Identify the Nash Equilibrium in Static Games with Random Payoffs

Yichi Zhou[1] Jialian Li[1] Jun Zhu[1]

Abstract

We study the problem on how to learn the pure Nash Equilibrium of a two-player zero-sum static game with random payoffs under unknown distributions via efficient payoff queries. We introduce a multi-armed bandit model to this problem due to its ability to find the best arm efficiently among random arms and propose two algorithms for this problem—LUCB-G based on the confidence bounds and a racing algorithm based on successive action elimination. We provide an analysis on the sample complexity lower bound when the Nash Equilibrium exists.

1. Introduction

We consider the static zero-sum game where two players are involved with finite pure strategies. From game theory, if both players use only pure strategies and the payoffs are distinct from each other, at most one pure Nash Equilibrium (NE) exists (Osborne & Rubinstein, 1994). We concentrate on the setting where all payoffs are random variables under some unknown distributions. Samples (or queries) can be obtained by submitting pure strategies of the two players and receiving the associated payoffs. Our target is to answer the questions: (1) whether there is a pure NE; and 2) how to identify it if exists using as few queries as possible.

Our motivation for this problem comes from the need of identifying NE in many practical competitive situations. Since NE is a fundamental concept in game theory and many other fields, the computational complexity needed for NE is of much interest. However, in practice we are often given access to the data generated from some practical phenomena, rather than a clear rule for the payoffs of the game. Hence, the *empirical game-theoretic analysis* (Wellman, 2006; Jordan et al., 2008) has received a lot of attention to estimate the practical games through simulation. In the empirical modeling, pure-strategy profiles of players are submitted to the game and we receive the associated payoffs. Fearnley et al. (2015) consider the process managed in an online manner by algorithms and analyze the complexity of these payoff-query algorithms. The main focus is on whether the query methods can figure out mixed Nash Equilibrium with only a fraction of profiles. Extensions have been made to obtain query complexity on approximate Nash Equilibrium (Babichenko, 2016), correlated equilibrium (Hart & Nisan, 2016), and well-supported approximate correlated equilibrium (Goldberg & Roth, 2016).

The above work is essentially a *revealed-payoff* search model (Jordan et al., 2008), where payoffs are deterministic and every profile only needs to be queried at most once. We concentrate on the *noisy-payoff* model (Jordan et al., 2008), where the received payoff of a query is a sample of an underlying distribution. This random payoff setting is more realistic in practice, where randomness naturally arises because of incomplete information, noise, or other stochastic factors in the world. A simple but well-known example is the coin flipping game, where two players throw a coin and guess its landing upper side. Since the physical process of coin landing can be determined by many noisy factors, the payoffs to the two players, which depend on the landing results, are random. This notable discrepancy leads to different algorithms and complexity bounds for the two models, since for noisy-payoff models, more queries are needed for any profile to get an estimated payoff near its expectation value with high probability and the additional computational cost can take a dominating role in complexity. Previous work has explored different methods for noisy payoff models, such as interleaving the samples (Walsh et al., 2003) and using regression for payoffs (Vorobeychik et al., 2007). This paper turns to bandit models, a relatively natural direction from the view of online learning, since query methods themselves hold a sequential property. Put another way, we select a strategy to submit based on previous observations at each round of query methods.

This task can be viewed as a variant of best arm identification (BAI) in the literature of multi-armed bandits (Jamieson & Nowak, 2014), where different pure strategy profiles are regarded as arms. The classical BAI problem is to identify which arm is the one with the highest mean.

[1] Dept. of Comp. Sci. & Tech., TNList Lab, State Key Lab for Intell. Tech. & Systems, CBICR Center, Tsinghua University. Correspondence to: Jun Zhu <dcszj@tsinghua.edu.cn>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

There are two basic settings for BAI problem—fixed budget and fixed confidence (Kaufmann et al., 2015). In this paper, we focus on the fixed confidence setting, where the purpose of an algorithm is to identify the best arm with a fixed probability by as few pulls (queries) as possible.

Contributions: We study both the sample complexity lower bound and algorithms in the fixed confidence setting for two-player zero-sum static games with random payoffs.

In Section 3, we discuss the sample complexity lower bound for the case NE exists. Previous proofs on the lower bound for BAI all rely on changes of distributions (Audibert & Bubeck, 2010; Kaufmann et al., 2015; Mannor & Tsitsiklis, 2004)—changes on a single arm can change the best arm in the bandit model. For our problem, we prove the lower bound for the arms in the same row or same column with the NE by similar techniques. However, this approach does not work for those arms that are in neither the same row nor column with the NE, since changing the distribution of such an arm does not change the NE (details are in Section 3). To get the lower bound on these arms, we rephrase the arm selection as a hypothesis testing problem and use the minimax techniques for hypothesis testing (Tsybakov, 2009) to get the bound.

There are two types of algorithms for the BAI problem in the fixed confidence setting (Jamieson & Nowak, 2014)—based on either confidence bound (Kalyanakrishnan et al., 2012) or successive eliminations on suboptimal arms (Maron & Moore, 1997). In Section 4, we propose two corresponding algorithms to identify NE for our problem in the fixed confidence setting. The first algorithm has a provable bound on sample complexity, and we show that the second one will stop in a finite number of time steps with probability at least $1 - \delta$.

Related work: Garivier et al. (2016) also study the two-player zero-sum game with random payoffs. They consider the case that each player selects her strategy one-by-one, while we focus on the case that both players select their strategies simultaneously. Our setting is suitable for the case that each player chooses actions independently. Many games are static, such as the tai sai game where players independently guess the range of the outputs of three dices.

Much work has been done on Nash Equilibrium. Daskalakis et al. (2009) shows that it is computationally hard to recognize exact Nash Equilibrium, even for the simplest two-person game (Chen et al., 2009). As empirical game-theoretical analysis (Wellman, 2006) is proposed, Fearnley et al. (2015) studies the payoff-query algorithms and considers the query complexity as a criterion for computational complexity. Following work (Babichenko, 2016; Goldberg & Roth, 2016) has extended this criterion to some other approximate equilibrium. Query bounds for NE have also been given on specific games such as two-strategy anonymous games (Goldberg & Turchetta, 2015) and bimatrix games (Fearnley & Savani, 2016).

2. Preliminaries

We start by presenting the basic settings, notations and assumptions that will be used in the sequel.

2.1. Basic settings

Two-player zero-sum static game with random payoffs: A static game is a model in which all players choose their strategies once and simultaneously. A two-player zero-sum game involves two players 1 and 2, and each player chooses her own strategy s_i from a strategy set $S_i, i \in \{1, 2\}$. After decisions are made, player 1 gets payoff rew_{s_1,s_2} and player 2 gets $-rew_{s_1,s_2}$. Each player tries to maximize her payoff. The game can be represented by a $m \times n$-matrix where $m = |S_1|, n = |S_2|$. If $rew_{i,j}$ is a deterministic value for any i, j, it is direct to identify the NE (i^*, j^*) which has the minimum value in row i^* and the maximum value in column j^*. We consider the more practical games with random payoffs, whose distributions are unknown. This makes the identification of NE difficult and hence we employ query methods to learn the NE empirically. In each query, the algorithm generates a pure strategy (s_1, s_2), and the environment returns the associated payoff. Our target is to determine whether the Nash Equilibrium (NE) exists[1] and what it is if it does exist with as few queries as possible.

Multi-armed bandits: In a bandit model, an agent is facing a set of actions (or arms), and needs to select one arm to pull every time. In our case, an arm is specified by $s \in [m] \times [n]$, where $[m]$ denotes the set $\{1, \cdots, m\}$. Successive pulls of an arm (i, j) yield a sequence of observations (or rewards) $Y_1^{i,j}, Y_2^{i,j}, \cdots$. A policy $I = \{I_t : t \in \mathbb{N}_+\}$ denotes a sequence of random variables, where the variable $I_t \in [m] \times [n]$ indicates which arm to pull at time step t.

The classical best arm identification (BAI) problem is to identify which arm is the one with the highest mean. There are two basic settings for the BAI problem—fixed budget and fixed confidence (Kaufmann et al., 2015). In this paper, we focus on the identification of the NE of a static game or detect its absence in the fixed confidence setting. That is, we aim to identify the correct NE with probability at least $1 - \delta$ with efficient sampling, where $\delta \in (0, 1)$ is the confidence parameter. Algorithms satisfying this requirement are known as δ-PAC algorithms (Kaufmann et al., 2015).

Following Kaufmann et al. (2015), a practical BAI algorithm in the fixed confidence setting typically consists of:

- **Policy**: given a sequence of past observations, a policy determines which arms to pull.
- **Stopping rule**: a stopping rule can be described as

[1] We focus on the NE of pure strategy. So NE may not exist.

a series of observation sets $\mathcal{F}_t, t \in \mathbb{N}_+$. When an element $o \in \mathcal{F}_t$ is observed, the policy stops sampling.

- **Recommendation rule**: a recommendation rule is usually to recommend the best arm.

As we shall see, in our problem the best arm may not exist, when the sampling process stops, the recommendation rule determines whether the NE exists or not, and if exists, it determines which is the NE.

2.2. Basic assumptions and notations

Let $\mathbb{P}_{i,j} \in \mathcal{P}$ be the underlying distribution of arm (i, j) and $\mathcal{P}$ is a set of probability measures. For an arm s, we use μ_s to denote the expectation of $\mathbb{P}_s$. And let $\bar{\mu}_s(t)$ denote the empirical mean of s at time step t, we omit t for simplicity when there is no ambiguity . Here, we consider pulling an arm once as a time step, that is, time step t means that we have pulled arms for t times. We use $\bar{\mathbf{M}}_t$ to denote the empirical matrix with entry (i, j) representing the empirical mean of $\mathbb{P}_{i,j}$ at time step t. For $s = (s_1, s_2)$, we define $row(s) = \{s' = (s'_1, s'_2) : s'_1 = s_1\}$, $col(s) = \{s' = (s'_1, s'_2) : s'_2 = s_2\}$, $nei(s) = row(s) \cup col(s)$, and $s[1] = s_1, s[2] = s_2$.

Let $NE(\mathbf{M})$ denote the NE of matrix $\mathbf{M}$. Formally, $NE(\mathbf{M}) = s$ if there is an arm s such that $\mu_s = \min_{s' \in row(s)} \mu_{s'}$ and $\mu_s = \max_{s' \in col(s)} \mu_{s'}$; otherwise if there is no such arm, we denote $NE(\mathbf{M}) = none$ to show that no Nash Equilibrium exists. Specifically, we use $\mathbb{M}$ to denote the matrix whose entry (i, j) is the expectation of distribution $\mathbb{P}_{i,j}$. Our target is to identify $NE(\mathbb{M})$. For convenience, let $s^* = NE(\mathbb{M})$. Let R_i, C_j be the sets of the arms corresponding to the i-th row and the j-th column of $\mathbb{M}$ respectively. Our lower bound mainly focuses on the case that the NE exists (i.e., $NE(\mathbb{M}) \neq none$) and our algorithms are δ-PAC. We assume that the expectations of the arms are mutually different.

In the proof of the lower bound, it is natural to make $\mathcal{P}$ abundant enough to include various continuous distributions while ruling out some extreme situations where distributions are not mutually absolutely continuous. So we assume that $\mathcal{P}$ consists of parametric distributions continuously parameterized by their means. This assumption has been widely used in studying multi-armed bandits (Lai & Robbins, 1985; Kaufmann et al., 2015).

Assumption 1. *For all $p, q \in \mathcal{P}$ such that $p \neq q$, for all $\alpha > 0$:*

- $\exists q_1 \in \mathcal{P}$, $KL(p, q) < KL(p, q_1) < KL(p, q) + \alpha$, $\mathbb{E}q_1 > \mathbb{E}q > \mathbb{E}p$.
- $\exists q_2 \in \mathcal{P}$, $KL(p, q) < KL(p, q_2) < KL(p, q) + \alpha$, $\mathbb{E}q_2 < \mathbb{E}q < \mathbb{E}p$.

Here $KL(q, p)$ is the KL-divergence. Many distributions are included in $\mathcal{P}$, such as the broad class of one-parameter exponential family distributions.

3. Lower bound

Let $N_\delta(s)$ denote the number of pulls on an arm s by a δ-PAC algorithm. For arm $s \in nei(s^*)$, we provide a lower bound of $N_\delta(s)$ in Lemma 1, which is obtained by the classical technique of changes of distributions (Kaufmann et al., 2015; Lai & Robbins, 1985; Audibert & Bubeck, 2010) and Theorem 1 in Kaufmann et al. (2015).

Theorem 1. *(Kaufmann et al., 2015). Let v and v' be two bandit models with K arms, such that for all $a \in [K]$, the distributions $\mathbb{P}_a$ and $\mathbb{P}'_a$ are mutually absolutely continuous. For any almost-surely finite stopping time σ with respect to $\mathcal{F}_t$, we have*

$$\sum_{a \leq K} \mathbb{E}[N_\delta(s_a)] KL(\mathbb{P}_a, \mathbb{P}'_a) \geq \sup_{\mathcal{E} \in (\mathcal{F}_t)} d(P_v(\varepsilon), P_{v'}(\mathcal{E})),$$

where $d(x, y) = x \log(x/y) + (1 - x) \log[(1 - x)/(1 - y)]$.

Lemma 1. *Let $s' = \arg\min_{s \in nei(s^*) \backslash s^*} KL(\mathbb{P}_{s^*}, \mathbb{P}_s)$, then the number of pulls on $nei(s^*)$ of any δ-PAC algorithm has a lower bound as follows:*

$$\sum_{s \in nei(s^*)} \mathbb{E}[N_\delta(s)] \geq \Bigg(\frac{1}{KL(\mathbb{P}_{s^*}, \mathbb{P}_{s'})} + \sum_{s \in nei(s^*) \backslash s^*} \frac{1}{KL(\mathbb{P}_s, \mathbb{P}_{s^*})} \Bigg) \log \frac{1}{2.4\delta}.$$

Proof. By Assumption 1, for all arms $s \in nei(s^*)$, there exists an alternative model, in which the only arm modified is arm s, and the modified distribution $\mathbb{P}'_s$ satisfies:

- $KL(\mathbb{P}_s, \mathbb{P}_{s^*}) < KL(\mathbb{P}_s, \mathbb{P}'_s) < KL(\mathbb{P}_s, \mathbb{P}_{s^*}) + \alpha$, and $\mathbb{E}\mathbb{P}'_s < \mu_{s^*}$ for $s \in row(s^*) \backslash s^*$
- $KL(\mathbb{P}_s, \mathbb{P}_{s^*}) < KL(\mathbb{P}_s, \mathbb{P}'_s) < KL(\mathbb{P}_s, \mathbb{P}_{s^*}) + \alpha$, and $\mathbb{E}\mathbb{P}'_s > \mu_{s^*}$ for $s \in col(s^*) \backslash s^*$
- $KL(\mathbb{P}_{s^*}, \mathbb{P}_{s'}) < KL(\mathbb{P}_{s^*}, \mathbb{P}'_{s^*}) < KL(\mathbb{P}_{s^*}, \mathbb{P}_{s'}) + \alpha$, and $\mathbb{E}\mathbb{P}'_{s^*} < \mu_{s'}$ for $s' \in col(s^*) \backslash s^*$ or $\mathbb{E}\mathbb{P}'_{s^*} > \mu_{s'}$ for $s' \in row(s^*) \backslash s^*$.

Denote the original bandit model by v and the modified one by v'. In particular, the NE for v' is no longer s^*. Consider the event $\mathcal{E}$: the recommendation rule recommends s^* as NE. Any δ-PAC algorithm satisfies $P_v(\mathcal{E}) > 1 - \delta$ and $P_{v'}(\mathcal{E}) < \delta$, so by Theorem 1, $\mathbb{E}[N_\delta(s)] KL(\mathbb{P}_s, \mathbb{P}'_s) \geq d(P_v(\mathcal{E}), P_{v'}(\mathcal{E})) \geq \log \frac{1}{2.4\delta}$. Hence we have:

$$\mathbb{E}[N_\delta(s)] \geq \frac{\log 1/(2.4\delta)}{KL(\mathbb{P}_s, \mathbb{P}'_s)} \geq \begin{cases} \frac{\log 1/(2.4\delta)}{KL(\mathbb{P}_s, \mathbb{P}_{s^*}) + \alpha}, s \neq s^* \\ \frac{\log 1/(2.4\delta)}{KL(\mathbb{P}_{s^*}, \mathbb{P}_{s'}) + \alpha}, s = s^* \end{cases}$$

Let $\alpha \to 0$ and we complete the proof. □

From the proof, we can see that the lower bound relies on the fact that we can change the best arm (i.e., $NE(\mathbb{M})$ in our case) by changing the distribution of a single arm. However, this proof technique is not suitable for

$s \notin nei(s^*)$ because the NE will not change no matter what the distribution of an arm $s \notin nei(s^*)$ is.

In theory, we only need to pull $s \in nei(s^*)$ to identify NE because of the same reason (i.e., the distributions of arm $s \notin nei(s^*)$ won't change NE). In practice however a policy does not know which arm is in $nei(s^*)$ in advance, so it may make some pulls on $s \notin nei(s^*)$ before making a sufficient number of pulls on $nei(s^*)$. So we can consider the arm selection as a hypothesis testing problem, and then use the lower bound techniques for the minimax risk of hypothesis testing (See Chapter 2 in Tsybakov (2009)). Specifically, our proof is based on the following lemma:

Lemma 2. *Let $P_1, \cdots, P_K$ be probability distributions supported on some set $\mathcal{X}$, with P_i absolutely continuous w.r.t P_1. For any measurable function $\psi : \mathcal{X} \to [K]$, we have:*

$$\sum_{k=1}^{K} P_k(\psi = k) \geq \frac{1}{e} \exp\{-\sum_{k=2}^{K} KL(P_1, P_k)\},$$

where $P_k(\psi{=}k) := P_k(\{x : \psi(x){=}k\})$ for clarity.

Proof. This lemma is an extension of Lemma 2.6 in Tsybakov (2009) from two distributions to multiple distributions. We put the proof in Appendix A. □

Now we show what the hypotheses to be tested in our problem are and how to apply Lemma 2. Without loss of generality, consider a game $\mathbb{M}$ with $NE(\mathbb{M}) = (1,1)$. Let's consider a set of hypotheses: new games $h_{i,j}$ constructed by swapping the i-th row with the first row and the j-th column with the first column of $\mathbb{M}$. Obviously, these games are essentially the same game up to permutation. We will show the lower bound on the maximum number of pulls among these hypotheses by arbitrary policies.

Formally, define $f_{i,j}(i', j') = (i'', j'')$ where $i'' = i'$ if $i' \notin \{1, i\}$ else $i'' = 1 + i - i'$ and $j'' = j'$ if $j' \notin \{1, j\}$ else $j'' = 1 + j - j'$. Let $h_{i,j}$ specify a game such that the distribution of arm (i', j') is $\mathbb{P}_{f_{i,j}(i',j')}$. Let $\mathcal{H}^t(h_{i,j})$ be the sum of the expected number of pulls on arms $s \notin nei((i,j))$ until time step t under hypothesis $h_{i,j}$:

$$\mathcal{H}^t(h_{i,j}) := \sum_{t' \leq t} \mathbb{1}[I_{t'} \notin nei((i,j)); h_{i,j}],$$

and let $\mathcal{H}^t := \max_{i,j} \mathcal{H}^t(h_{i,j})$ be the maximum number of pulls under any hypothesis. Then, theorem 2 shows the lower bound of $\mathcal{H}^t$.

Theorem 2. *For any $i, j \geq 2$, let $\Delta := (m-1)(n-1)(Mu(\mathbb{P}_{i,j}, \mathbb{P}_{1,1}) + Mu(\mathbb{P}_{i,1}, \mathbb{P}_{1,j}) + \sum_{j' \geq 2}^{j' \neq j} Mu(\mathbb{P}_{i,j'}, \mathbb{P}_{1,j'}) + \sum_{i' \geq 2}^{i' \neq i} Mu(\mathbb{P}_{i',1}, \mathbb{P}_{i',j})) + mMu(\mathbb{P}_{i,1}, \mathbb{P}_{i,j}) + nMu(\mathbb{P}_{1,j}, \mathbb{P}_{i,j})$ where $Mu(P_1, P_2) := KL(P_1, P_2) + KL(P_2, P_1)$. Then, we have the lower bound:*

$$\mathcal{H}^t \geq \frac{e^{-\Delta}(1 - e^{-t\Delta})}{4e(1 - e^{-\Delta})}. \quad (1)$$

Proof. With straight-forward computations, we have:

$$\begin{aligned}
\mathcal{H}^t \geq & \frac{1}{4}\left(\mathcal{H}^t(h_{1,1}) + \mathcal{H}^t(h_{i,j}) + \mathcal{H}^t(h_{1,j}) + \mathcal{H}^t(h_{i,1})\right) \\
= & \frac{1}{4}\Bigg(\sum_{t' \leq t} \mathbb{1}[I_{t'} \notin nei((1,1)); h_{1,1}] \\
& + \mathbb{1}[I_{t'} \notin nei((i,j)); h_{i,j}] + \mathbb{1}[I_{t'} \notin nei((1,j)); h_{1,j}] \\
& + \mathbb{1}[I_{t'} \notin nei((i,1)); h_{i,1}]\Bigg) \\
\geq & \frac{1}{4}\sum_{t' \leq t}\Bigg(\sum_{i',j' \geq 2} \mathbb{1}[I_{t'} = (i', j'); h_{1,1}] \\
& + \mathbb{1}[I_{t'} = (1,1); h_{i,j}] + \sum_{j' \geq 2} \mathbb{1}[I_{t'} = (1, j'); h_{i,1}] \\
& + \sum_{i' \geq 2} \mathbb{1}[I_{t'} = (i', 1); h_{1,j}]\Bigg).
\end{aligned}$$

Define $P^t_{h_{a,b}}$ as the distribution of observations until time step t under the hypothesis $h_{a,b}$, and define function

$$g(i', j') := \begin{cases} (1,1) & i', j' \geq 2, \\ (1,j) & i' \geq 2, j' = 1, \\ (i,1) & j' \geq 2, i' = 1, \\ (i,j) & i' = j' = 1. \end{cases}$$

Let $P^t_{i',j'} = P^t_{h_{g(i',j')}}$. Consider events $ev(t, s)$: policy selects arm s at time step t. We have:

$$\begin{aligned}
\mathcal{H}^t &\geq \frac{1}{4}\sum_{t' \leq t}\sum_{i',j'} P^{t'}_{i',j'}(ev(t', (i', j'))) \\
&\geq \frac{1}{4e}\sum_{t' \leq t} \exp\{-\sum_{i',j'} KL(P^{t'}_{1,1}, P^{t'}_{i',j'})\} \\
&\geq \frac{1}{4e}\sum_{t'=1}^{t} \exp\{-t\Delta\} \\
&= \frac{e^{-\Delta}(1 - e^{-t\Delta})}{4e(1 - e^{-\Delta})}.
\end{aligned}$$

The second inequality is proven by Lemma 2. The third is by the fact that let $P_{i,j}(a,b)$ denote the distribution of arm (a,b) under hypothesis $h_{g(a,b)}$, then we have $KL(P^t_{1,1}, P^t_{i',j'}) = \sum_{t' \leq t} KL(P_{1,1}(I_{t'}), P_{i',j'}(I_{t'}))$; and note that $KL(P_{1,1}(I_{t'}), P_{i',j'}(I_{t'})) \leq \sum_{i'',j''} KL(P_{1,1}(i'', j''), P_{i',j'}(i'', j''))$; summing over all i', j', we get the third inequality. □

We can get a different lower bound by choosing a different i, j in Theorem 2, and take the maximum one. Though our lower bound is not on the expected number of pulls, it intuitively answers why the pulls on $s \notin nei(s^*)$ are unavoidable. Obviously, $\frac{e^{-\Delta}(1-e^{-t\Delta})}{4e(1-e^{-\Delta})} \leq \frac{e^{-\Delta}}{4e(1-e^{-\Delta})}$, which suggests that there is a policy which pulls on $s \notin nei(s^*)$ for a bounded number of times. This result inspires us to design a policy with a bounded number of pulls on $s \notin nei(s^*)$, as shown in Section 4.

4. Algorithms

We now present two δ-PAC algorithms for our problem. The first one is inspired by LUCB (Kalyanakrishnan et al., 2012) and UCB1 (Auer et al., 2002), while the second one follows another line of BAI algorithms which are based on the successive action eliminations (Even-Dar et al., 2006; Maron & Moore, 1997).

4.1. LUCB-G

We first present and analyze the LUCB-G (i.e., LUCB for Game) algorithm, as illustrated in Alg. 1.

4.1.1. Algorithm

Informally, our problem can be divided into $m+n$ bandit tasks—m for identifying $s_r^*(i) := \arg\min_{s\in R_i} \mu_s$ and n for $s_c^*(j) := \arg\max_{s\in C_j} \mu_s$. So in each round[2], LUCB-G can be divided into two stages. In the first stage, it selects two bandit tasks—a row and a column. Note that LUCB-G tries to identify s_r^* and s_c^* after each round. If before round γ, it identified $\bar{s}_r^*(i)$ as the arm with minimum mean in R_i, or identified $\bar{s}_c^*(j)$ [3] as the arm with maximum mean in C_j, then LUCB-G will not select row i or column j. That is to say, we only select bandit models from the following rows and columns at the γ-th round:

$$ar(\gamma) := \{i \in [m] : \bar{s}_r^*(i) \text{ not identified until round } \gamma.\}$$
$$ac(\gamma) := \{j \in [n] : \bar{s}_c^*(j) \text{ not identified until round } \gamma.\}$$

We'll introduce how to identify these arms later in this section. In the second stage, we pull arms according to past observations and some confidence bound function $\beta : \mathbb{N}_+ \times \mathbb{N}_+ \to (0,\infty)$, which will be presented soon in Section 4.1.2. Here we show our first policy in Alg. 1.

We have a clock for each bandit task, and our confidence bounds rely on them. Define $\tau_r(i,t)$ as the set of all the time steps t' that satisfy the two requirements: (1) $t' < t$; (2) at the round when t' takes place, row i is chosen, and at least one line from 15 to 17 is executed. Similarly, define $\tau_c(j,t)$ as the set of all the time steps t' that satisfy the two requirements: (1) $t' < t$; (2) at the round when t' takes place, column j is chosen, and at least one line from 21 to 23 is executed.

The method of identifying $\bar{s}_r^*(i)$ and $\bar{s}_c^*(j)$ also relies on the confidence bound function β. For an arm s, define $L(s,u,t) = \bar{\mu}_s(t') - \beta(u,t), U(s,u,t) = \bar{\mu}_s(t') + \beta(u,t)$[4]. And let $T_s(\tau) = \{t : I_t = s, t \in \tau\}$. Alg. 1 determines $\bar{s}_r^*(i)$ and $\bar{s}_c^*(j)$ at time step t as follows:

- If $\exists s \in R_i$, for all $s' \in R_i \backslash s$, we have $U(s, |T_s(\tau)|, |\tau|) \leq L(s', |T_{s'}(\tau)|, |\tau|)$, where $\tau := \tau_r(i,t)$, then Alg. 1 takes s as $\bar{s}_r^*(i)$.
- If $\exists s \in C_j$, for all $s' \in C_j \backslash s$, we have $L(s, |T_s(\tau)|, |\tau|) \geq U(s', |T_{s'}(\tau)|, |\tau|)$, where $\tau := \tau_c(j,t)$, then Alg. 1 takes s as $\bar{s}_c^*(j)$.

Now we introduce the stopping and recommendation rules for Alg. 1.

Stopping and recommendation rules: The policy stops and recommends NE as follows:

- If after round γ, there is an arm s, Alg. 1 takes it as $\bar{s}_r^*(s[1])$ and $\bar{s}_c^*(s[2])$. Then Alg. 1 stops and recommends s as the NE.
- Else if after round γ, $\bar{s}_r^*(i)$ and $\bar{s}_c^*(j)$ have all been determined. Then the policy stops and the recommendation rule determines that the underlying game does not have a NE.

4.1.2. δ-PAC

We now show that Alg. 1 is a δ-PAC algorithm. Lemma 3 guarantees that if $\beta(u,t)$ satisfies the requirements in InEq. (2), then the probability that there is an arm violating its confidence bounds is less than δ. Our choice of the confidence bound function is $\beta(u,t) = \sqrt{\frac{\log(mnt^4/4\delta)}{2u}}$,[5] which satisfies this requirement. And Theorem 3 is a simple application of Lemma 3 since if no arm violates its confidence bounds, the stopping and recommendation rules won't make mistakes.

Lemma 3. *Let $\beta(u,t) : \mathbb{N}_+ \times \mathbb{N}_+ \to (0,\infty)$ be a function such that:*

$$\sum_{t=1}^{\infty}\sum_{u=1}^{t} \exp\{-2u\beta(u,t)^2\} \leq \frac{\delta}{2K}. \qquad (2)$$

Consider a bandit model v with K arms, for each arm, there is a sequence $(t_1,u_1),(t_2,u_2),\cdots$ such that $t_i \geq u_i, t_{i+1} \geq t_i, u_{i+1} \geq u_i$, and a sequence $u_1', u_2', \cdots$ such that $u_i' \geq u_i$, and then the probability that $\exists s \in v, i$ such that

$$|\frac{1}{u_i'}\sum_{i=1}^{u_i'} Y_i^s - \mu_s| > \beta(u_i,t_i)$$

is less than δ.

[2] We pull arms for several times in each round.

[3] With a probability, $\bar{s}_r^*(i) \neq s_r^*(i)$ or $\bar{s}_c^*(j) \neq s_c^*(j)$.

[4] In this paper, we have $t' \geq t$. Thus in fact L and U are functions of t', but for convenience we omit the notation of t'.

[5] For convenience, let $1/0 = +\infty$. So if $u = 0$, $\beta(u,t) = +\infty$.

Proof. The proof can be found in Appendix B. □

Theorem 3. *The probability of making mistakes by the recommendation and stopping rules of LUCB-G is at most δ.*

Proof. If all arms don't violate their confidence bounds, then it is easy to see that we determine NE correctly. As our choice of $\beta(u,t)$ satisfies the condition in InEq. (2), we can use Lemma 3 to get the result. □

4.1.3. Sample complexity

We analyze the sample complexity of Alg. 1 in this section. We provide a proof of the sample complexity when $NE(\mathbb{M}) \neq none$ here, and the sample complexity when $NE(\mathbb{M}) = none$ is a straight-forward application of the result in LUCB (Kalyanakrishnan et al., 2012).

For convenience, let $H_r(i) = \sum_{s\in R_i\backslash s_r^*(i)} \frac{1}{(\mu_s-\mu_{s_r^*(i)})^2}$ and $H_c(j) = \sum_{s\in C_j\backslash s_c^*(j)} \frac{1}{(\mu_s-\mu_{s_c^*(j)})^2}$. In the second stage of each round, LUCB-G pulls arms similarly as LUCB and UCB1. When the NE of the underlying game exists, the pulls can be divided into three parts:

- part1: Time steps t such that s^* is the NE of $\bar{\mathbf{M}}_t$.
- part2: Time steps t such that $s \neq s^*$ is the NE of $\bar{\mathbf{M}}_t$.
- part3: Time steps t such that there is no NE of $\bar{\mathbf{M}}_t$.

Therefore, the total sample complexity can be decomposed as the summation of the bounds for the three parts.

We first provide sample complexity bounds for the pulls in part2 and part3, which appear because of the algorithm's misjudgments on which arm is $NE(\mathbb{M})$ during training, while deferring the bound for part1 to Lemma 6, which is relatively standard.

Obviously, at time step t, if part2 or part3 happens, then $s^* \neq \arg\min_{s'\in row(s^*)} \bar{\mu}_{s'}(t)$ or $s^* \neq \arg\max_{s'\in col(s^*)} \bar{\mu}_{s'}(t)$. Lemma 4 ensures that these events will not happen with a high probability if Alg. 1 selects $row(s^*)$ and $col(s^*)$ for a sufficiently large number of times. The key idea is to use the UCB1 policy (line 20, 26) (Auer et al., 2002): considering column j, when the algorithm chooses it in the first stage, the algorithm pulls an arm in C_j by UCB1. This ensures that with a high probability, the policy pulls a sufficiently large number of times on $s_c^*(j)$, and then by Hoeffding's inequality, we get the bound. A formal statement is in Lemma 4.

Lemma 4. *Without loss of generality, consider column j, let $\phi(\gamma) := \mathbb{1}[s_c^*(j) \neq \arg\max_{s'\in C_j} \bar{\mu}_{s'}$ when Alg. 1 selects column j for the γ-th time$]$, let $\phi = \sum_{\gamma=1}^{\infty} \phi(\gamma)$. Then, the expectation of ϕ satisfies the inequality:*

$$\mathbb{E}[\phi] - c_1 H_c(j)(\log \mathbb{E}[\phi])^2 - c_2 H_c(j) - c_3 \leq 0, \qquad (3)$$

where c_1, c_2, c_3 are positive constants.

Algorithm 1 LUCB-G

```
 1: Input: distribution matrix $\mathbb{M}$, confidence $\delta$
 2: Pull all arms
 3: $Chr(i) = Chc(j) = 0$ for $i \in [m], j \in [n], t = m * n$
 4: while Not stop do
 5:   $sel_r(i) = sel_c(j) = 0$ for $i \in [m], j \in [n]$
 6:   if $s = NE(\bar{\mathbf{M}}_t) \neq none$ then
 7:     $sel_r(s[1]) = sel_c(s[2]) = 1$
 8:   else
 9:     Let $\hat{i} := \arg\min_{i\in ar[\gamma]} Chr(i)$
10:     Let $\hat{j} := \arg\min_{j\in ar[\gamma]} Chc(j)$
11:     $sel_r(\hat{i}) = sel_c(\hat{j}) = 1$
12:   end if
13:   if $\exists\hat{i} \in [m], sel_r(\hat{i}) = 1$ then
14:     $Chr(\hat{i}) = Chr(\hat{i}) + 1$
15:     Pull $s_1 := \arg\min_{s\in R_{\hat{i}}} \bar{\mu}_s(t), t = t + 1$
16:     Pull $\arg\min_{s\in R_{\hat{i}}} \bar{\mu}_s - \sqrt{2\frac{\log|\tau_r(\hat{i},t)|/3}{|T_s(\tau_r(\hat{i},t))|}}, t = t+1$
17:     Pull $\arg\min_{s\in R_{\hat{i}}\backslash s_1} L(s, |T_s(\tau_r(\hat{i},t))|, |\tau_r(\hat{i},t)|), t = t + 1$
18:   end if
19:   if $\exists\hat{j} \in [n], sel_c(\hat{j}) = 1$ then
20:     $Chc(\hat{j}) = Chc(\hat{j}) + 1$
21:     Pull $s_1 := \arg\max_{s\in C_{\hat{j}}} \bar{\mu}_s(t), t = t + 1$
22:     Pull $\arg\max_{s\in C_{\hat{j}}} \bar{\mu}_s + \sqrt{2\frac{\log|\tau_r(\hat{j},t)|/3}{|T_s(\tau_r(\hat{j},t))|}}, t = t+1$
23:     Pull $\arg\max_{s\in C_{\hat{j}}\backslash s_1} U(s, |T_s(\tau_r(\hat{j},t)|), |\tau_r(\hat{j},t)|), t = t + 1$
24:   end if
25: end while
```

Proof. Let t_γ denote the time step when Alg. 1 selects column j for the γ-th time and $\xi_\gamma = \arg\max_{s\in C_j} \bar{\mu}_s(t_\gamma)$. Let $\Delta_\gamma = \mu_{s_c^*(j)} - \mu_{\xi_\gamma}$. If $\xi_\gamma \neq s_c^*(j)$, then we have $\bar{\mu}_{s_c^*(j)}(t_\gamma) \leq \mu_{s_c^*(j)} - \Delta_\gamma/2$ or $\bar{\mu}_{\xi_\gamma}(t_\gamma) \geq \mu_{\xi_\gamma} + \Delta_\gamma/2$. So let $Set1(s) = \{\gamma : \xi_\gamma = s\}$, $Set2(s) = \{\gamma \in Set1(s) : \bar{\mu}_{\xi_\gamma}(t_\gamma) \geq \mu_{\xi_\gamma} + \Delta_\gamma/2\}$ and $Set3(s) = \{\gamma \in Set1(s) : \bar{\mu}_{s_c^*(j)}(t_\gamma) \leq \mu_{s_c^*(j)} - \Delta_\gamma/2\}$. With the above argument, for $s \in C_j\backslash s_c^*(j)$, we have $\mathbb{E}[|Set1(s)|] \leq \mathbb{E}(|Set2(s)| + |Set3(s)|)$.

Due to Alg. 1, the policy pulls s for rounds $\gamma \in Set1(s)$. So by Hoeffding's inequality, $\mathbb{E}|Set2(s)| \leq \sum_{\gamma=1}^{\infty} \exp\{-2\gamma((\mu_{s_c^*(j)} - \mu_s)/2)^2\} \leq 2/(\mu_{s_c^*(j)} - \mu_s)^2$.

Now consider $Set3(s)$, which is computed as:

$$\mathbb{E}|Set3(s)| = \mathbb{E}\left[\sum_{\gamma\in Set1(s)} \mathbb{1}[\bar{\mu}_{s_c^*(j)}(t_\gamma) \leq \frac{(\mu_{s_c^*(j)} + \mu_s)}{2}]\right].$$

Let $T(t)$ be the number of pulls on $s_c^*(j)$ at line 22 in Alg. 1 at that time step t. By Hoeffding's inequality and straight-forward computations, we have

$$\begin{aligned}\mathbb{E}|Set3(s)| \leq &\mathbb{E}\Big[\sum_{\gamma\in Set1(s)} \mathbb{1}[T(t_\gamma) \leq \frac{\gamma}{2}] \\ &+\mathbb{1}[\bar{\mu}_{s_c^*(j)}(t_\gamma) \leq \frac{(\mu_{s_c^*(j)}+\mu_s)}{2}; T(t_\gamma) \geq \frac{\gamma}{2}]\Big] \\ \leq &\mathbb{E}\Big[\sum_{\gamma\in Set1(s)} \mathbb{1}[T(t_\gamma) \leq \frac{\gamma}{2}] \\ &+ \exp\{-\gamma(\frac{\mu_{s_c^*(j)}-\mu_s}{2})^2\}\Big] \\ \leq &\mathbb{E}\left[\sum_{\gamma\in Set1(s)} \mathbb{1}[T(t_\gamma) \leq \frac{\gamma}{2}] + \frac{4}{(\mu_{s_c^*(j)}-\mu_s)^2}\right].\end{aligned}$$

Note that Line 22 is the UCB1 policy proposed by Auer et al. (2002). So by Theorem 1 in Auer et al. (2002) (A slightly modification on this theorem, see Appendix C), we have $\mathbb{E}[\gamma' - T(t_{\gamma'})] \leq O(H_c(j)\log\gamma')$. Then with Markov inequality, we can get $P[\gamma' - T(t_{\gamma'}) \geq \gamma'/2] \leq \frac{O(H_c(j)\log\gamma')}{\gamma'}$, that is, $P[T(t_{\gamma'}) \leq \gamma'/2] \leq \frac{O(H_c(j)\log\gamma')}{\gamma'}$. So let $Set3 = \cup_{s\in C_j \setminus s_c^*(j)} Set3(s)$, we have:

$$\begin{aligned}\mathbb{E}[|Set3|] \leq &O(H_c(j)) + \mathbb{E}\left[\sum_{\gamma\in Set1} \mathbb{1}[T(t_\gamma) \leq \frac{\gamma}{2}]\right] \\ \leq &O(H_c(j)) + \mathbb{E}\left[\sum_{\gamma\in Set1} \frac{O(H_c(j)\log\gamma)}{\gamma}\right] \\ = &O(H_c(j)) + \mathbb{E}\left[\sum_{\gamma=1}^{|Set1|} \frac{O(H_c(j) log \gamma)}{\gamma}\right] \\ \leq &O(H_c(j)) + \mathbb{E}\left[O(H_c(j)(\log|Set1|)^2)\right] \\ \leq &O(H_c(j) + H_c(j)(\log\mathbb{E}|Set1|)^2)\end{aligned}$$

where $Set1 = \cup_{s\in C_j \setminus s_c^*(j)} Set1(s)$. The third inequality is by simple integration, and the last inequality holds because $f(x) = (\log x)^2$ is a concave function for $x > e$. Note that $\phi = |Set1|$. With $\mathbb{E}|Set1(s)| \leq \mathbb{E}(|Set2(s)| + |Set3(s)|)$, we complete the proof. □

It is noteworthy that although we do not have an analytical solution of $\mathbb{E}[\phi]$ from InEq. (3), it is obvious that the solution is bounded, that is, it will not diverge as $\delta \to 0$. Then, we can get the sample complexity on part 2 and 3, as in Lemma 5.

Lemma 5. *Suppose $s^* = NE(\mathbb{M}) \neq$ none, let $S_w = \{$Rounds γ such that $R_{s^*[1]}$ or $C_{s^*[2]}$ is not chosen by Alg. 1$\}$. Let $\Lambda(a)$ be the maximum value among all solutions that satisfy the following inequality (constants c_1, c_2, c_3 are the same as in Lemma 4):*

$$x - c_1 a(\log x)^2 - c_2 a - c_3 \leq 0.$$

Then, the following inequality holds:

$$\begin{aligned}\mathbb{E}|S_w| \leq &O\Big((\sum_i \Lambda(H_r(i)) + \sum_j \Lambda(H_c(j))) \\ &+(m+n)(\Lambda(H_r(s^*[1])) + \Lambda(H_c(s^*[2])))\Big).\end{aligned}$$

Proof. We put the proof in Appendix D. □

The bound on part1 is based on the result of LUCB, as in Lemma 6, which has almost the same result on the sample complexity as policy LUCB.

Lemma 6. *Without loss of generality, considering column j, suppose $\bar{s}_c^*(j)$ is identified by Alg. 1 after being selected for $\gamma_c(j)$ rounds, then*

$$\mathbb{E}[\gamma_c(j)] = O\left(H_c(j)\log(\frac{H_c(j)}{\delta})\right).$$

Proof. The proof is the same as that of Theorem 6 in (Kalyanakrishnan et al., 2012), except slightly changes on description and constants. See Appendix E. □

With the above results, we are ready to get our major result on the sample complexity of LUCB-G, as in Theorem 4.

Theorem 4. *When $NE(\mathbb{M}) \neq$ none, the sample complexity of LUCB-G is:*

$$\begin{aligned}O\Big(&H_r(s^*[1])\log\frac{H_r(s^*[1])}{\delta} + H_c(s^*[2])\log\frac{H_c(s^*[2])}{\delta} \\ &+\mathbb{E}[|S_w|]\Big),\end{aligned}$$

where $\mathbb{E}[|S_w|]$ is bounded as in Lemma 5. When $NE(\mathbb{M}) =$ none, the sample complexity of LUCB-G is:

$$O\left(\sum_i H_r(i)\log\frac{H_r(i)}{\delta} + \sum_j H_c(j)\log\frac{H_c(j)}{\delta}\right).$$

Proof. By Lemma 5 and Lemma 6, with straight-forward computations, we get the complexity. □

Note that the sample complexity of LUCB-G is optimal within a constant gap if $NE(\mathbb{M}) \neq$ *none*. This is because that $\mathbb{E}[|s_w|]$ is bounded and for some family $\mathcal{P}$, $P_1, P_2 \in \mathcal{P}$, the KL-divergence $KL(P_1, P_2)$ has the same order as the squared mean-difference $(\mathbb{E}P_1 - \mathbb{E}P_2)^2$ (e.g., normal distributions with unit variances). Therefore, we can replace the KL-divergence terms in Lemma 1 by the corresponding squared mean-difference terms.

```
Algorithm 2 Racing
Input: distribution matrix $\mathbb{M}$, confidence $\delta$
$\gamma = 1$.
while Not stop do
    Pull all arms except those have been eliminated.
    For an active arm $s$, if $\exists s' \in row(s) : 2\beta_1(\gamma) < \bar{\mu}_s - \bar{\mu}_{s'}$ and $\exists s' \in col(s) : \bar{\mu}_{s'} - \bar{\mu}_s > 2\beta_1(\gamma)$, then we eliminate $s$.
    For all sequences of arms $S = \{s_1, s_2, \cdots, s_{2k}\}$, if $S$ satisfies $\forall i \in \mathbb{N}$:
        • $s_{2i+1} \in row(s_{2i}), s_{2i+2} \in col(s_{2i+1})$.
        • all arms are eliminated in $row(s_{2i}), col(s_{2i+1})$ except $s \in S$.
        • $\bar{\mu}_{s_{2i+1}} - \bar{\mu}_{s_{2i}} > 2\beta_1(\gamma)$.
        • $\bar{\mu}_{s_{2i+1}} - \bar{\mu}_{s_{2i+2}} > 2\beta_1(\gamma)$,
    where $s_j := s_{(j-1)\%(2k)+1}$, eliminate all arms in $S$.
    $\gamma = \gamma + 1$.
end while
```

4.2. A racing algorithm

Finally, we present another algorithm, along the line of racing algorithms for BAI (Even-Dar et al., 2006; Maron & Moore, 1997). A racing algorithm maintains a set of active arms, and during each round it samples all the active arms and then eliminates some arms according to certain rules.

However, we cannot eliminate an arm when the algorithm "knows" it cannot be the NE immediately. Consider a 2×2 game. Suppose an algorithm determines $(1, 1) \neq \bar{s}_r(1)$, and eliminates it immediately. Then we cannot determine whether $(2, 1)$ is NE or not. Therefore, our racing algorithm eliminates arm s only if Alg. 2 determines that $s \notin \{\bar{s}_r^*(s[1], \bar{s}_c^*(s[2]))\}$ or $\bar{s}_r^*(s[1]) \neq \bar{s}_c^*(s[2])$. Let $\beta_1(\gamma) = \sqrt{\frac{\log(cmn\gamma^2/\delta)}{2\gamma}}$. Our racing algorithm is shown in Alg. 2, whose stopping and recommendation rules are:

- If only an arm is not eliminated after round γ, then recommend this arm as NE.
- If all arms are eliminated after round γ, then the algorithm determines $NE = none$.

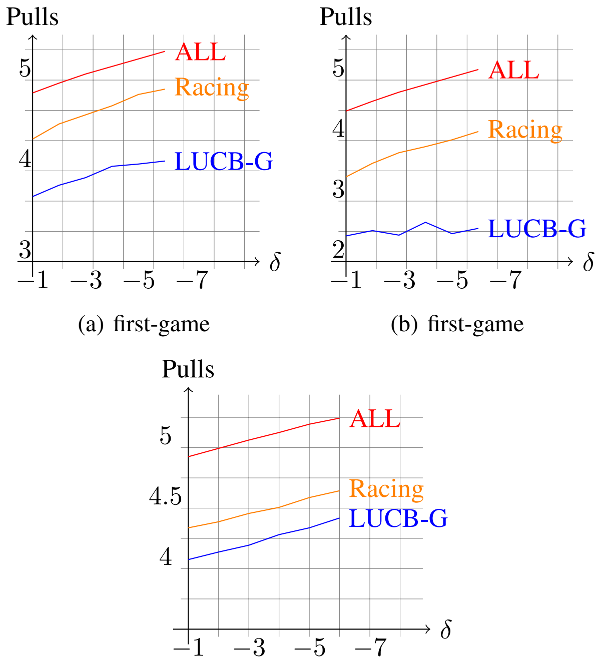
As shown in Theorem 5, this algorithm is δ-PAC and it will terminate in finite time step with probability at least $1 - \delta$.

Theorem 5. *Alg. 2 is δ-PAC and will terminate in finite time with probability at least $1 - \delta$.*

Proof. We put the proof in Appendix F. □

5. Experiments

We now empirically verify the sample complexity of our algorithms. We choose a simple algorithm as our baseline (denoted by ALL), which pulls all arms at each round until stopping. The stopping and recommendation rules are the same as LUCB-G, and the confidence bound for this baseline is slightly different, see Appendix G for details.



(a) first-game

(b) first-game

(c) second-game

Figure 1. The results on two simulated games.

We evaluate on synthetic 5×5 games, where the payoffs are all random Bernoulli variables. The first game has a NE, while the second game has no NE. The results are shown in Fig. 1, where both axes are in log-scale with base 10. The number of pulls needed for both games are shown in Fig. 1(a) and Fig. 1(c) separately and our algorithms outperform the baseline (i.e., ALL). Fig. 1(b) shows the number of pulls on $s \notin nei(s^*)$ in the first game and we can see that this number is bounded, agreeing with our analysis.

6. Conclusions and Discussions

We analyze the two-player zero-sum static game with random payoffs via efficient sampling and give a lower bound of the sample complexity in the case that the Nash Equilibrium (NE) exists. We then present two δ-PAC algorithms to identify the NE. They follow two lines of algorithms for the best arm identification problem in the fixed confidence setting. The sample complexity of the first algorithm is optimal within a constant gap if NE exists.

As we cannot give an explicit form for the expectation number of pulls wasting on arms in neither the row nor the column of the NE, our lower bound can be loose to some extent. It is worth of having a further study for tighter lower bounds. Moreover, an analysis of the sample complexity in the case that NE does not exist is still an open problem, and we expect better work on it in the future.

Acknowledgements

This work is supported by the National Basic Research (973) Program of China (No. 2013CB329403), NSFC Projects (Nos. 61620106010 and 61621136008), and the Youth Top-notch Talent Support Program.

References

Audibert, Jean-Yves and Bubeck, Sébastien. Best arm identification in multi-armed bandits. In *COLT-23th Conference on Learning Theory-2010*, pp. 13–p, 2010.

Auer, Peter, Cesa-Bianchi, Nicolo, and Fischer, Paul. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Babichenko, Yakov. Query complexity of approximate nash equilibria. *Journal of the ACM (JACM)*, 63(4):36, 2016.

Chen, Xi, Deng, Xiaotie, and Teng, Shang-Hua. Settling the complexity of computing two-player nash equilibria. *Journal of the ACM (JACM)*, 56(3):14, 2009.

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

Even-Dar, Eyal, Mannor, Shie, and Mansour, Yishay. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun): 1079–1105, 2006.

Fearnley, John and Savani, Rahul. Finding approximate nash equilibria of bimatrix games via payoff queries. *ACM Transactions on Economics and Computation (TEAC)*, 4(4):25, 2016.

Fearnley, John, Gairing, Martin, Goldberg, Paul W, and Savani, Rahul. Learning equilibria of games via payoff queries. *Journal of Machine Learning Research*, 16: 1305–1344, 2015.

Garivier, Aurélien, Kaufmann, Emilie, and Koolen, Wouter M. Maximin action identification: A new bandit framework for games. In *29th Annual Conference on Learning Theory*, pp. 1028–1050, 2016.

Goldberg, Paul W and Roth, Aaron. Bounds for the query complexity of approximate equilibria. *ACM Transactions on Economics and Computation (TEAC)*, 4(4):24, 2016.

Goldberg, Paul W and Turchetta, Stefano. Query complexity of approximate equilibria in anonymous games. In *International Conference on Web and Internet Economics*, pp. 357–369. Springer, 2015.

Hart, Sergiu and Nisan, Noam. The query complexity of correlated equilibria. *Games and Economic Behavior*, 2016.

Jamieson, Kevin and Nowak, Robert. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pp. 1–6. IEEE, 2014.

Jordan, Patrick R, Vorobeychik, Yevgeniy, and Wellman, Michael P. Searching for approximate equilibria in empirical games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pp. 1063–1070. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

Kalyanakrishnan, Shivaram, Tewari, Ambuj, Auer, Peter, and Stone, Peter. Pac subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 655–662, 2012.

Kaufmann, Emilie, Cappé, Olivier, and Garivier, Aurélien. On the complexity of best arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 2015.

Lai, Tze Leung and Robbins, Herbert. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Mannor, Shie and Tsitsiklis, John N. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun): 623–648, 2004.

Maron, Oded and Moore, Andrew W. The racing algorithm: Model selection for lazy learners. In *Lazy learning*, pp. 193–225. Springer, 1997.

Osborne, Martin J and Rubinstein, Ariel. *A course in game theory*. MIT press, 1994.

Tsybakov, Alexandre B. Introduction to nonparametric estimation. Springer, 2009.

Vorobeychik, Yevgeniy, Wellman, Michael P, and Singh, Satinder. Learning payoff functions in infinite games. *Machine Learning*, 67(1-2):145–168, 2007.

Walsh, William E, Parkes, David C, and Das, Rajarshi. Choosing samples to compute heuristic-strategy nash equilibrium. In *International Workshop on Agent-Mediated Electronic Commerce*, pp. 109–123. Springer, 2003.

Wellman, Michael P. Methods for empirical game-theoretic analysis. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, pp. 1552. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.