Theoretical Computer Science 962 (2023) 113931

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Lower bounds for the query complexity of equilibria in Lipschitz games ☆

Paul W. Goldberg, Matthew J. Katzman *

Department of Computer Science, University of Oxford, United Kingdom

ARTICLE INFO

Article history:
Received 3 January 2022
Received in revised form 23 November 2022
Accepted 28 April 2023
Available online 23 May 2023

Keywords:
Query complexity
Lipschitz games
Nash equilibrium

ABSTRACT

Nearly a decade ago, Azrieli and Shmaya introduced the class of λ-Lipschitz games in which every player's payoff function is λ-Lipschitz with respect to the actions of the other players. They showed that such games admit ϵ-approximate pure Nash equilibria for certain settings of ϵ and λ. They left open, however, the question of how hard it is to find such an equilibrium. In this work, we develop a query-efficient reduction from more general games to Lipschitz games. We use this reduction to show a query lower bound for any randomized algorithm finding ϵ-approximate *pure* Nash equilibria of n-player, binary-action, λ-Lipschitz games that is exponential in $n\lambda/\epsilon$. In addition, we introduce "Multi-Lipschitz games," a generalization involving player-specific Lipschitz values, and provide a reduction from finding equilibria of these games to finding equilibria of Lipschitz games, showing that the value of interest is the *average* of the individual Lipschitz parameters. Finally, we provide an exponential lower bound on the *deterministic* query complexity of finding ϵ-approximate Nash equilibria of n-player, m-action, λ-Lipschitz games for strong values of ϵ, motivating the consideration of explicitly randomized algorithms in the above results.

1. Introduction

A Lipschitz game is a multi-player game in which there is an additive limit λ (called the Lipschitz constant of the game) on how much any player's payoffs can change due to a deviation by any other player. Thus, every player's payoff function is λ-Lipschitz continuous as a function of the other players' mixed strategies. Lipschitz games were introduced about ten years ago by Azrieli and Shmaya [1]. A key feature of Lipschitz games is that they are guaranteed to have approximate Nash equilibria *in pure strategies*, where the quality of the approximation depends on the number of players n, the number of actions m, and the Lipschitz constant λ. In particular, [1] showed that this guarantee holds (keeping the number of actions constant) for Lipschitz constants of size $o(\epsilon/\sqrt{n\log n})$ (existence of approximate pure equilibria is trivial for Lipschitz constants of size $o(\epsilon/n)$, where players have such low effect on each others' payoffs that they can best-respond independently to get an approximate pure equilibrium). The general idea of the existence proof is to take a mixed Nash equilibrium (guaranteed to exist by Nash's theorem [16]), and prove that there is a positive probability that a pure profile sampled from it will constitute an approximate equilibrium.

☆ This article belongs to Section A: Algorithms, automata, complexity and games, Edited by Paul Spirakis.

* Corresponding author.
E-mail addresses: paul.goldberg@cs.ox.ac.uk (P.W. Goldberg), matthewkatzman26@gmail.com (M.J. Katzman).

https://doi.org/10.1016/j.tcs.2023.113931

As noted in [1], solutions in pure-strategy profiles can be considered a more plausible and satisfying model of a game's outcome than solutions in mixed-strategy profiles. On the other hand, the existence guarantee raises the question of how to *compute* an approximate equilibrium. In contrast with potential games, in which pure-strategy equilibria can often be found via best- and better-response dynamics, there is no obvious natural approach in the context of Lipschitz games, despite the existence guarantee. The general algorithmic question (of interest in the present paper) is:

Given a Lipschitz game, how hard is it to find a pure-strategy profile that constitutes an approximate equilibrium?

Recent work [8,10] has identified algorithms achieving additive constant approximation guarantees, but as noted by Babichenko [5], the extent to which we can achieve the approximate pure equilibria that are guaranteed by [1] (or alternatively, potential lower bounds on query or computational complexity) is unknown.

Variants and special cases of this question include classes of Lipschitz games having a concise representation, as opposed to unrestricted Lipschitz games for which an algorithm has query access to the payoff function (as we consider in this paper). In the latter case, the question subdivides into what we can say about the query complexity, and about the computational complexity (for concisely-represented games the query complexity is low, by Theorem 3.3 of [11]). Moreover, if equilibria can be easily computed, does that remain the case if we ask for this to be achievable via some kind of natural-looking decentralized process? Positive results for these questions help us to believe in "approximate pure Nash equilibrium" as a solution concept for Lipschitz games. Alternatively, it is of interest to identify computational obstacles to the search for a Nash equilibrium.

1.1. Prior work

In this paper we apply various important lower bounds on the query complexity of computing approximate Nash equilibria of *unrestricted* n-player games, a model of computation in which the input is withheld from the algorithm by an oracle and the measure of complexity is the number of queries the algorithm needs to make. Query complexity can be 'deterministic' or 'randomized', depending on whether the algorithm making the queries is itself deterministic or randomized. In general, lower bounds on the query complexity are known that are exponential in n (which motivates a focus on subclasses of games, such as Lipchitz games, and others). Hart and Mansour [13] showed that the deterministic communication (and thus query) complexity of computing an exact Nash equilibrium (pure or mixed) in a game with n players is $2^{\Omega(n)}$. Subsequent results have iteratively strengthened this lower bound. First, Babichenko [3] showed an exponential lower bound on the *randomized* query complexity of computing an ϵ-*well supported* Nash equilibrium (an approximate equilibrium in which every action in the support of a given player's mixed strategy is an ϵ-best response) for a constant value of ϵ, even when considering δ-distributional query complexity, as defined in Definition 5. Shortly after, Chen, Cheng, and Tang [6] showed a $2^{\Omega(n/\log n)}$ lower bound on the randomized query complexity of computing an (not necessarily well-supported) ϵ-approximate Nash equilibrium for a constant value of ϵ, which Rubinstein [19] improved to a $2^{\Omega(n)}$ lower bound, even allowing a constant fraction of players to experience regret greater than ϵ (taking regret as defined in Definition 3). These intractability results motivate us to consider a restricted class of games (Lipschitz games) which contain significantly more structure than do general games.

Lipschitz games were initially considered by Azrieli and Shmaya [1], who showed that any λ-Lipschitz game (as defined in Section 2.1) with n players and m actions admits an ϵ-approximate *pure* Nash equilibrium for any $\epsilon \geq \lambda\sqrt{8n\log 2mn}$.[1] In Section 3 we provide a lower bound on the query complexity of finding such an equilibrium.

Positive algorithmic results have been found for classes of games that combine the Lipschitz property with others, such as *anonymous* games [7] and *aggregative* games [4]. For anonymous games (in which each player's payoffs depend only on the *number* of other players playing each action, and not *which* players), Daskalakis and Papadimitriou [7] improved upon the upper bound of [1] to guarantee the existence of ϵ-approximate pure Nash equilibria for $\epsilon = \Omega(\lambda)$ (the only dependence on n coming from λ itself), and presented a polynomial-time algorithm to find one. Peretz et al. [17] analyze Lipschitz values that result from perturbing anonymous games to bring the payoffs closer to their averages. Goldberg and Turchetta [12] showed that a 3λ-approximate pure Nash equilibrium of a λ-Lipschitz anonymous game can be found querying $O(n\log n)$ individual payoffs.

Goldberg et al. [10] showed a logarithmic upper bound on the randomized query complexity of computing $\frac{1}{8}$-approximate Nash equilibria in binary-action $\frac{1}{n}$-Lipschitz games. They also presented a randomized algorithm finding a $(\frac{3}{4}+\alpha)$-approximate Nash equilibrium when the number of actions is unbounded.

1.2. Our contributions

The primary contribution of this work is the development and application of a query-efficient version of a reduction technique used in [1,2] in which an algorithm finds an equilibrium in one game by reducing it to a population game (in

[1] General Lipschitz games cannot be written down concisely, so we assume black-box access to the payoff function of a Lipschitz game. This emphasizes the importance of considering *query complexity* in this context. Note that a pure approximate equilibrium can still be *checked* using mn queries.

which entities play actions based by aggregating the votes of a constituent population) with a smaller Lipschitz parameter. As the former is a known hard problem, we prove hardness for the latter.

In Section 2 we introduce notation and relevant concepts, and describe the query model assumed for our results. Sections 3 and 4 contain our main contributions. In particular, Theorem 2 presents a query-efficient reduction to a population game with a small Lipschitz parameter while preserving the equilibrium. Hence, selecting the parameters appropriately, the hardness of finding well-supported equilibria in general games proven in [3] translates to finding approximate pure equilibria in Lipschitz games. Whilst several papers have discussed both this problem and this technique, none has put forward this observation.

In Section 3.3 we introduce "Multi-Lipschitz" games, a generalization of Lipschitz games that allows player-specific Lipschitz values (the amount of influence each specific player has on all others). We show that certain results of Lipschitz games extend to these more general games, and that the measure of interest is the average of the individual Lipschitz values (in a standard Lipschitz game, they are all equal). Theorem 4 provides a query-efficient reduction from finding equilibria in Multi-Lipschitz games to finding equilibria in Lipschitz games. In particular, whenever there is a query-efficient approximation algorithm for the latter, there is one for the former as well.

Finally, Section 4 provides an extension of the result of [14] showing exponential query lower-bounds for *deterministically* finding approximate Nash equilibria. Theorem 6 provides a more general result for games with more than 2 actions, and Corollary 4 extends this idea further to apply to Lipschitz games.

2. Preliminaries

Throughout, we use the following notation.

- Boldface capital letters denote matrices, and boldface lowercase letters denote vectors.
- The symbol $\mathbf{a}$ is used to denote a pure action profile, and $\mathbf{p}$ is used when the strategy profiles may be mixed.
- $[n]$ and $[m]$ denote the sets $\{1,\dots,n\}$ of players and $\{1,\dots,m\}$ of actions, respectively. Furthermore, $i \in [n]$ will always refer to a player, and $j \in [m]$ will always refer to an action.
- Whenever a query returns an approximate answer, the payoff vector $\tilde{\mathbf{u}}$ will be used to represent the approximation and $\mathbf{u}$ will represent the true value.
- A logarithm with no specified base will be assumed to be base 2, while the natural logarithm of x will be denoted $\ln x$.

2.1. The game model

We introduce standard concepts of strategy profiles, payoffs, regret, and equilibria for pure and mixed strategies.

Types of strategy profile; notation:

- A *pure* action profile $\mathbf{a} = (a_1,\dots,a_n) \in [m]^n$ is an assignment of one action to each player. We use $\mathbf{a}_{-i} = (a_1,\dots,a_{i-1},a_{i+1},\dots,a_n) \in [m]^{n-1}$ to denote the set of actions played by players in $[n]\setminus\{i\}$.
- A (possibly *mixed*) strategy profile $\mathbf{p} = (p_1,\dots,p_n) \in (\Delta[m])^n$ (where $\Delta(S)$ is the probability simplex over S) is a collection of n independent probability distributions, each taken over the action set of a player, where p_{ij} is the probability with which player i plays action j. The set of distributions for players in $[n]\setminus\{i\}$ is denoted $\mathbf{p}_{-i} = (p_1,\dots,p_{i-1},p_{i+1},\dots,p_n)$. When $\mathbf{p}$ contains just 0-1 values, $\mathbf{p}$ is equivalent to some action profile $\mathbf{a} \in [m]^n$.
  Furthermore, when considering binary-action games with action set $\{1,2\}$, we instead describe strategy profiles by $\mathbf{p} = (p_1,\dots,p_n)$, where p_i is the probability that player i plays action 1.
  Abusing notation slightly, we will denote
  $$\mathbf{p}(\mathbf{a}) = \Pr_{\mathbf{a}'\sim\mathbf{p}^*}(\mathbf{a}'=\mathbf{a}) = \prod_{i=1}^{n} p_{i,a_i}.$$
  (the probability that $\mathbf{a}$ is drawn at random from distribution $\mathbf{p}$).

Notation for payoffs: Given player i, action j, and pure action profile $\mathbf{a}$,

- $u_i(j,\mathbf{a}_{-i})$ is the payoff that player i obtains for playing action j when all other players play the actions given in $\mathbf{a}_{-i}$.
- $u_i(\mathbf{a}) = u_i(a_i,\mathbf{a}_{-i})$ is the payoff that player i obtains when all players play the actions given in $\mathbf{a}$.
- Similarly for mixed-strategy profiles:
  $$u_i(j,\mathbf{p}_{-i}) = \mathbb{E}_{\mathbf{a}_{-i}\sim\mathbf{p}_{-i}}[u_i(j,\mathbf{a}_{-i})], \qquad u_i(\mathbf{p}) = \mathbb{E}_{\mathbf{a}\sim\mathbf{p}}[u_i(\mathbf{a})].$$

Beginning our definitions, we introduce the class of games that is the focus of this work:

Definition 1 *(Lipschitz games [1]).* A game is λ*-Lipschitz* if, for every player $i \in [n]$, all action profiles $\mathbf{a}, \mathbf{a}'$ such that

$$||\mathbf{a}_{-i} - \mathbf{a}'_{-i}||_0 = 1$$

satisfy

$$|u_i(a_i, \mathbf{a}_{-i}) - u_i(a_i, \mathbf{a}'_{-i})| \leq \lambda.$$

Here we consider games in which all payoffs are in the range $[0, 1]$ (in particular, note that any general game is, by definition, 1-Lipschitz).

The set of n-player, m-action, λ-Lipschitz games will be denoted $\mathcal{G}(n, m, \lambda)$.

Definition 2 *(Best response).* Given a player i and a strategy profile $\mathbf{p}$, define the *best response*

$$\mathrm{Br}_i(\mathbf{p}) = \arg\max_{j \in [m]} u_i(j, \mathbf{p}_{-i}).$$

In addition, for $\epsilon > 0$, any action j' satisfying

$$u_i(\mathbf{p}) - u_i(j', \mathbf{p}_{-i}) \leq \epsilon$$

is labeled an ϵ*-best response*.

Definition 3 *(Regret).* Given a player i and a strategy profile $\mathbf{p}$, define the regret

$$\mathrm{reg}_i(\mathbf{p}) = \max_{j \in [m]} u_i(j, \mathbf{p}_{-i}) - u_i(\mathbf{p})$$

to be the difference between the payoffs of player i's best response to $\mathbf{p}_{-i}$ and i's strategy p_i.

Definition 4 *(Equilibrium).*

- An ϵ*-approximate Nash equilibrium* (ϵ-ANE) is a strategy profile $\mathbf{p}^*$ such that, for every player $i \in [n]$, $\mathrm{reg}_i(\mathbf{p}^*) \leq \epsilon$.
- An ϵ*-well supported Nash equilibrium* (ϵ-WSNE) is an ϵ-ANE $\mathbf{p}^*$ for which every action j in the support of p_i^* is an ϵ-best response to $\mathbf{p}^*_{-i}$.
- An ϵ*-approximate pure Nash equilibrium* (ϵ-PNE) is a pure action profile $\mathbf{a}$ such that, for every player $i \in [n]$, $\mathrm{reg}_i(\mathbf{a}) \leq \epsilon$.

Note that any ϵ-PNE is an ϵ-WSNE.

2.2. *The query model*

This section introduces the model of queries we consider.

Definition 5 *(Query).*

- A *profile query* of a pure action profile $\mathbf{a}$ of a game G, denoted $\mathcal{Q}^G(\mathbf{a})$, returns a vector $\mathbf{u}$ of payoffs $u_i(\mathbf{a})$ for each player $i \in [n]$.
- A δ*-distribution query* of a strategy profile $\mathbf{p}$ of a game G, denoted $\mathcal{Q}^G_\delta(\mathbf{p})$, returns a vector $\tilde{\mathbf{u}}$ of n values such that $\left|\left|\tilde{\mathbf{u}} - \mathbf{u}\right|\right|_\infty \leq \delta$, where $\mathbf{u}$ is the players' expected utilities from $\mathbf{p}$. We also define a (δ, γ)*-distribution query* to be a δ-distribution query of a strategy profile $\mathbf{p}$ in which every action j in the support of p_i is allocated probability at least γ for every player $i \in [n]$.
- The *(profile) query complexity* of an algorithm A on input game G is the number of calls A makes to $\mathcal{Q}^G$. The δ*-distribution query complexity* of A is the number of calls A makes to $\mathcal{Q}^G_\delta$.

Babichenko [3] points out that it is uninteresting to consider 0-distribution queries, as any game in which every payoff is a multiple of $1/M$ for some $M \in \mathbb{N}$ can be completely learned by a single 0-distribution query. On the other hand, additive approximations to the expected payoffs can be computed via sampling from $\mathbf{p}$. Indeed, for general binary-action games we have from [11]:

Theorem 1 *([11]). Take $G \in \mathcal{G}(n, 2, 1)$, $\eta > 0$. Any (δ, γ)-distribution query of G can be simulated with probability at least $1 - \eta$ by*

$$\max\left\{\frac{1}{\gamma\delta^2}\log\left(\frac{8n}{\eta}\right), \frac{8}{\gamma}\log\left(\frac{4n}{\eta}\right)\right\}$$

profile queries.

Corollary 1. *Take $G \in \mathcal{G}(n, 2, 1)$, $\eta > 0$. Any (δ, γ)-distribution query of G can be simulated with probability at least $1 - \eta$ by*

$$\frac{8}{\gamma^2\delta^2}\log^2\left(\frac{8n}{\eta}\right)$$

profile queries (i.e. the simulation algorithm will output a vector of values $\tilde{u}_i$, each within δ of player i's true payoff u_i with probability at least $1 - \eta$). Furthermore, any algorithm making q (δ, γ)-distribution queries of G can be simulated with probability at least $1 - \eta$ by

$$\frac{8q}{\gamma^2\delta^2}\log^2\left(\frac{8nq}{\eta}\right) = \text{poly}\left(n, \frac{1}{\gamma}, \frac{1}{\delta}, \log\frac{1}{\eta}\right) \cdot q \log q$$

profile queries.

Proof. The first claim is a weaker but simpler version of the upper bound of Theorem 1. The second claim follows from the first by a union bound. □

3. Lipschitz games

3.1. Section-specific preliminaries

In this section we introduce a few general ideas that we will use to prove our results.

The induced population game We first introduce a reduction utilized by [1] in an alternative proof of Nash's Theorem, and by [2] to upper bound the support size of ϵ-ANEs.

Definition 6 *(Induced population game).* Given a game G with payoff function $\mathbf{u}$, we define the *population game* induced by G, $G' = g_G(L)$ with payoff function $\mathbf{u}'$ as follows. Every player i is replaced by a population of L players (v^i_ℓ for $\ell \in [L]$), each playing G against the aggregate behavior of the other $n - 1$ populations. More precisely, $u'_{v^i_\ell}(\mathbf{p}') = u_i\left(p'_{v^i_\ell}, \mathbf{p}_{-i}\right)$ where

$$p_{i'} = \frac{1}{L}\sum_{\ell=1}^{L} p'_{v^{i'}_\ell} \tag{1}$$

for all $i' \neq i$.

Population games date back even to Nash's thesis [15], in which he uses them to justify the consideration of mixed equilibria. To date, the reduction to the induced population game has been focused on proofs of existence. We show that the reduction can be made query-efficient: an equilibrium of $g_G(L)$ induces an equilibrium on G *which can be found with few additional queries*. This technique is the foundation for the main results of this work. We begin by establishing the following preliminary claim as a warm-up:

Claim. Any δ-distribution query of $G' = g_G(L)$ can be simulated by Ln δ-distribution queries of G.

At a high level, this is because we will make a single query of the original game for each player $v^{(i)}_\ell$ in the population game in which we assume that player i plays the strategy of player $v^{(i)}_\ell$ and the remaining $n - 1$ players play the aggregate behavior of their entire populations. More formally:

Proof of Claim. Consider any δ-distribution query of the Ln-player game G'. Such a query takes a mixed strategy profile $\mathbf{p}'$ and returns a δ-approximate payoff for each of the Ln players. If we consider the aggregate strategy profile $\mathbf{p}$ directly, the query $\mathcal{Q}^G_\delta(\mathbf{p})$ will return the payoff to each of the n players in G. However, we cannot derive the payoff to a player $v^{(i)}_\ell$ in G' from the average payoff to each such player. Instead, for every $i \in [n]$, $\ell \in [L]$, consider the strategy profile $\mathbf{p}^{(i)}_\ell$ replacing player i with player $v^{(i)}_\ell$ and replacing the remaining $n - 1$ players by the aggregate behavior of their populations. Then, a query $\mathcal{Q}^G_\delta(\mathbf{p}^{(i)}_\ell)$ will return a vector of n payoffs, the i^{th} element of which will correctly be a δ-approximation of the payoff to player $v^{(i)}_\ell$ when $\mathbf{p}'$ is played in game G'. □

With the above claim in mind, we can now prove the following lemma:

Lemma 1. *Given an n-player, m-action game G and a population game $G' = g_G(L)$ induced by G, if an ϵ-PNE of G' can be found by an algorithm making q (δ, γ)-distribution queries of G', then an ϵ-WSNE of G can be found by an algorithm making $n \cdot m \cdot q$ $(\delta, \gamma/L)$-distribution queries of G.*

To complete the proof of this lemma, we will improve the blowup factor from Ln (in the claim above) to mn (since we are only considering binary-action games, this will ensure only a linear blowup). In essence, this is because we do not actually need to make a query for every single player in each population. Instead, if we query every pure strategy for player i against the aggregate behavior of the other $n-1$ populations, we can then calculate the payoff to *any* mixed strategy of a player $v_\ell^{(i)}$ in population i.

Lemma 1. Consider any δ-distribution query of the Ln-player game G'. In the proof of the claim above, this query is simulated by a single query for each player in G'. Instead, we can simulate it by m queries for every player in G. Instead of queries $\mathbf{p}_\ell^{(i)}$ for every $i \in [n]$, $\ell \in [L]$, consider queries $\mathbf{p}_j^{(i)}$ for every $i \in [n]$, $j \in [m]$ in which player i plays action j and the remaining $n-1$ players still play the aggregate behavior of their populations. From these $n \cdot m$ queries, one can derive the payoffs from the Ln queries $\mathbf{p}_\ell^{(i)}$ above by directly calculating the expected payoff to the mixed strategy played by player $v_\ell^{(i)}$ given the payoffs for each pure action. So each δ-distribution query of G' can be simulated by $n \cdot m$ δ-distribution queries of G. Furthermore, if the query was a (δ, γ)-distribution query of G', the aggregate behavior of L players guarantees that the new queries are all $(\delta, \gamma/L)$-distribution queries. Finally, any ϵ-PNE $\mathbf{a}'^*$ of G' must also induce an ϵ-WSNE $\mathbf{p}^*$ of G in which each player of G plays the aggregate strategy of their population (as defined in (1) above). If $\mathbf{p}^*$ were not an ϵ-WSNE, then some player in G' must suffer regret $> \epsilon$ in $\mathbf{a}'^*$ (as a strategy profile is an ϵ-WSNE if every action in the support of every player is an ϵ-best response). Since we can find $\mathbf{p}^*$, and thus $\mathbf{a}'^*$, with only a blowup of nm queries, this completes the proof of Lemma 1. $\square$

Scaling payoffs Next we introduce a very simple, but important, observation. It allows us to apply results to games with different Lipschitz parameters.

Observation 1. *For any constants $\lambda' < \lambda \leq 1, \epsilon > 0$, there is a query-free reduction from finding ϵ-approximate equilibria of games in $\mathcal{G}(n, m, \lambda)$ to finding $\frac{\lambda'}{\lambda}\epsilon$-approximate equilibria of games in $\mathcal{G}(n, m, \lambda')$.*

In other words, query complexity upper bounds hold as λ and ϵ are scaled up together, and query complexity lower bounds hold as they are scaled down. The proof is very simple - the reduction multiplies every payoff by $\frac{\lambda'}{\lambda}$ (making no additional queries) and outputs the result. Note that the lemma does not hold for $\lambda' > \lambda$, as the reduction could introduce payoffs that are larger than 1.

Simulating profile queries Finally, we note that we can simulate profile query algorithms using δ-distribution queries for small enough values of δ. This fact should be unsurprising, but care is required to ensure that the value of δ is sufficient.

Lemma 2. *Take any $\delta \leq \min\{\epsilon/4, \lambda/4\}$. Given a game $G \in \mathcal{G}(n, 2, \lambda)$, there is another game $G' \in \mathcal{G}\left(n, 2, \frac{3\lambda}{2}\right)$ such that*

(1) For any action profile $\mathbf{a} \in \{0,1\}^n$, $||\mathbf{u}'(\mathbf{a}) - \mathbf{u}(\mathbf{a})||_\infty \leq \delta$.
(2) $G' \in \mathcal{G}\left(n, 2, \frac{3\lambda}{2}\right)$.
(3) Any $\epsilon/2$-PNE of G' is an ϵ-PNE of G.

Fig. 1 visually depicts this observation.

Proof. In order to prove the lemma, we need to find a game G' that agrees with all distribution queries of G satisfying the three requirements. So consider the incomplete payoff function $\tilde{\mathbf{u}}$ learned by δ-distribution queries of G. Define the game G' agreeing with $\tilde{\mathbf{u}}$ on all queried actions and agreeing with G on all others:

$$\mathbf{u}'(\mathbf{a}) = \begin{cases} \tilde{\mathbf{u}}(\mathbf{a}) & \mathbf{a} \text{ was queried} \\ \mathbf{u}(\mathbf{a}) & \text{otherwise} \end{cases}$$

G' is well-defined, and for every action profile $\mathbf{a} \in \{0,1\}^n$,

$$\left|\left|\mathbf{u}'(\mathbf{a}) - \mathbf{u}(\mathbf{a})\right|\right|_\infty \leq \left|\left|\tilde{\mathbf{u}}(\mathbf{a}) - \mathbf{u}(\mathbf{a})\right|\right|_\infty \leq \delta,$$

so G' satisfies requirement (1).

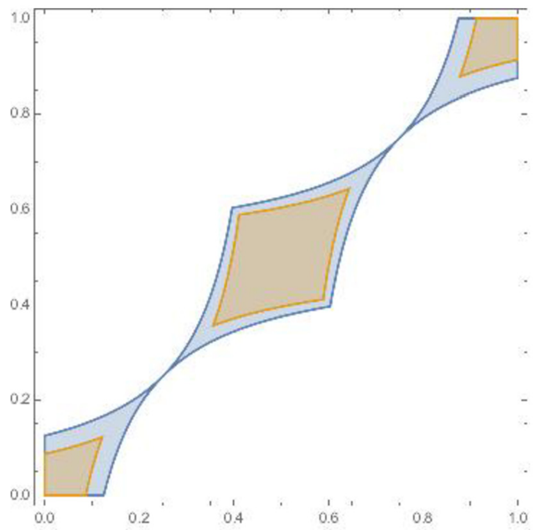


Fig. 1. Taking G to be the Coordination Game for fixed values of ϵ and δ, the outer region shows the set of ϵ-approximate equilibria of G while the inner region shows the set of all $\frac{\epsilon}{2}$-approximate equilibria of any possible game G' in which each payoff may be perturbed by at most δ.

Now, consider any player $i \in [n]$ and any action profiles $\mathbf{a}^{(1)}, \mathbf{a}^{(2)} \in \{0,1\}^n$ such that

$$\left|\left|\mathbf{a}_{-i}^{(1)} - \mathbf{a}_{-i}^{(2)}\right|\right|_1 = c \geq 1.$$

Then, since G is λ-Lipschitz,

$$\begin{aligned} \left|u_i'\left(\mathbf{a}^{(1)}\right) - u_i'\left(\mathbf{a}^{(2)}\right)\right| &\leq \left|u_i\left(\mathbf{a}^{(1)}\right) - u_i\left(\mathbf{a}^{(2)}\right)\right| + 2\delta \\ &\leq c\lambda + 2\delta \\ &\leq c\left(\lambda + 2\delta\right) \\ &< c\frac{3\lambda}{2} \qquad \text{(using the fact that } \delta \leq \lambda/4) \end{aligned}$$

This proves that G' satisfies requirement (2).

Finally, consider any $\epsilon/2$-approximate pure Nash equilibrium $\mathbf{a}^*$ of G, and note that, for any player $i \in [n]$,

$$u_i(\mathbf{a}^*) \geq u_i'(\mathbf{a}^*) - \delta, \qquad u_i(j, \mathbf{a}_{-i}^*) \leq u_i'(j, \mathbf{a}_{-i}^*) + \delta$$

for any $j \in \{0,1\}$. Combining these two inequalities, and keeping in mind that $\delta < \epsilon/4$,

$$\text{reg}_i'(\mathbf{a}^*) \leq \frac{\epsilon}{2} + 2\delta \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon$$

proving requirement (3) and the claim. □

3.2. Query complexity of equilibria in Lipschitz games

We are now ready to present our main result:

Theorem 2 *(Main result). There exists some constant ϵ_0 such that, for any*

$$n \in \mathbb{N}, \qquad \epsilon < \epsilon_0, \qquad \lambda \leq \frac{\epsilon}{\sqrt{8n\log 4n}},$$

while every game in $\mathcal{G}(n, 2, \lambda)$ has an ϵ-PNE, any randomized algorithm finding such equilibria with probability at least $\beta = 1/\text{poly}(n)$ must make $\lambda^2 2^{\Omega(n\lambda/\epsilon)}$ profile queries.

We will rely on the following result of Babichenko.

Theorem 3 *([3]). There is a constant $\epsilon_0 > 0$ such that, for any $\beta = 2^{-o(n)}$, the randomized δ-distribution query complexity of finding an ϵ_0-WSNE of n-player binary-action games with probability at least β is $\delta^2 2^{\Omega(n)}$.*

For the remainder of this work, the symbol ϵ_0 refers to this specific constant. A simple application of Observation 1 yields

Corollary 2. *There is a constant $\epsilon_0 > 0$ such that, for any $\beta = 2^{-o(n)}$, the randomized δ-distribution query complexity of finding an $\epsilon_0\lambda$-WSNE of games in $\mathcal{G}(n, 2, \lambda)$ with probability at least β is $\delta^2 2^{\Omega(n)}$.*

With Corollary 2 in mind, we can now prove Theorem 2. We will break the proof into parts, proceeding via contradiction. As an overview:

Step 1 (Algorithm A) Take $q = \lambda^2 2^{o(n\lambda/\epsilon)}$ and assume such an algorithm A exists making q profile queries of games $G \in \mathcal{G}(n, 2, \lambda)$.
Step 2 (Algorithm B) Convert it to an algorithm B making q δ-distribution queries.
Step 3 (Algorithm C) Use Lemma 1 to derive an algorithm C finding $\epsilon_0\lambda$-WSNE in λ-Lipschitz games contradicting the lower bound of Corollary 2.

Proof. (Theorem 2) Assume for the sake of contradiction that some such algorithm A exists finding ϵ-PNEs of games in $\mathcal{G}(n, 2, \lambda)$ making at most $\lambda^2 2^{o(n\lambda/\epsilon)}$ profile queries. Consider any

$$\epsilon < \epsilon_0, \qquad \lambda' < \frac{\epsilon}{\sqrt{8n \log 4n}},$$

and define

$$\lambda = \frac{\epsilon}{\epsilon_0}, \qquad L = \frac{\lambda}{\lambda'}, \qquad N = Ln.$$

Over the course of three steps we derive an algorithm C (with an intermediate algorithm B) that contradicts Corollary 2.

Step 1: Algorithm A

Given a game $G'' \in \mathcal{G}\left(N, 2, \frac{3\lambda'}{2}\right)$, algorithm A can find an $\epsilon/2$-PNE of G'' with probability at least $2^{-o(N)}$ making at most

$$\left(\frac{3\lambda'}{2}\right)^2 2^{o\left(\frac{3N\lambda'/2}{\epsilon/2}\right)} = \lambda'^2 2^{o(N\lambda'/\epsilon)}$$

profile queries. In fact, the success probability can be amplified to constant by running algorithm A $O(N)$ times.

Step 2: Algorithm B

Consider an algorithm that only has access to distribution queries. We will want to show that such an algorithm can still simulate algorithm A well enough to find an approximate equilibrium of its input.

Let $\delta = \frac{\epsilon_0\lambda'}{4}$. For any game $G' \in \mathcal{G}(N, 2, \lambda')$, consider an algorithm making δ-distribution queries of *pure action profiles* of G' (introducing the uncertainty without querying mixed strategies). Since $\epsilon_0 < 1$ and $\lambda' < \epsilon$, δ satisfies the condition of Lemma 2 and thus this algorithm learns a game $G'' \in \mathcal{G}\left(N, 2, \frac{3\lambda'}{2}\right)$ for which every $\epsilon/2$-PNE is also an ϵ-PNE of G' (i.e. the payoffs provided to this algorithm are consistent with at least one such game).

So define the algorithm B that takes input G' and proceeds as though it is algorithm A (but makes δ-distribution queries instead). By the claim above, after at most $\lambda'^2 2^{o(N\lambda'/\epsilon)}$ queries, it has found an $\epsilon/2$-PNE of some $G'' \in \mathcal{G}\left(N, 2, \frac{3\lambda'}{2}\right)$ that it believes it has learned (i.e. one that is consistent with all the payoff queries), which is also guaranteed to be an ϵ-PNE of G' (we can still promise a constant success probability for this simulation). Fig. 1 shows the acceptable outputs of algorithm B as a superset of the possible outputs of algorithm B.

Step 3: Algorithm C

Consider any game $G \in \mathcal{G}(n, 2, \lambda)$, and let $G' = g_G(L)$ be the population game induced by G. There is an algorithm C described by Lemma 1 that takes input G and simulates algorithm B on G' (making

$$2n \cdot \lambda'^2 2^{o(N\lambda'/\epsilon)} = \delta^2 2^{o(n\lambda/\epsilon)}$$

δ-distribution queries) and correctly outputs an ϵ-WSNE (i.e. an $\epsilon_0\lambda$-WSNE) of G with constant success probability (so certainly at least $2^{-o(n)}$).

The existence of algorithm C directly contradicts the result of Corollary 2, proving that algorithm A cannot exist. □

Remark 1. Note that, with small changes to the values of the parameters, this lower bound can apply to profile or δ-distribution query complexity, and to pure or mixed approximate equilibria.

3.3. Multi-Lipschitz games

In this section, we consider a generalization of Lipschitz games in which each player $i \in [n]$ has a "player-specific" Lipschitz value λ_i in the sense that, if player i changes actions, the payoffs of all other players are changed by at most λ_i.

Definition 7 *(Multi-Lipschitz game).* A Λ-Multi-Lipschitz game G is an n-player, m-action game G in which each player $i \in [n]$ is associated with a constant $\lambda_i \leq 1$ such that

$$\frac{1}{n}\sum_{i'=1}^{n} \lambda_{i'} = \Lambda$$

and, for any player $i' \neq i$ and action profiles $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}$ with $\mathbf{a}_{-i}^{(1)} = \mathbf{a}_{-i}^{(2)}$,

$$\left| u_{i'}\left(\mathbf{a}^{(1)}\right) - u_{i'}\left(\mathbf{a}^{(2)}\right) \right| \leq \lambda_i.$$

The class of such games is denoted $\mathcal{G}_\Lambda(n, m)$, and for simplicity it is assumed that $\lambda_1 \leq \ldots \leq \lambda_n$.

The consideration of this generalized type of game allows real-world situations to be more accurately modeled. Geopolitical circumstances, for example, naturally take the form of Multi-Lipschitz games, since individual countries have different limits on how much their actions can affect the rest of the world. Financial markets present another instance of such games; they not only consist of individual traders who have little impact on each other, but also include a number of institutions that might each have a much greater impact on the market as a whole. This consideration is further motivated by the recent GameStop frenzy; the institutions still wield immense power, but so do the aggregate actions of millions of individuals [18].

Notice that a λ-Lipschitz game is also a λ-Multi-Lipschitz game. Equivalently, any algorithm that finds ϵ-ANEs of Λ-Multi-Lipschitz games is also applicable to Λ-Lipschitz games. Theorem 4 shows a kind of converse for query complexity, reducing from finding ϵ-ANE of Λ-Multi-Lipschitz games to finding ϵ-ANE of λ-Lipschitz games, for λ a constant multiple of Λ.

Theorem 4 *(Multi-Lipschitz reduction). If ϵ-ANEs of games in $\mathcal{G}_\Lambda(n, 2)$ can be computed with probability at least $1-\eta$ by an algorithm making q queries, then $\epsilon/2$-ANEs of games in $\mathcal{G}\left(2n, 2, \frac{3\Lambda}{2}\right)$ can be computed with probability at least $1-\eta$ by an algorithm making*

$$poly(n, \frac{1}{\epsilon}, \log\frac{1}{\eta})q\log q$$

queries.

As we now consider ϵ-ANEs, existence is no longer a question: such equilibria are *always* guaranteed to exist by Nash's Theorem [16]. This proof will also utilize a more general population game $G' = g_G(L_1, \ldots, L_n)$ in which player i is replaced by a population of size L_i (where the L_i may differ from each other), and the queries in Lemma 1 become $(\delta, \min_{i\in[n]}\{\gamma/L_i\})$-distribution queries (this will now be relevant, as we need to apply Corollary 1). Otherwise, the proof follows along the same lines as that of Theorem 2.

Proof. Consider some $\epsilon > 0$ and a game $G \in \mathcal{G}_\Lambda(n, 2)$ (WLOG take $\lambda_1 \leq \ldots \leq \lambda_n$). First, if $\Lambda < \epsilon/n^2$, finding an ϵ-ANE is trivial (each player can play their best-response to the uniform mixed strategy, found in $2n$ queries). So assume $\Lambda \geq \epsilon/n^2$. Define $L_i = \max\{\frac{\lambda_i}{\Lambda}, 1\}$ and, taking $i' = \max_{i\in[n]}\{i : L_i = 1\}$, note that

$$\sum_{i=1}^{n} L_i = \sum_{i=1}^{i'} 1 + \sum_{i=i'+1}^{n} \frac{\lambda_i}{\Lambda} = \sum_{i=1}^{i'} 1 + \frac{1}{\Lambda}\sum_{i=i'+1}^{n} \lambda_i \leq \sum_{i=1}^{i'} 1 + \frac{n\Lambda}{\Lambda} \leq 2n.$$

Thus the population game $G' = g_G(L_1, \ldots, L_n) \in \mathcal{G}(2n, 2, \Lambda)$.

Step 1: Algorithm A

Consider an algorithm A that finds $\epsilon/2$-ANEs of games in $\mathcal{G}\left(2n, 2, \frac{3\Lambda}{2}\right)$, with probability at least $1-\eta$ making q profile queries.

Step 2: Algorithm B

Taking

$$\delta = \frac{\epsilon^2}{4n^2} \leq \frac{\epsilon\Lambda}{4},$$

we can once again apply Lemma 2 to construct the same algorithm B from the proof of Theorem 2 that simulates A but makes $(\delta, 1)$-distribution queries and finds an ϵ-ANE of G'.

Step 3: Algorithm C

By Lemma 1, there is an algorithm C on input $G \in \mathcal{G}_\Lambda(n, 2)$ that simulates B (replacing each $(\delta, 1)$-distribution query of G' with $2n$ $(\delta, \frac{1}{2n})$-distribution queries of G since $1/L_n \geq 1/2n$) finding an ϵ-ANE of G' (and thus of G) with probability at least $1 - \eta$.

Applying Corollary 1 (using $\delta = \frac{\epsilon^2}{4n^2}, \gamma = \frac{1}{2n}$) to create a profile-query algorithm from C completes the proof. □

As an example application of Theorem 4, an algorithm of [10] finds $\left(\frac{1}{8} + \alpha\right)$-approximate Nash equilibria of games in $\mathcal{G}\left(n, 2, \frac{1}{n}\right)$; Theorem 5 states that result in detail, and Corollary 3 extends it to Multi-Lipschitz games.

Theorem 5 *([10]). Given constants $\alpha, \eta > 0$, there is a randomized algorithm that, with probability at least $1 - \eta$, finds $\left(\frac{1}{8} + \alpha\right)$-approximate Nash equilibria of games in $\mathcal{G}\left(n, 2, \frac{1}{n}\right)$ making $O\left(\frac{1}{\alpha^4} \log\left(\frac{n}{\alpha\eta}\right)\right)$ profile queries.*

We now have some ability to apply this to Multi-Lipschitz games; for a small range of values we can improve upon the trivial $\frac{1}{2}$-approximate equilibrium of Proposition 1.

Corollary 3. *For $\alpha, \eta > 0$, $\Lambda \geq \frac{1}{n}, \epsilon \geq \frac{n\Lambda}{8} + \alpha$, there is an algorithm finding ϵ-ANEs of games in $\mathcal{G}_\Lambda(n, 2)$ with probability at least $1 - \eta$ making at most $\text{poly}(n, \frac{1}{\alpha}, \log \frac{1}{\eta})$ profile queries.*

Remark 2. This is actually a slight improvement over just combining Theorems 4 and 5, since the choice of δ can be made slightly smaller to shrink α as necessary.

4. General deterministic lower bounds

In this section, we provide lower bounds on the *deterministic* query complexity of finding approximate Nash equilibria in order to motivate the consideration of explicitly randomized algorithms.

4.1. Nash equilibria

While the proof in [14] utilizes a reduction from `ApproximateSink` to provide a lower bound for correlated equilibria, we employ a more streamlined approach when it comes to Nash equilibria, presenting an explicit family of "hard" games that allows us to uncover the optimal value of ϵ as a function of the number of actions:

Theorem 6. *Given some $m \in \mathbb{N}$, for any $\epsilon < \frac{m-1}{m}$, the deterministic profile query complexity of finding ϵ-ANEs of n-player, m-action games is $2^{\Omega(n)}$.*

Furthermore, this value of ϵ cannot be improved:

Proposition 1. *Given some $n, m \in \mathbb{N}$, for any $\epsilon \geq \frac{m-1}{m}$, an ϵ-ANE of an n-player, m-action game can be found making no profile queries.*

The upper bound of Proposition 1 can be met if every player plays the uniform mixed strategy over their actions. Finally, we can apply Observation 1 to scale Theorem 6 and obtain a Lipschitz version of this result:

Corollary 4. *Given some $m \in \mathbb{N}$, $\lambda \in (0, 1]$, for any $\epsilon < \frac{m-1}{m}\lambda$, the deterministic profile query complexity of finding ϵ-ANEs of n-player, m-action, λ-Lipschitz games is $2^{\Omega(n)}$.*

In order to prove these results, we introduce a family of games $\{G_{k,m}\}$. For any $k, m \in \mathbb{N}$, $G_{k,m}$ is a $2k$-player, m-action generalization of k Matching Pennies games in which every odd player $2i - 1$ wants to match the even player $2i$ and every even player $2i$ wants to mismatch with the odd player $2i - 1$.

Definition 8 *(Generalized Matching Pennies).* Define $G_{1,2}$ to be the generalized Matching Pennies game, as described in Fig. 2(a). Define the generalization $G_{k,m}$ to be the $2k$-player m-action game such that, for any $i \in [k]$, player $2i - 1$ has a payoff 1 for matching player $2i$ and 0 otherwise (and vice versa for player $2i$) ignoring all other players.

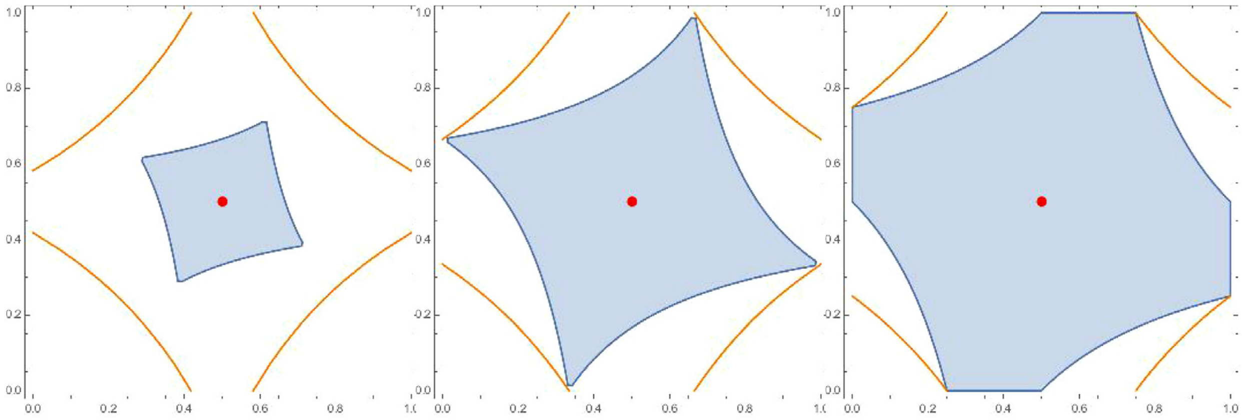
Player 1 \ Player 2	1	2
1	1, 0	0, 1
2	0, 1	1, 0

(a) The payoff matrix of $G_{1,2}$, the Matching Pennies game.

Player 1 \ Player 2	1	2	3
1	1, 0	0, 1	0, 1
2	0, 1	1, 0	0, 1
3	0, 1	0, 1	1, 0

(b) The payoff matrix of $G_{1,3}$, the generalized Matching Pennies game.

Fig. 2. The payoff matrices of $G_{1,2}$ and $G_{1,3}$.



(a) $\alpha = \frac{1}{3}$ (b) $\alpha = \frac{1}{6}$ (c) $\alpha \to 0$

Fig. 3. The region of possible values for $(\mathbf{p}_1^*, \mathbf{p}_2^*)$ in any $\left(\frac{1}{2} - \alpha\right)$-approximate Nash equilibrium of $G_{k,2}$. The only exact Nash equilibrium is shown by the center point, and the corresponding values of ρ are displayed as the outer lines.

The critical property of the generalized Matching Pennies game is that we can bound the probability that any given action profile is played in any ϵ-ANE of $G_{k,m}$. If too much probability is jointly placed on matching actions, player 2 will have high regret. Conversely, if too much probability is jointly placed on mismatched actions, player 1 will have high regret.

Lemma 3. *For any $k, m \in \mathbb{N}$, $\alpha > 0$, take $\epsilon = \frac{m-1}{m} - \alpha$. In any ϵ-ANE $\mathbf{p}^*$ of $G_{k,m}$, every pair of players $(2i-1, 2i)$ plays any pair of actions (j_{2i-1}, j_{2i}) with probability less than ρ where*

$$\rho = \frac{(2-\alpha)m - 1}{2m}.$$

This phenomenon can be seen in Fig. 3.

Proof. Define $2k$ to be the number of players in $G_{k,m}$, and consider some ϵ-ANE $\mathbf{p}^*$. Now WLOG consider players 1 and 2 and assume, for the sake of contradiction, that there exist some actions j_1, j_2 such that $p_{1,j_1} p_{2,j_2} \geq \rho$. We will need to consider the two cases $j_1 = j_2$ and $j_1 \neq j_2$.

Matching Actions In this case, WLOG assume $j_1 = j_2 = 1$. We show that player 2 can improve her payoff by more than ϵ. Under $\mathbf{p}^*$, with probability greater than ρ, any random realization will yield player 2 a payoff of 0. In other words,

$$u_2(\mathbf{p}^*) < 1 - \rho.$$

Furthermore, we are guaranteed that

$$\sum_{j=2}^{m} p_{1,j} < 1 - \rho$$

so there must exist some action (WLOG action 2) for which

$$p_{1,2} < \frac{1-\rho}{m-1}.$$

As such,

$$u_2(2, \mathbf{p}^*_{-2}) - u_2(\mathbf{p}^*) > 1 - \frac{1-\rho}{m-1},$$

so

$$\text{reg}_2(\mathbf{p}^*) > \underbrace{\left(1 - \frac{1-\rho}{m-1}\right)}_{u_2(2,\mathbf{p}^*_{-2})} - \underbrace{(1-\rho)}_{u_2(\mathbf{p}^*)} = \frac{\rho m - 1}{m-1} \geq \frac{m-1}{m} - \alpha$$

for all $m \geq 2$. This contradicts our assumption of an ϵ-ANE.

Mismatched Actions In this case, WLOG assume $j_1 = 1, j_2 = 2$. The situation is simpler:

$$u_1(\mathbf{p}^*) < 1 - \rho$$

and

$$u_1(2, \mathbf{p}^*_{-1}) > \rho,$$

so

$$\text{reg}_1(\mathbf{p}^*) > \frac{m-1}{m} - \alpha.$$

This too contradicts our assumption of an ϵ-ANE, and thus completes the proof of Lemma 3. □

We can now prove Theorem 6. The general idea is that, should an efficient algorithm exist, because any equilibrium of $G_{k,m}$ must have large support by Lemma 3, there is significant probability assigned to action profiles that are not queried by the algorithm. We show there is a game that the algorithm cannot distinguish from $G_{k,m}$ that shares no approximate equilibria with $G_{k,m}$.

Proof. (Theorem 6) Consider any $\alpha > 0$ and let $\epsilon = \frac{m-1}{m} - \alpha$. Taking ρ as in the statement of Lemma 3, assume there exists some deterministic algorithm A that takes an n-player, m-action game G as input and finds an ϵ-ANE of G querying the payoffs of $q < \frac{\alpha}{2}\rho^{-\frac{n}{2}}$ action profiles. Fix some $k \in \mathbb{N}$ and consider input $G_{k,m}$ as defined in Definition 8. Then $\mathbf{p}^* = A\left(G_{k,m}\right)$ is an ϵ-ANE of $G_{k,m}$. Note that, for some j,

$$p_{1,j} \leq \frac{1}{m}$$

(WLOG assume $j = 1$).

Now define the perturbation $G'_{k,m}$ of $G_{k,m}$ with payoffs defined to be equal to $G_{k,m}$ for every action profile queried by A, 1 for every remaining action profile in which player 1 plays action 1 (chosen because it is assigned low probability by $\mathbf{p}^*$ by assumption), and 0 otherwise. Note that, by definition, A cannot distinguish between $G_{k,m}$ and $G'_{k,m}$, so $A(G'_{k,m}) = \mathbf{p}^*$.

The quantity we need to bound is

$$\text{reg}_i(\mathbf{p}^*) \geq u'_i(1, \mathbf{p}^*_{-1}) - u'_i(\mathbf{p}^*).$$

We must bound the components of this expression as follows:

Claim *(proven below).* $u'_1(1, \mathbf{p}^*_{-1}) > (1 - q\rho^{\frac{n}{2}})$ and $u'_1(\mathbf{p}^*) < \left(\frac{1}{m} + q\rho^{\frac{n}{2}}\right)$.

Using the claim and once again recalling the assumption that $q < \frac{\alpha}{2}\rho^{-\frac{n}{2}}$, we see

$$\text{reg}'_1(\mathbf{p}^*) > \left(1 - \frac{1}{m} - 2q\rho^{\frac{n}{2}}\right) = \frac{m-1}{m} - \alpha = \epsilon.$$

So $\mathbf{p}^*$ cannot actually be an ϵ-ANE of $G_{k,m}$. This completes the proof of Theorem 6. □

Now we prove the above claim. Let $\mathbf{p}^* = A(G_{k,m})$ be an ϵ-ANE of $G_{k,m}$, and WLOG assume $p_{1,1} \leq 1/m$. Define the sets $Q_i^{(j)}$ to consist of the pure action profiles queried by A in which player i plays action j (as a basic bound, note that none of these sets can contain more than q elements). The perturbation $G'_{k,m}$ is the game with payoffs that agree with $G_{k,m}$ on all queried action profiles of A, and otherwise provide player 1 a payoff of 1 for playing action 1 and a payoff of 0 otherwise. Because A cannot distinguish between $G_{k,m}$ and $G'_{k,m}$, it must hold that $\mathbf{p}^* = A(G'_{k,m})$. The claim is proven below in two parts.

Claim. $u_1'(1, \mathbf{p}^*_{-1}) > (1 - q\rho^{\frac{n}{2}})$.

Proof. We first separate the payoffs into those which were queried by A and those which were not (i.e. those that differ between $G_{k,m}$ and $G'_{k,m}$ and those that do not):

$$u_1'(1, \mathbf{p}^*_{-1}) = \sum_{\mathbf{a} \in Q_1^{(1)}} \mathbf{p}^*(\mathbf{a}) u_1(1, \mathbf{a}_{-1}) + \sum_{\mathbf{a} \notin Q_1^{(1)}} \mathbf{p}^*(\mathbf{a}).$$

Unfortunately, the best lower bound we can claim for the first sum is 0. However, the second sum is more friendly.

$$u_1'(1, \mathbf{p}^*_{-1}) \geq \sum_{\mathbf{a} \notin Q_1^{(1)}} \mathbf{p}^*(\mathbf{a}) = \left(1 - \sum_{\mathbf{a} \in Q_1^{(1)}} \mathbf{p}^*(\mathbf{a})\right).$$

Combining our assumption that $\left|Q_1^{(1)}\right| \leq q$ with the result of Lemma 3 (which provides an upper bound on the value of $\mathbf{p}^*(\mathbf{a})$), this yields

$$u_1'(1, \mathbf{p}^*_{-1}) > \left(1 - q\rho^{\frac{n}{2}}\right),$$

proving the claim. □

Claim. $u_1'(\mathbf{p}^*) < \left(\frac{1}{m} + q\rho^{\frac{n}{2}}\right)$.

Proof. It is slightly more involved to obtain an upper bound on the value of $u_1'(\mathbf{p}^*)$. We first note the following.

$$u_1'(\mathbf{p}^*) = \sum_{\substack{\mathbf{a} \in [m]^n \\ a_1 = 1}} \mathbf{p}^*(\mathbf{a}) u_1'(\mathbf{a}^*) + \sum_{\substack{\mathbf{a} \in [m]^n \\ a_1 \neq 1}} \mathbf{p}^*(\mathbf{a}) u_1'(\mathbf{a}^*).$$

By assumption, the first expression is upper bounded at $\frac{1}{m}$ since without loss of generality we have selected an action to which player 1 does not assign much probability. Our focus, therefore, must lie on the second expression. This can be broken up further, again by the Law of Total probability:

$$\sum_{\substack{\mathbf{a} \in [m]^n \\ a_1 \neq 1}} \mathbf{p}^*(\mathbf{a}) u_1'(\mathbf{a}^*) = \sum_{\substack{\mathbf{a} \in [m]^n \\ a_1 \neq 1 \\ \mathbf{a} \in Q_1^{(a_1)}}} \mathbf{p}^*(\mathbf{a}) u_1'(\mathbf{a}^*) + \sum_{\substack{\mathbf{a} \in [m]^n \\ a_1 \neq 1 \\ \mathbf{a} \notin Q_1^{(a_1)}}} \mathbf{p}^*(\mathbf{a}) u_1'(\mathbf{a}^*)$$

Since $u_1'(\mathbf{p}^*) = 0$ whenever $\mathbf{a} \notin Q_1^{(a_1)}$, we need only consider the case when $\mathbf{a} \in Q_1^{(a_1)}$. While we cannot bound the payoff we can, fortunately, bound the probability

$$\Pr_{\mathbf{a} \sim \mathbf{p}^*}(a_1 \neq 1, \mathbf{a} \in Q_1^{(a_1)}) \leq q\rho^{\frac{n}{2}}.$$

This bounds the entire payoff at

$$u_1'(\mathbf{p}^*) \leq \left(\frac{1}{m} + q\rho^{\frac{n}{2}}\right). \quad \square$$

5. Further directions

An important additional question is the query complexity of finding ϵ-PNEs of n-player, λ-Lipschitz games in which $\epsilon = \Omega(n\lambda)$. Theorem 2 says nothing in this parameter range, yet Theorem 5 provides a logarithmic upper bound in this regime. The tightness of this bound is of continuing interest.

Furthermore, the query- and computationally-efficient reduction discussed in Lemma 1 provides a hopeful avenue for further results bounding the query, and computational, complexity of finding equilibria in many other classes of games. Indeed, the authors have also utilized this reduction to prove the following:

Theorem 7 *([9]). There exists some constant $\epsilon > 0$ such that, for all functions $\lambda(n) = \Theta(n^{-\frac{3}{4}})$, the problem of finding ϵ-approximate pure Nash equilibria of n-player, binary-action, λ-Lipschitz games is* ***PPAD****-complete.*

Further results of interest are likely to result from applying this technique in various scenarios.

Finally, while Theorem 6 is able to describe a complete dichotomy for the deterministic query complexity of finding Nash equilibria, no such dichotomy is currently known for *correlated* equilibria. While it seems unlikely that a $\frac{1}{2}$-CCE can be easily found in games with any number of actions, the proof technique of [14] may be too weak to prove otherwise. This problem clearly demands further attention.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Matthew J. Katzman reports financial support was provided by DeepMind Technologies Ltd.

Acknowledgements

We thank Francisco Marmalejo Cossío and Rahul Santhanam for their support in the development of this work, and the reviewers of earlier versions for helpful comments. Matthew Katzman was supported by an Oxford-DeepMind Studentship for this work.

References

[1] Y. Azrieli, E. Shmaya, Lipschitz games, Math. Oper. Res. 38 (2) (2013) 350–357, https://doi.org/10.1287/moor.1120.0557.
[2] Y. Babichenko, Small support equilibria in large games, CoRR, arXiv:1305.2432 [abs], 2013, https://arxiv.org/abs/1305.2432.
[3] Y. Babichenko, Query complexity of approximate Nash equilibria, J. ACM 63 (4) (2016) 36, https://doi.org/10.1145/2908734.
[4] Y. Babichenko, Fast convergence of best-reply dynamics in aggregative games, Math. Oper. Res. 43 (1) (2018) 333–346, https://doi.org/10.1287/moor.2017.0868.
[5] Y. Babichenko, Informational bounds on equilibria (a survey), ACM SIGecom Exch. 17 (2) (2019) 25–45, https://doi.org/10.1145/3381329.3381333.
[6] X. Chen, Y. Cheng, B. Tang, Well-supported vs. approximate Nash equilibria: query complexity of large games, in: C.H. Papadimitriou (Ed.), 8th Innovations in Theoretical Computer Science Conference, ITCS, vol. 67, LIPIcs, 2017, pp. 57:1–57:9.
[7] C. Daskalakis, C.H. Papadimitriou, Approximate Nash equilibria in anonymous games, J. Econ. Theory 156 (2015) 207–245, https://doi.org/10.1016/j.jet.2014.02.002.
[8] A. Deligkas, J. Fearnley, P.G. Spirakis, Lipschitz continuity and approximate equilibria, Algorithmica 82 (10) (2020) 2927–2954, https://doi.org/10.1007/s00453-020-00709-3.
[9] P.W. Goldberg, M.J. Katzman, PPAD-complete pure approximate Nash equilibria in Lipschitz games, in: P. Kanellopoulos, M. Kyropoulou, A.A. Voudouris (Eds.), Algorithmic Game Theory - 15th International Symposium, SAGT 2022, Colchester, UK, September 12-15, 2022, Proceedings, in: Lecture Notes in Computer Science, vol. 13584, Springer, 2022, pp. 169–186.
[10] P.W. Goldberg, F.J. Marmolejo Cossío, Z.S. Wu, Logarithmic query complexity for approximate Nash computation in large games, Theory Comput. Syst. 63 (1) (2019) 26–53, https://doi.org/10.1007/s00224-018-9851-8.
[11] P.W. Goldberg, A. Roth, Bounds for the query complexity of approximate equilibria, ACM Trans. Econ. Comput. 4 (4) (2016) 24.
[12] P.W. Goldberg, S. Turchetta, Query complexity of approximate equilibria in anonymous games, J. Comput. Syst. Sci. 90 (2017) 80–98, https://doi.org/10.1016/j.jcss.2017.07.002.
[13] S. Hart, Y. Mansour, How long to equilibrium? The communication complexity of uncoupled equilibrium procedures, Games Econ. Behav. 69 (1) (2010) 107–126.
[14] S. Hart, N. Nisan, The query complexity of correlated equilibria, Games Econ. Behav. 108 (2018) 401–410, https://doi.org/10.1016/j.geb.2016.11.003.
[15] J. Nash, Non-Cooperative Games, Ph.D. thesis, Princeton University, May 1950.
[16] J. Nash, Non-cooperative games, Ann. Math. (1951) 286–295.
[17] R. Peretz, A. Schreiber, E. Schulte-Geers, The Lipschitz constant of perturbed anonymous games, CoRR, arXiv:2004.14741 [abs], 2020, https://arxiv.org/abs/2004.14741.
[18] M. Phillips, T. Lorenz, 'Dumb money' is on gamestop, and it's beating wall street at its own game, https://www.nytimes.com/2021/01/27/business/gamestop-wall-street-bets.html, Feb 2021. (Accessed 5 February 2021).
[19] A. Rubinstein, Settling the complexity of computing approximate two-player Nash equilibria, in: I. Dinur (Ed.), IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS, IEEE Computer Society, 2016, pp. 258–265.