approach stationary equilibria due to the induced nonconvexity in utilities and the failure of infinite-horizon dynamic programming. Our paper fills this gap and provides a comprehensive characterization of complexity of computing stationary NE for infinite-horizon multi-player SimSGs and TBSGs.

A trivial observation is that the problem of exact computation for general-sum stochastic games is only harder than approximation; in general, exact computation for NE of stochastic games is outside the scope of this paper and we refer readers to [26] for recent hardness result following that thread.

The zero-sum case. There is a substantial literature on equilibrium computation, sample complexity and learning dynamics in the case of zero-sum SimSG and TBSG. For a detailed overview of advances in learning in zero-sum stochastic games, see the survey paper [72]. In contrast, our results address the general-sum case. Positive results for zero-sum TBSG, such as the property of strongly-polynomial-time computation of an exact NE with a constant discount factor [31], leverage special structure that does not carry over to the general-sum case. In particular, a pure NE always exists for a zero-sum TBSG owing to the convergence of Shapley's value iteration [61]. [73] showed that a pure NE need not exist for general-sum TBSGs. We further show in Section 8 of full version that pure NE are NP-hard to compute (at least in part due to their possible lack of existence). On the more positive side, we also characterize specializations of general-sum TBSGs for which pure NE always exist and are polynomial-time computable.

It is crucial to note that our results only address the equilibrium computation problem of general-sum SimSGs and TBSGs with a constant discount factor. When the rewards are zero-sum this is known to be polynomial-time [61] and additionally strongly polynomial-time in the case of TBSG [31]. Whether it is possible to compute an (exact or approximate) NE in even zero-sum TBSGs with an increasing discount factor remains open [2]. This open problem has important connections to simple stochastic games [12, 23], mean-payoff games [30, 74], and parity games [22, 69, 36].

Algorithmic game theory for normal-form and market equilibria. The PPAD complexity class was introduced by [53] to capture the complexity of all total search problems (i.e. problems for which a solution is known) [47] that are polynomial-time reducible to the problem of finding at least one unbalanced vertex on a directed graph. [16] first showed that NE computation for n-player k-action normal-form games lies in PPAD. By definition, the utilities of normal-form games are always *linear* in the mixed strategies. This is not the case for SimSG or TBSG, whose utilities are not even convex in their argument. The membership of nonconvex general-sum games in PPAD is not obvious. For example, [18] recently showed PPAD-hardness of even *zero-sum* constrained nonconvex-nonconcave games. Moreover, the complexity of all general-sum games satisfying a succinct representation and the property of polynomial-time evaluation of expected utility (which includes SimSG) is believed to lie in a strictly harder complexity class than PPAD [60]. General-sum nonlinear game classes that are known to be in PPAD primarily involve market equilibrium [8, 68, 11, 28] and Bayes-NE of auctions [25] and make distinct assumptions of either a) *separable concave and piecewise linear (SPLC)* assumptions on the utilities or b) *constant-elasticity-of-substitution (CES) utilities* [11]. They also utilize in part linearity in sufficient conditions for NE (e.g. Walras's law for market equilibrium). These structures, while interesting in their own right, are also not satisfied by SimSGs or TBSGs. The pseudo-linear property of SimSGs that we uncover in Section 5.1 of full version is key to showing PPAD-membership. Our subsequent proof in Section 6.1 in full version is a useful generalization of the traditional proof for linear utilities [16] to pseudo-linear utilities.

In addition to being in PPAD, general-sum normal-form games were established to be PPAD-hard by [9, 16]. Since then, PPAD-hardness has been shown for several structured classes of normal-form games [48, 44, 10, 19, 52] as well as for weaker objectives in normal-form games such as constant-additive approximation [15, 58, 59] and smoothed-analysis [6]. Our approach to prove PPAD-hardness for TBSG takes inspiration from the approach to prove PPAD-hardness for n-player graphical games [39, 40] (which was subsequently used to prove PPAD-hardness for constant-player normal-form games by [16]). In particular we construct *game gadgets* to implement real-valued arithmetic circuit operations through TBSG NE. As summarized in Section 3.3, the details of our TBSG game gadgets are significantly more intricate than the corresponding graphical game gadgets due to the additional challenges of global shared structure across players and the nonlinearity of the utilities. These challenges do not manifest in graphical games as, by definition, they only possess local structure and satisfy linearity in utilities. Whether TBSGs are directly reducible to graphical games or bimatrix games remains an intriguing open question.

Relation of TBSG to other game-theoretic paradigms. We conclude our overview of related work with a brief summarization of solution concepts and paradigms that are partially related to TBSG's. First, the class of sequential or *extensive-form games* is known to lie in PPAD [71] and is trivially PPAD-hard due to normal-form games being a special case. We

note that computation of non-stationary equilibria in the finite-horizon SimSG and TBSG are special cases of these. Second, the solution concept of (coarse) correlated equilibrium (CCE) is polynomial-time computable, in contrast with NE, even for multiplayer games with linear utilities [54]. Since TBSG involves a non-trivial action set for only one player at each state, the solution concepts of NE and CCE all become equivalent for both stationary and non-stationary equilibria. On the positive side, this may imply the convergence of recently designed finite-horizon learning dynamics [65, 34, 46] to TBSG NE. On the negative side, our PPAD-hardness of approximation of stationary NE in TBSG (Section 7.3 of full version) implies hardness of *stationary* CCE equilibria in SimSGs. Finally, we contextualize our NP-hardness results on certain decision problems (i.e. *does there exist an equilibrium with certain properties?*) in Section 8.2 of full version. In normal-form games, such decision problems are known to be NP-hard [13].

References

1 Eitan Altman. Flow control using the theory of zero sum markov games. *IEEE transactions on automatic control*, 39(4):814–818, 1994.

2 Daniel Andersson and Peter Bro Miltersen. The complexity of solving stochastic games on graphs. In *International Symposium on Algorithms and Computation*, pages 112–121. Springer, 2009.

3 Yu Bai, Chi Jin, and Tiancheng Yu. Near-optimal reinforcement learning with self-play. *Advances in neural information processing systems*, 33:2159–2170, 2020.

4 Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

5 Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1(2). Athena scientific Belmont, MA, 1995.

6 Shant Boodaghians, Joshua Brakensiek, Samuel B Hopkins, and Aviad Rubinstein. Smoothed complexity of 2-player nash equilibria. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 271–282. IEEE, 2020.

7 Ron N Borkovsky, Ulrich Doraszelski, and Yaroslav Kryukov. A user's guide to solving dynamic stochastic games using the homotopy method. *Operations Research*, 58(4-part-2):1116–1132, 2010.

8 Xi Chen, Decheng Dai, Ye Du, and Shang-Hua Teng. Settling the complexity of arrow-debreu equilibria in markets with additively separable utilities. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 273–282. IEEE, 2009.

9 Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272. IEEE, 2006.

10 Xi Chen, David Durfee, and Anthi Orfanou. On the complexity of nash equilibria in anonymous games. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 381–390, 2015.

11 Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. The complexity of non-monotone markets. *Journal of the ACM (JACM)*, 64(3):1–56, 2017.

12 Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992.

13 Vincent Conitzer and Tuomas Sandholm. Complexity results about nash equilibria. *arXiv preprint*, 2002. `arXiv:cs/0205074`.

14 Partha Dasgupta and Eric Maskin. The existence of equilibrium in discontinuous economic games, i: Theory. *The Review of economic studies*, 53(1):1–26, 1986.

15 Constantinos Daskalakis. On the complexity of approximating a nash equilibrium. *ACM Transactions on Algorithms (TALG)*, 9(3):1–35, 2013.

16 Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

17 Constantinos Daskalakis, Noah Golowich, and Kaiqing Zhang. The complexity of markov equilibrium in stochastic games. *arXiv preprint*, 2022. arXiv:2204.03991.

18 Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1466–1478, 2021.

19 Argyrios Deligkas, John Fearnley, and Rahul Savani. Tree polymatrix games are ppad-hard. *arXiv preprint*, 2020. arXiv:2002.12119.

20 Xiaotie Deng, Yuhao Li, David Henry Mguni, Jun Wang, and Yaodong Yang. On the complexity of computing markov perfect equilibrium in general-sum stochastic games. *arXiv preprint*, 2021. arXiv:2109.01795.

21 Liam Dermed and Charles Isbell. Solving stochastic games. *Advances in Neural Information Processing Systems*, 22, 2009.

22 E Allen Emerson and Charanjit S Jutla. Tree automata, mu-calculus and determinacy. In *FoCS*, volume 91, pages 368–377. Citeseer, 1991.

23 Kousha Etessami and Mihalis Yannakakis. On the complexity of nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.

24 Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

25 Aris Filos-Ratsikas, Yiannis Giannakopoulos, Alexandros Hollender, Philip Lazos, and Diogo Poças. On the complexity of equilibrium computation in first-price auctions. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 454–476, 2021.

26 Aris Filos-Ratsikas, Kristoffer Arnsfelt Hansen, Kasper Høgh, and Alexandros Hollender. Fixp-membership via convex optimization: Games, cakes, and markets. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 827–838. IEEE, 2022.

27 Arlington M Fink. Equilibrium in a stochastic n-person game. *Journal of science of the hiroshima university, series ai (mathematics)*, 28(1):89–93, 1964.

28 Jugal Garg, Ruta Mehta, Vijay V Vazirani, and Sadra Yazdanbod. Settling the complexity of leontief and plc exchange markets under exact and approximate equilibria. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 890–901, 2017.

29 Amy Greenwald, Keith Hall, Roberto Serrano, et al. Correlated q-learning. In *ICML*, volume 3, pages 242–249, 2003.

30 Vladimir A Gurvich, Alexander V Karzanov, and LG Khachivan. Cyclic games and an algorithm to find minimax cycle means in directed graphs. *USSR Computational Mathematics and Mathematical Physics*, 28(5):85–91, 1988.

31 Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1–16, 2013.

32 P Jean-Jacques Herings and Ronald JAP Peeters. Stationary equilibria in stochastic games: structure, selection, and computation. *Journal of Economic Theory*, 118:32–60, 2004.

33 Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

34 Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent rl. *arXiv preprint*, 2021. arXiv:2110.14555.

35 Yujia Jin and Aaron Sidford. Towards tight bounds on the sample complexity of average-reward mdps. In *International Conference on Machine Learning*, pages 5055–5064. PMLR, 2021.

36 Marcin Jurdziński, Mike Paterson, and Uri Zwick. A deterministic subexponential algorithm for solving parity games. *SIAM Journal on Computing*, 38(4):1519–1532, 2008.

37 Sham Machandranath Kakade et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

38 Ioannis Karatzas, Martin Shubik, and William D Sudderth. A strategic market game with secured lending. *Journal of mathematical economics*, 28(2):207–247, 1997.

39 Michael Kearns. Graphical games. *Algorithmic game theory*, 3:159–180, 2007.

40 Michael Kearns, Michael L Littman, and Satinder Singh. Graphical models for game theory. *arXiv preprint*, 2013. `arXiv:1301.2281`.

41 Vijaymohan R Konda and Vivek S Borkar. Actor-critic–type learning algorithms for markov decision processes. *SIAM Journal on control and Optimization*, 38(1):94–123, 1999.

42 David Levhari and Leonard J Mirman. The great fish war: an example using a dynamic cournot-nash solution. *The Bell Journal of Economics*, pages 322–334, 1980.

43 Michael L Littman et al. Friend-or-foe q-learning in general-sum games. In *ICML*, volume 1, pages 322–328, 2001.

44 Zhengyang Liu and Ying Sheng. On the approximation of nash equilibria in sparse win-lose games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32(1), 2018.

45 Dmitrii Lozovanu. Stationary nash equilibria for average stochastic positional games. In *Frontiers of Dynamic Games*, pages 139–163. Springer, 2018.

46 Weichao Mao and Tamer Başar. Provably efficient reinforcement learning in decentralized general-sum markov games. *Dynamic Games and Applications*, pages 1–22, 2022.

47 Nimrod Megiddo and Christos H Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.

48 Ruta Mehta. Constant rank bimatrix games are ppad-hard. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 545–554, 2014.

49 Dov Monderer and Lloyd S Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

50 Roger B Myerson. *Game theory: analysis of conflict*. Harvard university press, 1997.

51 John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

52 Christos Papadimitriou and Binghui Peng. Public goods games in directed networks. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 745–762, 2021.

53 Christos H Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and system Sciences*, 48(3):498–532, 1994.

54 Christos H Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM (JACM)*, 55(3):1–29, 2008.

55 Julien Pérolat, Florian Strub, Bilal Piot, and Olivier Pietquin. Learning nash equilibrium for general-sum markov games from batch data. In *Artificial Intelligence and Statistics*, pages 232–241. PMLR, 2017.

56 HL Prasad, Prashanth LA, and Shalabh Bhatnagar. Two-timescale algorithms for learning nash equilibria in general-sum stochastic games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1371–1379, 2015.

57 Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

58 Aviad Rubinstein. Settling the complexity of computing approximate two-player nash equilibria. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 258–265. IEEE, 2016.

59 Aviad Rubinstein. Inapproximability of nash equilibrium. *SIAM Journal on Computing*, 47(3):917–959, 2018.

60 Grant R Schoenebeck and Salil Vadhan. The computational complexity of nash equilibria in concisely represented games. *ACM Transactions on Computation Theory (TOCT)*, 4(2):1–50, 2012.

61 Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

62 Aaron Sidford, Mengdi Wang, Lin Yang, and Yinyu Ye. Solving discounted stochastic two-player games with near-optimal time and sample complexity. In *International Conference on Artificial Intelligence and Statistics*, pages 2992–3002. PMLR, 2020.

63 David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

64 David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

65 Ziang Song, Song Mei, and Yu Bai. When can we learn general-sum markov games with a large number of players sample-efficiently? *arXiv preprint*, 2021. `arXiv:2110.04184`.

66 Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

67 Masayuki Takahashi. Equilibrium points of stochastic non-cooperative n-person games. *Journal of Science of the Hiroshima University, Series AI (Mathematics)*, 28(1):95–99, 1964.

68 Vijay V Vazirani and Mihalis Yannakakis. Market equilibrium under separable, piecewise-linear, concave utilities. *Journal of the ACM (JACM)*, 58(3):1–25, 2011.

69 Jens Vöge and Marcin Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *International conference on computer aided verification*, pages 202–215. Springer, 2000.

70 Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.

71 Peyton Young and Shmuel Zamir. *Handbook of game theory*. Elsevier, 2014.

72 Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021.

73 Martin Zinkevich, Amy Greenwald, and Michael Littman. Cyclic equilibria in markov games. *Advances in Neural Information Processing Systems*, 18:1641, 2006.

74 Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1-2):343–359, 1996.