Lipschitz Continuity and Approximate Equilibria

Argyrios Deligkas[1(✉)], John Fearnley[1], and Paul Spirakis[1,2]

[1] University of Liverpool, Liverpool, UK
a.deligkas@liverpool.ac.uk
[2] Computer Technology Institute (CTI), Patras, Greece

Abstract. In this paper, we study games with continuous action spaces and non-linear payoff functions. Our key insight is that Lipschitz continuity of the payoff function allows us to provide algorithms for finding approximate equilibria in these games. We begin by studying Lipschitz games, which encompass, for example, all concave games with Lipschitz continuous payoff functions. We provide an efficient algorithm for computing approximate equilibria in these games. Then we turn our attention to penalty games, which encompass biased games and games in which players take risk into account. Here we show that if the penalty function is Lipschitz continuous, then we can provide a quasi-polynomial time approximation scheme. Finally, we study distance biased games, where we present simple strongly polynomial time algorithms for finding best responses in L_1, L_2^2, and L_∞ biased games, and then use these algorithms to provide strongly polynomial algorithms that find 2/3, 5/7, and 2/3 approximations for these norms, respectively.

1 Introduction

Nash equilibria [18] are the central solution concept in game theory. However, recent advances have shown that computing an *exact* Nash equilibrium is PPAD-complete [8,9], and so there are unlikely to be polynomial time algorithms for this problem. The hardness of computing exact equilibria has lead to the study of *approximate* equilibria: while an exact equilibrium requires that all players have no incentive to deviate from their current strategy, an ϵ-approximate equilibrium requires only that their incentive to deviate is less than ϵ.

A fruitful line of work has developed studying the best approximations that can be found in polynomial-time for *bimatrix games*, which are two-player strategic form games. There, after a number of papers [5,10,11], the best known algorithm was given by Tsaknakis and Spirakis [21], who provide a polynomial time algorithm that finds a 0.3393-equilibrium. The existence of an FPTAS was ruled out by Chen, Deng, and Teng [8] unless PPAD = P. Recently, Rubinstein [20] proved that there is no PTAS for the problem, assuming the Exponential Time Hypothesis for PPAD. However, there is a *quasi-polynomial* approximation scheme given by Lipton, Markakis, and Mehta [16].

In a strategic form game, the game is specified by giving each player a finite number of strategies, and then specifying a table of payoffs that contains one

M. Gairing and R. Savani (Eds.): SAGT 2016, LNCS 9928, pp. 15–26, 2016.
DOI: 10.1007/978-3-662-53354-3_2

entry for every possible combination of strategies that the players might pick. The players are allowed to use mixed strategies, and so ultimately the payoff function is a convex combination of the payoffs given in the table. However, some games can only be modelled in a more general setting where the action spaces are continuous, or the payoff functions are non-linear.

For example, Rosen's seminal work [19] considered *concave games*, where each player picks a vector from a convex set. The payoff to each player is specified by a function that satisfies the following condition: if every other player's strategy is fixed, then the payoff to a player is a concave function over his strategy space. Rosen proved that concave games always have an equilibrium. A natural subclass of concave games, studied by Caragiannis, Kurokawa, and Procaccia [6], is the class of biased games. A biased game is defined by a strategic form game, a *base strategy* and a *penalty function*. The players play the strategic form game as normal, but they all suffer a penalty for deviating from their base strategy. This penalty can be a non-linear function, such as the L_2^2 norm.

In this paper, we study the computation of approximate equilibria in such games. Our main observation is that Lipschitz continuity of the players' payoff functions (with respect to changes in the strategy space) allows us to provide algorithms that find approximate equilibria. Several papers have studied how the Lipschitz continuity of the players' payoff functions affects the existence, the quality, and the complexity of the equilibria of the underlying game. Azrieli and Shmaya [1] studied many player games and derived bounds for the Lipschitz constant of the utility functions for the players that guarantees the existence of pure approximate equilibria for the game. We have to note though, that the games Azrieli and Shmaya study are significantly different from our games. In [1] the Lipschitz coefficient refers to the payoff function of player i as a function of $\mathbf{x}_{-i}$, i.e. when x_i is fixed. In this paper, the Lipschitz coefficient refers to the payoff function of player i as a function of x_i when the $\mathbf{x}_{-i}$ is fixed. We used this definition of the Lipschitz continuity in order to follow Rosen's definition of concave games that requires the payoff function of player i to be concave for every fixed strategy profile for the rest of the players. Daskalakis and Papadimitriou [12] proved that anonymous games posses pure approximate equilibria whose quality depends on the Lipschitz constant of the payoff functions and the number of pure strategies the players have and proved that these approximate equilibria can be computed in polynomial time. Furthermore, they gave a polynomial-time approximation scheme for anonymous games with many players and constant number of pure strategies. Babichenko [2] presented a best-reply dynamic for n-players Lipschitz anonymous games with two strategies which reaches an approximate pure equilibrium in $O(n \log n)$ steps. Deb and Kalai [13] studied how some variants of the Lipschitz continuity of the utility functions are sufficient to guarantee hindsight stability of equilibria.

1.1 Our Contribution

Lipschitz Games. We begin by studying a very general class of games, where each player's strategy space is continuous, and represented by a convex set of vectors, and where the only restriction is that the payoff function is Lipschitz continuous. This class is so general that exact equilibria, and even approximate equilibria may not exist. Nevertheless, we give an efficient algorithm that either outputs an ϵ-equilibrium, or determines that the game has no exact equilibria. More precisely, for M player games with a strategy space defined as the convex hull of n vectors, that have λ-Lipschitz continuous payoff functions in the L_p norm, for $p \geq 2$, and where $\gamma = \max \|\mathbf{x}\|_p$ over all $\mathbf{x}$ in the strategy space, we either compute an ϵ-equilibrium or determine that no exact equilibrium exists in time $O\left(Mn^{Mk+l}\right)$, where $k = O\left(\frac{\lambda^2 Mp\gamma^2}{\epsilon^2}\right)$ and $l = O\left(\frac{\lambda^2 p\gamma^2}{\epsilon^2}\right)$. Observe that this is a polynomial time algorithm when λ, p, γ, M, and ϵ are constant.

To prove this result, we utilize a recent result of Barman [4], which states that for every vector in a convex set, there is another vector that is ϵ close to the original in the L_p norm, and is a convex combination of b points on the convex hull, where b depends on p and ϵ, but does not depend on the dimension. Using this result, and the Lipschitz continuity of the payoffs, allows us to reduce the task of finding an ϵ-equilibrium to checking only a small number of strategy profiles, and thus we get a brute-force algorithm that is reminiscent of the QPTAS given by Lipton, Markakis, and Mehta for bimatrix games [16] and by the QPTAS of Babichenko, Barman, and Peretz [3] for many player games.

However, life is not so simple for us. Since we study a very general class of games, verifying whether a given strategy profile is an ϵ-equilibrium is a non-trivial task. It requires us to compute a *regret* for each player, which is the difference between the player's best response payoff and their actual payoff. Computing a best response in a bimatrix game is trivial, but for Lipschitz games, it may be a hard problem. We get around this problem by instead giving an algorithm to compute *approximate* best responses. Hence we find *approximate* regrets, and it turns out that this is sufficient for our algorithm to work.

Penalty Games. We then turn our attention to *penalty games*. In these games, the players play a strategic form game, and their utility is the payoff achieved in the game *minus* a penalty. The penalty function can be an arbitrary function that depends on the player's strategy. This is a general class of games that encompasses a number of games that have been studied before. The biased games studied by Caragiannis, Kurokawa, and Procaccia [6] are penalty games where the penalty is determined by the amount that a player deviates from a specified base strategy. The biased model was studied in the past by psychologists [22] and it is close to what they call *anchoring* [7,15]. In their seminal paper, Fiat and Papadimitriou [14] introduced a model for *risk prone* games, which resemble penalty games since the risk component can be encoded as a penalty. Mavronicolas and Monien [17] followed this line of research and provided results on the complexity of deciding if such games possess an equilibrium.

We again show that Lipschitz continuity helps us to find approximate equilibria. The only assumption that we make is that the penalty function is Lipschitz continuous in an L_p norm with $p \geq 2$. Again, this is a weak restriction, and it does not guarantee that exact equilibria exist. Even so, we give a quasi-polynomial time algorithm that either finds an ϵ-equilibrium, or verifies that the game has no exact equilibrium.

Our result can be seen as a generalisation of the QPTAS given by Lipton, Markakis, and Mehta [16] for bimatrix games. Their approach is to show the existence of an approximate equilibrium with a logarithmic support. They proved this via the probabilistic method: if we know an exact equilibrium of a bimatrix game, then we can take logarithmically many samples from the strategies, and playing the sampled strategies uniformly will be an approximate equilibrium with positive probability. We take a similar approach, but since our games are more complicated, our proof is necessarily more involved. In particular, for Lipton, Markakis, and Mehta, proving that the sampled strategies are an approximate equilibrium only requires showing that the expected payoff is close to the best response payoff. In penalty games, best response strategies are not necessarily pure, and so the events that we must consider are more complex.

Distance Biased Games. Finally, we consider distance biased games, which form a subclass of penalty games that have been studied recently by Caragiannis, Kurokawa, and Procaccia [6]. They showed that, under very mild assumptions on the bias function, biased games always have an exact equilibrium. Furthermore, for the case where the bias function is either the L_1 norm, or the L_2^2 norm, they give an exponential time algorithm for finding an exact equilibrium.

Our results for penalty games already give a QPTAS for biased games, but we are also interested in whether there are polynomial-time algorithms that can find non-trivial approximations. We give a positive answer to this question for games where the bias is the L_1 norm, the L_2^2 norm, or the L_∞ norm. We follow the well-known approach of Daskalakis, Mehta, Papadimitriou [11], who gave a simple algorithm for finding a 0.5-approximate equilibrium in a bimatrix game.

We show that this algorithm also works for biased games, although the generalisation is not entirely trivial. Again, this is because best responses cannot be trivially computed in biased games. For the L_1 and L_∞ norms, best responses can be computed via linear programming, and for the L_2^2 norm, best responses can be formulated as a quadratic program, and it turns out that this particular QP can be solved in polynomial time by the ellipsoid method. However, none of these algorithms are strongly polynomial. We show that, for each of the norms, best responses can be found by a simple strongly-polynomial combinatorial algorithm. We then analyse the quality of approximation provided by the technique of Daskalakis, Mehta, Papadimitriou [11]. We obtain a strongly polynomial algorithm for finding a 2/3 approximation in L_1 and L_∞ biased games, and a strongly polynomial algorithm for finding a 5/7 approximation in L_2^2 biased games. For the latter result, in the special case where the bias function is the inner product of the player's strategy we find a 13/21 approximation.

2 Preliminaries

We start by fixing some notation. For each positive integer n we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$, we use Δ^n to denote the $(n-1)$-dimensional simplex, and $\|x\|_p^q$ to denote the (p, q)-norm of a vector $x \in \mathbb{R}^d$, i.e. $\|x\|_p^q = (\sum_{i \in [d]} |x_i|^p)^{q/p}$. When $q = 1$, then we will omit it for notation simplicity. Given a set $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$, we use $conv(X)$ to denote the convex hull of X. A vector $y \in conv(X)$ is said to be k-uniform with respect to X if there exists a size k multiset S of $[n]$ such that $y = \frac{1}{k}\sum_{i \in S} x_i$. When X is clear from the context we will simply say that a vector is k uniform without mentioning that uniformity is with respect to X.

Games and Strategies. A game with M players can be described by a set of available actions for each player and a utility function for each player that depends both on his chosen action and the actions the rest of the players chose. For each player $i \in [M]$ we use S_i to denote his set of available actions and we call it his *strategy space*. We will use $x_i \in S_i$ to denote a specific action chosen by player i and we will call it the *strategy* of player i, we use $\mathbf{x} = (x_1, \ldots, x_M)$ to denote a *strategy profile* of the game, and we will use $\mathbf{x}_{-i}$ to denote the strategy profile where the player i is excluded, i.e. $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_M)$. We use $T_i(x_i, \mathbf{x}_{-i})$ to denote the utility of player i when he plays the strategy x_i and the rest of the players play according to the strategy profile $\mathbf{x}_{-i}$. A strategy $\hat{x}_i$ is a *best response* against the strategy profile $\mathbf{x}_{-i}$, if $T_i(\hat{x}_i, \mathbf{x}_{-i}) \geq T_i(x_i, \mathbf{x}_{-i})$ for all $x_i \in S_i$. The *regret* player i suffers under a strategy profile $\mathbf{x}$ is the difference between the utility of his best response and his utility under $\mathbf{x}$, i.e. $T_i(\hat{x}_i, \mathbf{x}_{-i}) - T_i(x_i, \mathbf{x}_{-i})$.

An $n \times n$ bimatrix game is a pair (R, C) of two $n \times n$ matrices: R gives payoffs for the *row* player and C gives the payoffs for the *column* player. We make the standard assumption that all payoffs lie in the range $[0, 1]$. If $\mathbf{x}$ and $\mathbf{y}$ are mixed strategies for the row and the column player, respectively, then the expected payoff for the row player under strategy profile $(\mathbf{x}, \mathbf{y})$ is given by $\mathbf{x}^T R\mathbf{y}$ and for the column player by $\mathbf{x}^T C\mathbf{y}$.

λ_p-Lipschitz Games. We will use the notion of the λ_p-Lipschitz continuity.

Definition 1 (λ_p-Lipschitz). A function $f : A \to \mathbb{R}$, with $A \subseteq \mathbb{R}^d$ is λ_p-Lipschitz continuous if for every x and y in A, it is true that $|f(x) - f(y)| \leq \lambda \cdot \|x - y\|_p$.

We call the game $\mathfrak{L} := (M, n, \lambda, p, \gamma, \mathcal{T})$ λ_p-*Lipschitz* if for each player $i \in [M]$ the strategy space S_i is the convex hull of n vectors $y_1, \ldots, y_n$ in $\mathbb{R}^d$, $\max_{x_i \in S_i} \|x_i\|_p \leq \gamma$, and the utility function $T_i(\mathbf{x}) \in \mathcal{T}$ is λ_p-Lipschitz continuous.

Two-Player Penalty Games. A two-player penalty game $\mathcal{P}$ is defined by a tuple $\big(R, C, \mathfrak{f}_r(\mathbf{x}), \mathfrak{f}_c(\mathbf{y})\big)$, where (R, C) is a bimatrix game and $\mathfrak{f}_r(\mathbf{x})$ and $\mathfrak{f}_c(\mathbf{y})$ are the penalty functions for the row and the column player respectively. The utilities for the players under a strategy profile $(\mathbf{x}, \mathbf{y})$, denoted by $T_r(\mathbf{x}, \mathbf{y})$ and $T_c(\mathbf{x}, \mathbf{y})$, are given by $T_r(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T R\mathbf{y} - \mathfrak{f}_r(\mathbf{x})$ and $T_c(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T C\mathbf{y} - \mathfrak{f}_c(\mathbf{y})$. We will use $\mathcal{P}_{\lambda_p}$ to denote the set of two-player penalty games with λ_p-Lipschitz penalty functions. A special class of penalty games is obtained when $\mathfrak{f}_r(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$ and $\mathfrak{f}_c(\mathbf{y}) = \mathbf{y}^T\mathbf{y}$. We call these games as *inner product* penalty games.

Two-Player Biased Games. This is a subclass of penalty games, where extra constraints are added to the penalty functions $\mathfrak{f}_r(\mathbf{x})$ and $\mathfrak{f}_c(\mathbf{y})$ of the players. In this class of games there is a *base strategy* and for each player and the penalty they receive is increasing with the distance between the strategy they choose and their base strategy. Formally, the row player has a base strategy $\mathbf{p} \in \Delta^n$, the column player has a base strategy $\mathbf{q}$ and their strictly increasing penalty functions are defined as $\mathfrak{f}_r(\|\mathbf{x} - \mathbf{p}\|_t^s)$ and $\mathfrak{f}_c(\|\mathbf{y} - \mathbf{q}\|_m^l)$ respectively.

Two-Player Distance Biased Games. This is a special class of biased games where the penalty function is a fraction of the distance between the base strategy of the player and his chosen strategy. Formally, a two player distance biased game $\mathcal{B}$ is defined by a tuple $\big(R, C, \mathfrak{b}_r(\mathbf{x}, \mathbf{p}), \mathfrak{b}_c(\mathbf{y}, \mathbf{q}), d_r, d_c\big)$, where (R, C) is a bimatrix game, $\mathbf{p} \in \Delta^n$ is a base strategy for the row player, $\mathbf{q} \in \Delta^n$ is a base strategy for the column player, $\mathfrak{b}_r(\mathbf{x}, \mathbf{p}) = \|\mathbf{x} - \mathbf{p}\|_t^s$ and $\mathfrak{b}_c(\mathbf{y}, \mathbf{q}) = \|\mathbf{y} - \mathbf{q}\|_m^l$ are the penalty functions for the row and the column player respectively. The utilities for the players under a strategy profile $(\mathbf{x}, \mathbf{y})$, denoted by $T_r(\mathbf{x}, \mathbf{y})$ and $T_c(\mathbf{x}, \mathbf{y})$, are given by $T_r(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T R\mathbf{y} - d_r \cdot \mathfrak{b}_r(\mathbf{x}, \mathbf{p})$ and $T_c(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T C\mathbf{y} - d_c \cdot \mathfrak{b}_c(\mathbf{y}, \mathbf{q})$, where d_r and d_c are non negative constants.

Solution Concepts. A strategy profile is an equilibrium if no player can increase his utility by unilaterally changing his strategy. A relaxed version of this concept is the approximate equilibrium, or ϵ-equilibrium, in which no player can increase his utility more than ϵ by unilaterally changing his strategy. Formally, a strategy profile $\mathbf{x}$ is an ϵ-equilibrium in a game $\mathfrak{L}$ if for every player $i \in [M]$ it holds that $T_i(x_i, \mathbf{x}_{-i}) \geq T_i(x_i', \mathbf{x}_{-i}) - \epsilon$ for all $x_i' \in S_i$.

In [20] it was proven that, unless $\mathtt{P} = \mathtt{PPAD}$, there is no PTAS for computing an ϵ-NE in bimatrix games. The same result holds for the class of penalty games where the penalty functions $\mathfrak{f}(n, \mathbf{x})$ for the players depend on n, the size of the underlying bimatrix game, and $\lim_{n\to\infty} \mathfrak{f}(n, \mathbf{x}) = 0$ for every player, for every possible $\mathbf{x}$. Let $\mathcal{P}'$ denote this class of games.

Theorem 1. *Unless* $\mathtt{P} = \mathtt{PPAD}$*, there is no PTAS for computing an* ϵ*-equilibrium in penalty games in* $\mathcal{P}'$*.*

3 Approximate Equilibria in λ_p-Lipschitz Games

In this section, we give an algorithm for computing approximate equilibria in λ_p Lipschitz games. Note that, our definition of a λ_p-Lipschitz game does not guarantee that an equilibrium always exists. Our technique can be applied *irrespective* of whether an exact equilibrium exists. If an exact equilibrium does exist, then our technique will always find an ϵ-equilibrium. If an exact equilibrium does not exist, then our algorithm either finds an ϵ-equilibrium or reports that the game does not have an exact equilibrium.

We will utilize the following theorem that was recently proved by Barman [4].

Theorem 2 (Barman [4]). *Given a set of vectors* $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$*, let* $conv(X)$ *denote the convex hull of* X*. Furthermore, let* $\gamma := \max_{x\in X} \|x\|_p$ *for some* $2 \leq p < \infty$*. For every* $\epsilon > 0$ *and every* $\mu \in conv(X)$*, there exists an* $\frac{4p\gamma^2}{\epsilon^2}$ *uniform vector* $\mu' \in conv(X)$ *such that* $\|\mu - \mu'\|_p \leq \epsilon$*.*

Combining Theorem 2 with the Definition 1 we get the following lemma.

Lemma 1. *Let $X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^d$, let $f : conv(X) \to \mathbb{R}$ be a λ_p-Lipschitz continuous function for some $2 \le p < \infty$, let $\epsilon > 0$ and let $k = \frac{4\lambda^2 p\gamma^2}{\epsilon^2}$, where $\gamma := \max_{x \in X} \|x\|_p$. Furthermore, let $f(\mathbf{x}^*)$ be the optimum value of f. Then we can compute a k-uniform point $\mathbf{x}' \in conv(X)$ in time $O(n^k)$, such that $|f(\mathbf{x}^*) - f(\mathbf{x}')| < \epsilon$.*

We now prove our result about Lipschitz games. In what follows we will study a λ_p-Lipschitz game $\mathfrak{L} := (M, n, \lambda, p, \gamma, \mathcal{T})$. Assuming the existence of an exact Nash equilibrium, we establish the existence of a k-uniform approximate equilibrium in the game $\mathfrak{L}$, where k depends on M, λ, p and γ. Note that λ depends heavily on p and the utility functions for the players.

Since by the definition of λ_p-Lipschitz games the strategy space S_i for every player i is the convex hull of n vectors $y_1, \dots, y_n$ in $\mathbb{R}^d$, any $x_i \in S_i$ can be written as a convex combination of y_js. Hence, $x_i = \sum_{j=1}^n \alpha_j y_j$, where $\alpha_j > 0$ for every $j \in [n]$ and $\sum_{j=1}^n \alpha_j = 1$. Then, $\alpha = (\alpha_1, \dots, \alpha_n)$ is a probability distribution over the vectors $y_1, \dots, y_n$, i.e. vector y_j is drawn with probability α_j. Thus, we can sample a strategy x_i by the probability distribution α.

So, let $\mathbf{x}^*$ be an equilibrium for $\mathfrak{L}$ and let $\mathbf{x}'$ be a sampled uniform strategy profile from $\mathbf{x}^*$. For each player i we define the following events

$$\begin{aligned}
\phi_i &= \left\{|T_i(x'_i, \mathbf{x}'_{-i}) - T_i(x^*_i, \mathbf{x}^*_{-i})| < \epsilon/2\right\} \\
\pi_i &= \left\{T_i(x_i, \mathbf{x}'_{-i}) < T_i(x'_i, \mathbf{x}'_{-i}) + \epsilon\right\} \quad \text{for all possible } x_i \\
\psi_i &= \left\{\|x'_i - x^*_i\|_p < \frac{\epsilon}{2M\lambda}\right\} \quad \text{for some } p > 1.
\end{aligned}$$

Notice that if all the events π_i occur at the same time, then the sampled profile $\mathbf{x}'$ is an ϵ-equilibrium. We will show that if for a player i the events ϕ_i and $\bigcap_j \psi_j$ hold, then the event π_i is also true.

Lemma 2. *For all $i \in [M]$ it holds that $\bigcap_{j \in [M]} \psi_j \cap \phi_i \subseteq \pi_i$.*

We are ready to prove the main result of the section.

Theorem 3. *In any λ_p-Lipschitz game $\mathfrak{L}$ that possess an equilibrium and any $\epsilon > 0$, there is a k-uniform strategy profile, with $k = \frac{16M^2\lambda^2 p\gamma^2}{\epsilon^2}$ that is an ϵ-equilibrium.*

Theorem 3 establishes the existence of a k-uniform approximate equilibrium, but this does not immediately give us our approximation algorithm. The obvious approach is to perform a brute force check of all k-uniform strategies, and then output the one that provides the best approximation. There is a problem with this, however, since computing the quality of approximation requires us to compute the regret for each player, which in turn requires us to compute a best response for each player. Computing an exact best response in a Lipschitz game is a hard problem in general, since we make no assumptions about the utility functions of the players. Fortunately, it is sufficient to instead compute an *approximate* best response for each player, and Lemma 1 can be used to do this.

Lemma 3. *Let* $\mathbf{x}$ *be a strategy profile for a* λ_p*-Lipschitz game* $\mathfrak{L}$*, and let* $\hat{x}_i$ *be a best response for player* i *against the profile* $\mathbf{x}_{-i}$*. There is a* $\frac{4\lambda^2 p\gamma^2}{\epsilon^2}$*-uniform strategy* x'_i *that is an* ϵ*-best response against* $\mathbf{x}_{-i}$*.*

Our goal is to *approximate* the approximation guarantee for a given strategy profile. More formally, given a strategy profile $\mathbf{x}$ that is an ϵ-equilibrium, and a constant $\delta > 0$, we want an algorithm that outputs a number within the range $[\epsilon - \delta, \epsilon + \delta]$. Lemma 3 allows us to do this. For a given strategy profile $\mathbf{x}$, we first compute δ-approximate best responses for each player, then we can use these to compute δ-approximate regrets for each player. The maximum over the δ-approximate regrets then gives us an approximation of ϵ with a tolerance of δ. This is formalised in the following algorithm.

Algorithm 1. Evaluation of approximation guarantee

Input: A strategy profile $\mathbf{x}$ for $\mathfrak{L}$, and a constant $\delta > 0$.
Output: An additive δ-approximation of the approximation guarantee $\alpha(\mathbf{x})$ for the strategy profile $\mathbf{x}$.

1. Set $l = \frac{4\lambda^2 p\gamma^2}{\delta^2}$.
2. For every player $i \in [M]$
   (a) For every l-uniform strategy x'_i of player i compute $T_i(x'_i, \mathbf{x}_{-i})$.
   (b) Set $m^* = \max_{x'_i} T_i(x'_i, \mathbf{x}_{-i})$.
   (c) Set $\mathcal{R}_i(\mathbf{x}) = m^* - T_i(x_i, \mathbf{x}_{-i})$.
3. Set $\alpha(\mathbf{x}) = \delta + \max_{i\in[M]} \mathcal{R}_i(\mathbf{x})$.
4. Return $\alpha(\mathbf{x})$.

Utilising the above algorithm, we can now produce an algorithm to find an approximate equilibrium in Lipschitz games. The algorithm checks all k-uniform strategy profiles, using the value of k given by Theorem 3, and for each one, computes an approximation of the quality approximation using the algorithm given above.

Algorithm 2. 3ϵ-equilibrium for λ_p-Lipschitz game $\mathfrak{L}$

Input: Game $\mathfrak{L}$ and $\epsilon > 0$.
Output: An 3ϵ-equilibrium for $\mathfrak{L}$.

1. Set $k > \frac{16\lambda^2 Mp\gamma^2}{\epsilon^2}$.
2. For every k-uniform strategy profile $\mathbf{x}'$
   (a) Compute an ϵ-approximation of $\alpha(\mathbf{x}')$.
   (b) If the ϵ-approximation of $\alpha(\mathbf{x}')$ is less than 2ϵ, return $\mathbf{x}'$.

If the algorithm returns a strategy profile $\mathbf{x}$, then it must be a 3ϵ equilibrium. This is because we check that an ϵ-approximation of $\alpha(\mathbf{x})$ is less than 2ϵ, and therefore $\alpha(\mathbf{x}) \leq 3\epsilon$. Secondly, we argue that if the game has an exact Nash equilibrium, then this procedure will always output a 3ϵ-approximate equilibrium.

From Theorem 3 we know that if $k > \frac{16\lambda^2 M p \gamma^2}{\epsilon^2}$, then there is a k-uniform strategy profile $\mathbf{x}$ that is an ϵ-equilibrium for $\mathfrak{L}$. When we apply our approximate regret algorithm to $\mathbf{x}$, to find an ϵ-approximation of $\alpha(\mathbf{x})$, the algorithm will return a number that is less than 2ϵ, hence $\mathbf{x}$ will be returned by the algorithm.

To analyse the running time, observe that there are $\binom{n+k-1}{k} = O(n^k)$ possible k-uniform strategies for each player, thus $O(n^{Mk})$ k-uniform strategy profiles. Furthermore, our regret approximation algorithm runs in time $O(Mn^l)$, where $l = \frac{4\lambda^2 p \gamma^2}{\epsilon^2}$. Hence, we get the next theorem.

Theorem 4. *Given a λ_p-Lipschitz game $\mathfrak{L}$ that possess an equilibrium and any $\epsilon > 0$, a 3ϵ-equilibrium can be computed in time $O\left(Mn^{Mk+l}\right)$, where $k = O\left(\frac{\lambda^2 M p \gamma^2}{\epsilon^2}\right)$ and $l = O\left(\frac{\lambda^2 p \gamma^2}{\epsilon^2}\right)$.*

Although it might be hard to decide whether a game has an equilibrium, our algorithm can be applied in *any* λ_p-Lipschitz game. Notice that our algorithm never uses the fact that the game possess an equilibrium. If the game does not posses an exact equilibrium then our algorithm either finds an approximate equilibrium or determines that the game does not posses an exact equilibrium.

Theorem 5. *For any λ_p-Lipschitz game $\mathfrak{L}$ in time $O\left(Mn^{Mk+l}\right)$, we can either compute a 3ϵ-equilibrium, or decide that $\mathfrak{L}$ does not posses an exact equilibrium, where $k = O\left(\frac{\lambda^2 M p \gamma^2}{\epsilon^2}\right)$ and $l = O\left(\frac{\lambda^2 p \gamma^2}{\epsilon^2}\right)$.*

4 A Quasi-polynomial Algorithm for Penalty Games

In this section we present an algorithm that, for any $\epsilon > 0$, can compute an ϵ-equilibrium for any penalty game in $\mathcal{P}_{\lambda_p}$ that posses one in quasi-polynomial time. For the algorithm, we take the same approach as we did in the previous section for Lipschitz games: we show that if an exact equilibrium exists, then a k-uniform approximate equilibrium always exists too, and provide a brute-force search algorithm for finding it. Once again, since best response computation may be hard for this class of games, we must provide an approximation algorithm for finding the quality of an approximate equilibrium.

We first focus on penalty games that posses an exact equilibrium. So, let $(\mathbf{x}^*, \mathbf{y}^*)$ be an equilibrium of the game and let $(\mathbf{x}', \mathbf{y}')$ be a k-uniform strategy profile sampled from this equilibrium. We define the following four events:

$$\begin{aligned}
\phi_r &= \left\{ |T_r(\mathbf{x}', \mathbf{y}') - T_r(\mathbf{x}^*, \mathbf{y}^*)| < \epsilon/2 \right\} \\
\pi_r &= \left\{ T_r(\mathbf{x}, \mathbf{y}') < T_r(\mathbf{x}', \mathbf{y}') + \epsilon \right\} \qquad \text{for all } \mathbf{x} \\
\phi_c &= \left\{ |T_c(\mathbf{x}', \mathbf{y}') - T_c(\mathbf{x}^*, \mathbf{y}^*)| < \epsilon/2 \right\} \\
\pi_c &= \left\{ T_c(\mathbf{x}', \mathbf{y}) < T_c(\mathbf{x}', \mathbf{y}') + \epsilon \right\} \qquad \text{for all } \mathbf{y}.
\end{aligned}$$

The goal is to derive a value for k such that all the four events above are true, or equivalently $Pr(\phi_r \cap \pi_r \cap \phi_c \cap \pi_r) > 0$.

Note that in order to prove that $(\mathbf{x}', \mathbf{y}')$ is an ϵ-equilibrium we *only* have to consider the events π_r and π_c. Nevertheless, as we show in Lemma 4, the events

ϕ_r and ϕ_c are crucial in our analysis. The proof of the main theorem boils down to the events ϕ_r and ϕ_c.

We will focus only on the row player, since the same analysis can be applied to the column player. Firstly we study the event π_r.

Lemma 4. *For all penalty games it holds that* $Pr(\pi_r^c) \leq n \cdot e^{-\frac{k\epsilon^2}{2}} + Pr(\phi_r^c)$.

With Lemma 4 in hand, we can see that in order to compute a value for k it is sufficient to study the event ϕ_r. We introduce the following auxiliary events that we will study separately: $\phi_{ru} = \left\{|\mathbf{x}'^T R\mathbf{y}' - \mathbf{x}^{*^T} R\mathbf{y}^*| < \epsilon/4\right\}$ and $\phi_{r\mathfrak{b}} = \left\{|\mathfrak{f}_r(\mathbf{x}') - \mathfrak{f}_r(\mathbf{x}^*)| < \epsilon/4\right\}$. It is easy to see that if both $\phi_{r\mathfrak{b}}$ and ϕ_{ru} are true, then the event ϕ_r must be true too. So we have $\phi_{r\mathfrak{b}} \cap \phi_{ru} \subseteq \phi_r$. Using the analysis from [16] we can prove that $Pr(\phi_{ru}^c) \leq 2e^{-\frac{k\epsilon^2}{8}}$. Finally, we must prove an upper bound on the event $\phi_{r\mathfrak{b}}^c$, which we provide in the following lemma.

Lemma 5. $\Pr(\phi_{r\mathfrak{b}}^c) \leq \frac{8\lambda\sqrt{p}}{\epsilon\sqrt{k}}$.

Let us define the event $GOOD = \phi_r \cap \phi_c \cap \pi_r \cap \pi_c$. To prove our theorem it suffices to prove that $Pr(GOOD) > 0$. Notice that for the events ϕ_c and π_c the same analysis as for ϕ_r and π_r can be used. Then, using Lemmas 4, 5 and the analysis for ϕ_{ru} we get that $Pr(GOOD^c) < 1$ for the chosen value of k.

Theorem 6. *For any equilibrium* $(\mathbf{x}^*, \mathbf{y}^*)$ *of a penalty game from the class* $\mathcal{P}_{\lambda_p}$*, any* $\epsilon > 0$*, and any* $k \in \frac{\Omega(\lambda^2 \log n)}{\epsilon^2}$*, there exists a k-uniform strategy profile* $(\mathbf{x}', \mathbf{y}')$ *that:*

1. $(\mathbf{x}', \mathbf{y}')$ *is an* ϵ*-equilibrium for the game,*
2. $|T_r(\mathbf{x}', \mathbf{y}') - T_r(\mathbf{x}^*, \mathbf{y}^*)| < \epsilon/2$*,*
3. $|T_c(\mathbf{x}', \mathbf{y}') - T_c(\mathbf{x}^*, \mathbf{y}^*)| < \epsilon/2$*.*

Theorem 6 establishes the *existence* of a k-uniform strategy profile $(\mathbf{x}', \mathbf{y}')$ that is an ϵ-equilibrium, but as before, we must provide an efficient method for approximating the quality of approximation provided by a given strategy profile. To do so, we first give the following lemma, which shows that approximate best responses can be computed in quasi-polynomial time for penalty games.

Lemma 6. *Let* $(\mathbf{x}, \mathbf{y})$ *be a strategy profile for a penalty game* $\mathcal{P}_{\lambda_p}$*, and let* $\hat{\mathbf{x}}$ *be a best response against* $\mathbf{y}$*. There is an l-uniform strategy* $\mathbf{x}'$*, with* $l = \frac{17\lambda^2\sqrt{p}}{\epsilon^2}$*, that is an* ϵ*-best response against* $\mathbf{y}$*, i.e.* $T_r(\hat{\mathbf{x}}, \mathbf{y}) < T_r(\mathbf{x}', \mathbf{y}) + \epsilon$*.*

Given this lemma, we can reuse Algorithm 1, but with l set equal to $\frac{17\lambda^2\sqrt{p}}{\epsilon^2}$, to provide an algorithm that aproximates the quality of approximation of a given strategy profile. Then, we can reuse Algorithm 2 with $k = \frac{\Omega(\lambda^2 \log n)}{\epsilon^2}$ to provide a quasi-polynomial time algorithm that finds approximate equilibia in penalty games. Notice again that our algorithm can be applied in games in which it is computationally hard to verify whether an exact equilibrium exists. Our algorithm either will compute an approximate equilibrium or it will fail to find one, in which case the game does not posses an exact equilibrium.

Theorem 7. *In any penalty game $\mathcal{P}_{\lambda_p}$ and any $\epsilon > 0$, in quasi polynomial time we can either compute a 3ϵ-equilibrium, or decide that $\mathcal{P}_{\lambda_p}$ does not posses an exact equilibrium.*

5 Distance Biased Games

In this section, we focus on three particular classes of distance biased games, and we provide polynomial-time approximation algorithms when the penalty function is one of the L_1, L_2^2 and L_∞ norm. Our approach is to follow the technique of Daskalakis, Mehta, Papadimitriou [11] that finds a 0.5-NE in a bimatrix game. The algorithm that we will use for all three penalty functions is given below.

Algorithm 3. The Base Algorithm

1. Compute a best response $\mathbf{y}^*$ against $\mathbf{p}$.
2. Compute a best response $\mathbf{x}$ against $\mathbf{y}^*$.
3. Set $\mathbf{x}^* = \delta \cdot \mathbf{p} + (1 - \delta) \cdot \mathbf{x}$, for some $\delta \in [0, 1]$.
4. Return the strategy profile $(\mathbf{x}^*, \mathbf{y}^*)$.

While this is a well-known technique for bimatrix games, it cannot immediately be applied to penalty games, because the algorithm requires us to compute two best responses. While computing a best-response is trivial in bimatrix games, this is not the case for penalty games. Best responses for L_1 and L_∞ penalties can be computed in polynomial-time via linear programming, and for L_2^2 penalties, the ellipsoid algorithm can be applied to a specialized quadratic program. However, these methods work as black boxes and do not provide strongly polynomial algorithms.

For each of the penalties we develop a simple combinatorial algorithm for computing best response strategies. We use the nature of these penalty functions and we provide strongly polynomial algorithms that compute best responses. More specifically, for the L_1 and L_∞ norms we compute the exact probability each pure strategy should be played in a best response by studying how the utility function increases. For the L_2^2 norm we use the KKT conditions of a quadratic program to produce a closed formula for the solution. Our algorithms, which are strongly polynomial, allow us to optimize the value of δ, and produce the following approximation guarantees.

Theorem 8. *In biased games with L_1, L_2^2 and L_∞ penalties a 2/3, 5/7 and 2/3-equilibrium respectively can be computed in polynomial time. For inner product games the approximation guarantee is 13/21.*

References

1. Azrieli, Y., Shmaya, E.: Lipschitz games. Math. Oper. Res. **38**(2), 350–357 (2013)
2. Babichenko, Y.: Best-reply dynamics in large binary-choice anonymous games. Games Econ. Behav. **81**, 130–144 (2013)

3. Babichenko, Y., Barman, S., Peretz, R.: Simple approximate equilibria in large games. In: Proceeding of EC, pp. 753–770 (2014)
4. Barman, S.: Approximating Nash equilibria and dense bipartite subgraphs via an approximate version of Caratheodory's theorem. In: Proceeding of STOC 2015, pp. 361–369 (2015)
5. Bosse, H., Byrka, J., Markakis, E.: New algorithms for approximate Nash equilibria in bimatrix games. Theor. Comput. Sci. **411**(1), 164–173 (2010)
6. Caragiannis, I., Kurokawa, D., Procaccia, A.D.: Biased games. In: Proceeding of AAAI, pp. 609–615 (2014)
7. Chapman, G.B., Johnson, E.J.: Anchoring, activation, and the construction of values. Organ. Behav. Hum. Decis. Process. **79**(2), 115–153 (1999)
8. Chen, X., Deng, X., Teng, S.-H.: Settling the complexity of computing two-player Nash equilibria. J. ACM **56**(3), 14:1–14:57 (2009)
9. Daskalakis, C., Goldberg, P.W., Papadimitriou, C.H.: The complexity of computing a Nash equilibrium. SIAM J. Comput. **39**(1), 195–259 (2009)
10. Daskalakis, C., Mehta, A., Papadimitriou, C.H.: Progress in approximate Nash equilibria. In: Proceeding of EC, pp. 355–358 (2007)
11. Daskalakis, C., Mehta, A., Papadimitriou, C.H.: A note on approximate Nash equilibria. Theor. Comput. Sci. **410**(17), 1581–1588 (2009)
12. Daskalakis, C., Papadimitriou, C.H.: Approximate Nash equilibria in anonymous games. J. Econ. Theory (2014, to appear)
13. Deb, J., Kalai, E.: Stability in large Bayesian games with heterogeneous players. J. Econ. Theor. **157**(C), 1041–1055 (2015)
14. Fiat, A., Papadimitriou, C.H.: When the players are not expectation maximizers. In: SAGT, pp. 1–14 (2010)
15. Kahneman, D.: Reference points, anchors, norms, and mixed feelings. Organ. Behav. Hum. Decis. Process. **51**(2), 296–312 (1992)
16. Lipton, R.J., Markakis, E., Mehta, A.: Playing large games using simple strategies. In: EC, pp. 36–41 (2003)
17. Mavronicolas, M., Monien, B.: The complexity of equilibria for risk-modeling valuations. CoRR, abs/1510.08980 (2015)
18. Nash, J.: Non-cooperative games. Ann. Math. **54**(2), 286–295 (1951)
19. Rosen, J.B.: Existence and uniqueness of equilibrium points for concave n-person games. Econometrica **33**(3), 520–534 (1965)
20. Rubinstein, A.: Settling the complexity of computing approximate two-player nash equilibria. CoRR, abs/1606.04550 (2016)
21. Tsaknakis, H., Spirakis, P.G.: An optimization approach for approximate Nash equilibria. Internet Math. **5**(4), 365–382 (2008)
22. Tversky, A., Kahneman, D.: Judgment under uncertainty: heuristics and biases. Science **185**(4157), 1124–1131 (1974)

http://www.springer.com/978-3-662-53353-6

Algorithmic Game Theory
9th International Symposium, SAGT 2016, Liverpool, UK, September 19–21, 2016, Proceedings
Gairing, M.; Savani, R. (Eds.)
2016, XI, 347 p. 38 illus., Softcover
ISBN: 978-3-662-53353-6