How hard is it to approximate the best Nash equilibrium?

Elad Hazan
IBM Almaden
ehazan@cs.princeton.edu

Robert Krauthgamer[*]
Weizmann Institute of Science
robert.krauthgamer@weizmann.ac.il

Abstract

The quest for a PTAS for Nash equilibrium in a two-player game seeks to circumvent the PPAD-completeness of an (exact) Nash equilibrium by finding an approximate equilibrium, and has emerged as a major open question in Algorithmic Game Theory. A closely related problem is that of finding an equilibrium maximizing a certain objective, such as the social welfare. This optimization problem was shown to be NP-hard by Gilboa and Zemel [Games and Economic Behavior 1989]. However, this NP-hardness is unlikely to extend to finding an approximate equilibrium, since the latter admits a quasi-polynomial time algorithm, as proved by Lipton, Markakis and Mehta [Proc. of 4th EC, 2003].

We show that this optimization problem, namely, finding in a two-player game an approximate equilibrium achieving large social welfare is unlikely to have a polynomial time algorithm. One interpretation of our results is that the quest for a PTAS for Nash equilibrium should not extend to a PTAS for finding the best Nash equilibrium, which stands in contrast to certain algorithmic techniques used so far (e.g. sampling and enumeration).

Technically, our result is a reduction from a notoriously difficult problem in modern Combinatorics, of finding a planted (but hidden) clique in a random graph $G(n, 1/2)$. Our reduction starts from an instance with planted clique size $k = O(\log n)$. For comparison, the currently known algorithms due to Alon, Krivelevich and Sudakov [Random Struct. & Algorithms, 1998], and Krauthgamer and Feige [Random Struct. & Algorithms, 2000], are effective for a much larger clique size $k = \Omega(\sqrt{n})$.

1 Introduction

Computational aspects of equilibrium concepts, and in particular of Nash equilibrium, have seen major advances over the last few years, both from the side of algorithms and in terms of computational complexity (namely, completeness and hardness results). Perhaps the most celebrated result in this area [CDT06, DGP06] proves that computing a Nash equilibrium in a finite game with two players is PPAD-complete. Consequently, a weaker notion of ε-approximate Nash equilibrium, or in short an ε-equilibrium, was suggested, and the following has emerged as a central open question:

Is there a PTAS for Nash equilibrium?

In other words, is there a polynomial time algorithm that finds an ε-Nash equilibrium for arbitrarily small but fixed $\varepsilon > 0$? Here and in the sequel we follow the literature and assume that the game's payoffs are in the interval $[0, 1]$, and approximations are measured additively; see Section 2 for precise definitions.

While every game has at least one Nash equilibrium, the game may actually have many equilibria, some more desirable than others. Thus, an attractive solution concept is to find Nash equilibrium maximizing an objective such as the social welfare (the total utility of all players). For two-player games this problem is known to be NP-hard [GZ89, CS03]. But as we shall soon see, this hardness result is unlikely to extend to ε-equilibrium.

A fairly simple technique, yet surprisingly powerful, is *random sampling*, where each player's mixed strategy $\vec{x}$ is replaced by another mixed strategy $\vec{x}'$ that has small support, obtained by sampling a few pure strategies independently from $\vec{x}$ and taking $\vec{x}'$ be a uniform distribution over the chosen pure strategies. (We allow repetitions, i.e. the support is viewed as a multiset.) This technique leads to a simple algorithm that finds in a two-player game an ε-equilibrium in quasi-polynomial time $N^{O(\varepsilon^{-2} \log N)}$ [LMM03], assuming that the game is represented as two $N \times N$ matrices.[1] Indeed, applying random sampling on any Nash equilibrium together with Chernoff-like bounds yields an ε-equilibrium consisting of mixed strategies that are each uniform over a

---

[1]Throughout, f is called *quasi-polynomial* if $f(n) \leq n^{O(\log n)}$.

multiset of size $O(\varepsilon^{-2} \log N)$, and such an ε-equilibrium can be found by exhaustive search. In fact, this argument applies also to the social-welfare maximization problem, and thus the algorithm of [LMM03] finds in time $N^{O(\varepsilon^{-2} \log N)}$ an ε-equilibrium whose social-welfare is no more than ε smaller than the maximum social-welfare of a Nash equilibrium in the game.

The existence of a quasi-polynomial algorithm may seem as promising evidence that a polynomial algorithm exists. This goal has drawn intense work [DMP06, KPS06, DMP07, BBM07, TS07], which made encouraging progress and culminated (so far) with a polynomial time algorithm that computes a 0.3393-equilibrium [TS07]. All these algorithms, with the sole exception of [TS07], rely on the aforementioned approach of proving the existence of a small support ε-equilibrium via sampling, and then finding such an equilibrium using exhaustive search in conjunction with other algorithmic tools (such as linear programming).

While progress on the approximation side remains steady, the other side of computational lower bounds has resisted attempts to exclude PTAS by extending the known hardness results to approximations (beyond FPTAS), either for any equilibrium or for an objective-maximizing one. The reason for this difficulty might be the aforementioned quasi-polynomial time algorithms, due to which it is less plausible that we can prove hardness of approximation based on NP-hardness or PPAD-hardness for the corresponding question.

In this paper we give the first negative evidence for the existence of a PTAS for the objective-maximizing question. Since NP-hardness is out of the question, we design a reduction from the well known problem of finding a hidden (planted) clique in a random graph. The latter choice is non-standard, as the problem appears to be hard on the average rather than in a worst-case sense. However, in several respects it is an ideal choice. First, it admits a straightforward quasi-polynomial time algorithm. Second, the average-case nature of the problem is particularly suited for constructing games with a highly regular structure, which will be important in our reduction.

The hidden clique problem. In this problem, the input is a graph on n vertices drawn at random from the following distribution $G_{n,1/2,k}$: pick a random graph from $G_{n,1/2}$ and plant in it a clique of size $k = k(n)$.[2] The goal is to recover the planted clique (in polynomial time), with probability at least (say) $1/2$ over the input distribution. Note that the clique

is hidden in the sense that its "location" is adversarial and not known to the algorithm (but independent of the random graph, e.g. of its degrees). In a random graph, the maximum size of a clique is, with high probability, roughly $2 \log n$, and thus when the parameter k is larger than this value, the planted clique will be, with high probability, the unique maximum clique in the graph, and the problem's goal is simply to find the maximum clique in the graph (see Lemma 2.2 for details). The problem was suggested independently by Jerrum [Jer92] and by Kučera [Kuč95].

It is not difficult to see that the hidden clique problem becomes only easier as k gets larger, and the best polynomial-time algorithm to date, due to Alon, Krivelevich and Sudakov [AKS98], solves the problem whenever $k \geq \Omega(\sqrt{n})$ (see also [FK00]). Improving over this bound is a well-known open problem, and certain algorithmic techniques provably fail this task, namely the Metropolis process [Jer92] and the Lovász-Schrijver hierarchy of relaxations [FK03]. The hidden clique problem can be easily solved in quasi-polynomial time $n^{O(\log n)}$; for the most difficult regime $k = O(\log n)$, this is obviously true even in a worst-case sense.

1.1 Our Results We relate the worst case hardness of finding an approximate equilibrium to that of solving the hidden clique problem, formally stated as follows.

THEOREM 1.1. *There are constants $\hat{\varepsilon}, \hat{c} > 0$ such that the following holds. If there is a polynomial-time algorithm that finds in an input two-player game an $\hat{\varepsilon}$-equilibrium whose social-welfare is no more than $\hat{\varepsilon}$ smaller than the maximum social-welfare of an equilibrium in this game, then there is a (randomized) polynomial-time algorithm that solves the hidden clique problem for $k = \hat{c} \log n$ with high probability.*

We remark that our proof is actually shown for the special case of symmetric two player games (see Section 2 for definitions), which makes the result only stronger (since this is a hardness result). We make no attempt to optimize various constants in the proofs.

1.2 Related Work There are complexity classes that attempt to capture problems requiring running time $n^{O(\log n)}$, see [PY96] and references in Section 5 therein. It is plausible that our approach of relying on the hidden clique problem, may be used to prove hardness of approximation for problems mentioned in [PY96], such as the VC dimension of a 0-1 matrix, and the minimum dominating set in a tournament graph.

Average-case hardness. The hidden clique problem is related to the assumption that refuting 3SAT is hard on average (for low-density formulas), which was

---

[2]$G_{n,p}$ denotes the distribution over graphs on n vertices generated by placing an edge between every pair of vertices independently with probability p.

used by Feige [Fei02] to derive constant factor hardness of approximation for several problems, such as minimum bisection, dense k-subgraph and maximum bipartite clique. His results may be interpreted as evidence that approximation within the same (or similar) factor is actually NP-hard, which is a plausible possibility but not known to date. In fact, the random 3SAT refutation conjecture may be viewed [Feige, private communication] as a scaled version of the hidden clique problem, so that straightforward algorithms based on enumeration require exponential, rather than quasi-polynomial, running time. It is not difficult to see that some of the combinatorial optimization problems addressed in [Fei02] are also hard to approximate under the assumption that the hidden clique problem cannot be solved in polynomial time. Consider for example the dense k-subgraph problem; the hidden clique graph itself obviously contains a k-vertex subgraph of maximum density, while any algorithm that is likely to find in it a sufficiently dense k-vertex subgraph can be used to find the planted clique, see e.g. Lemma 5.3. The argument for the maximum bipartite clique problem is similar.

It is worth noting that the assumption that the hidden clique problem is hard was used in a few other contexts, including for cryptographic applications [JP00], and for hardness of testing almost k-wise independence [AAK+07]. The decision version of the hidden clique problem, namely, to distinguish between the distributions $G_{n,1/2,k}$ and $G_{n,1/2}$, is attributed to M. Saks in [KV02, Section 5].

Computing equilibria. The last decade has seen a vast literature on computational aspects of equilibria in various scenarios, including for example graphical games (where the direct interaction between players is limited by a graph structure), succinct games (where the payoffs can be represented succinctly, e.g. due to strong symmetries or a combinatorial structure), and markets (where sellers and buyers interact via prices). For more details, we refer the reader to the excellent and timely surveys [Pap08, Rou08] and the many references therein. More concretely for Nash equilibrium in bimatrix games, see the recent surveys [Pap07, Spi08].

In general, the problems of finding any equilibrium and that of finding an equilibrium that maximizes some objective need not have the same (runtime) complexity, although certain algorithmic techniques may be effective to both. As mentioned earlier, this indeed happens for ε-equilibrium in two-player games, when employing random sampling combined with quasi-polynomial exhaustive search. Another example is the use of the discretization method [KLS01], which was recently used in [DP08] to find an ε-equilibrium in anonymous games

with fixed number of strategies, but actually extends to the value-maximization version [Daskalakis, private communication]. Yet another example is the algorithm of [EGG07] for graphical games on bounded degree trees and whose best response policy [KLS01] has polynomial size.

2 Preliminaries

Let $[n] = \{1, 2, ..., n\}$. An event $\mathcal{E}$ is said to occur with high probability if $\Pr[\mathcal{E}] \geq 1 - 1/n^{\Omega(1)}$; the value of n will be clear from the context. An algorithm is called *efficient* if it runs in polynomial time $n^{O(1)}$.

2.1 Nash Equilibria in games
In the sequel, we restrict our attention to symmetric games, hence our definition assumes square matrices for the payoff. A (square) *two-player bi-matrix game* is defined by two payoff matrices $R, C \in \mathbb{R}^{n \times n}$, such that if the row and column players choose pure strategies $i, j \in [n]$, respectively, the payoff to the row and column players are $R(i, j)$ and $C(i, j)$, respectively. The game is called *symmetric* if $C = R^{\top}$.

A *mixed strategy* for a player is a distribution over pure strategies (i.e. rows/columns), and for brevity we may refer to it simply as a strategy. An ε-approximate Nash equilibrium is a pair of mixed strategies (x, y) such that

$$\forall i \in [n], \quad e_i^{\top} R y \leq x^{\top} R y + \varepsilon,$$
$$\forall j \in [n], \quad x^{\top} C e_j \leq x^{\top} C y + \varepsilon.$$

Here and throughout, e_i is the i-th standard basis vector, i.e. 1 in i-th coordinate i, and 0 in all other coordinates. If $\varepsilon = 0$, the strategy pair is called a *Nash equilibrium* (NE). The definition immediately implies the following.

PROPOSITION 2.1. *For an ε-equilibrium (x, y), it holds that for every mixed strategies $\tilde{x}, \tilde{y}$,*

$$\tilde{x}^{\top} R y \leq x^{\top} R y + \varepsilon,$$
$$x^{\top} R \tilde{y} \leq x^{\top} R y + \varepsilon.$$

As we are concerned with an additive notion of approximation, we assume that the entries of the matrices are in the range $[0, M]$, for M which is a constant independent of all the other parameters. Our results easily translate to the case $M = 1$ by scaling all payoffs.

Consider a pair of strategies (x, y). We call the *payoff* of the row player $x^{\top} R y$ (this is actually the expected payoff), and similarly for the column player. The *value* of an (approximate) equilibrium for the game is the average of the payoffs of the two players. Recall that *social-welfare* is the total payoff of the two players, and thus equals twice the value.

2.2 The hidden clique problem Recall that in this problem the input is drawn at random from the distribution $G_{n,1/2,k}$. Intuitively, the problem only becomes easier as k gets larger, at least in our regime of interest, namely $k \geq c_0 \log n$ for sufficiently large constant $c_0 > 0$. This intuition can be made precise as follows.

LEMMA 2.2. *Suppose there are a constant $c_1 > 0$ and a polynomial time algorithm such that given an instance of the hidden clique problem with $k \geq c_1 \log n$ finds a clique of size $100 \log n$ with probability at least $1/2$. Then there exists a constant $c_0 > 0$ and a randomized polynomial time algorithm that solves the hidden clique problem for every $k \geq c_0 \log n$.*

This lemma is probably known, but since we could not provide a reference, we prove it in Appendix A, essentially using ideas from [AKS98] and [McS01]. Notice that due to potential correlations, one cannot employ simple techniques that are useful in worst-case instances, such as repeatedly finding and removing from the input graph a clique of size $100 \log n$ (using the assumed algorithm), not to mention of course that the assumed algorithm only succeeds with probability $1/2$ (and repeating it need not amplify the success probability).

3 The Reduction

We prove Theorem 1.1 by reducing the hidden clique problem to the Nash equilibrium problem. That is, given an input instance of the hidden clique problem we construct a two-player game such that with high probability (over the randomness in our construction and in the hidden clique instance), a high-value approximate equilibrium leads, in polynomial time, to a solution to the hidden clique instance.

Techniques. Our construction is motivated by an observation of Halperin and Hazan [HH05], that for every graph, the quadratic form corresponding to the graph's adjacency matrix, when considered over the the unit simplex (i.e. all probability distributions over $[n]$), is maximized exactly at a (normalized) incidence vector of a maximum clique in the graph. We rely on this observation, as one portion of the game we construct is exactly the adjacency matrix of the hidden clique instance. However, this is not enough to obtain a suitable Nash equilibrium instance.

First, an equilibrium is a bi-linear form rather than a quadratic form, hence the results of [HH05] are not directly applicable. We thus use (mainly in Lemma 5.2 below) the special properties of an approximate equilibrium to prove a relationship of similar flavor

between bi-linear forms on the adjacency matrix and large cliques (or actually dense subgraphs) in the graph.

Second, a simple use of the adjacency matrix yields a very small gap (between vectors corresponding to a clique and those that do not) that is by far insufficient to rule out a PTAS. To boost this gap we use an idea of Feder, Nazerzadeh, and Saberi [FNS07] to eliminate from the game all equilibria of small support (cardinality at most $O(\log n)$).

The construction. Let $\hat{\varepsilon}, \hat{c}$ and M, c_1, c_2 be constants to be defined shortly. Given an instance $G \in G_{n,1/2,k}$ of the hidden clique problem, consider the two-player game defined by the following payoff matrices (for the row-player and the column player, respectively):

$$ R = \begin{pmatrix} A & -B^\top \\ B & \mathbf{0} \end{pmatrix} \quad ; \quad C = \begin{pmatrix} A & B^\top \\ -B & \mathbf{0} \end{pmatrix} $$

The matrices R, C are of size $N \times N$ for $N = n^{c_1}$. These matrices are constructed from the following blocks.

1. The upper left $n \times n$ block in both R, C is the adjacency matrix of G with self loops added.

2. The lower right block $\mathbf{0}$ in both R, C is the all zeros matrix of size $(N - n) \times (N - n)$.

3. All other entries are set via an $(N - n) \times n$ matrix B whose entries are set independently at random to be

$$ B_{i,j} = \begin{cases} M & \text{with probability } \frac{3}{4} \cdot \frac{1}{M}; \\ 0 & \text{otherwise.} \end{cases} $$

Notice that the game is symmetric, i.e. $C = R^\top$, and that outside the upper left block A, the game is zero-sum.

Choice of parameters. We set the parameters in our construction as follows.

- $M = 12$;
- $c_2 = 200$;
- $c_1 = 2 + c_2 \log(4M/3)$; (recall $N = n^{c_1}$)
- $\hat{c} = 32M^2(c_1 + 2)$; and
- $\hat{\varepsilon} = 1/50M$.

Note that these parameters do not depend on k, the size of the hidden clique, and in particular, k can be much larger than $\log n$. The reason is that the algorithm in our reduction focuses on finding a clique of size $100 \log n$, which clearly must have a large overlap with the planted clique, and from which the entire planted clique can be found using standard techniques (Lemma 2.2).

As is standard in computational complexity, we prove Theorem 1.1 by analyzing the completeness and soundness of the reduction.

In the sequel, when we say with high probability, it means over the choice of G from $G_{n,1/2,k}$ over the construction of the game (namely the randomness in B), and over the coin tosses of our algorithms. We note however, that most of our algorithms are deterministic; the only exception is Lemma 2.2 (and of course the algorithms that invoke it).

4 Completeness

LEMMA 4.1. *With high probability, the game above has an equilibrium of value 1.*

Proof. Consider the distributions (mixed strategies) $x = y$ which are a uniform distribution over the strategies corresponding to the planted k-clique, $x_i = \frac{1}{k}$ if i is in the planted clique, and $x_i = 0$ otherwise. The value of this strategy set is $\frac{1}{2}x^\top(R+C)y = 1$. We shall prove that with high probability this is an equilibrium.

Consider without loss of generality the row player, and observe that her payoff when she plays x is exactly 1. We need to show that she cannot obtain a larger payoff by playing instead any strategy $i \in [N]$. For $i \leq n$, her new payoff is at most the largest entry in A, i.e. 1. For each $i > n$, her new payoff is the average of k distinct values in B, which is highly concentrated around its mean $3/4$. Formally, we use the following Chernoff-Hoeffding bound.

THEOREM 4.2. ([HOE63]) *Let $X_1, ..., X_m$ be independent random variables bounded by $|X_j| \leq C$, and let $\bar{X} = \frac{1}{m}\sum_j X_j$. Then for all $t > 0$,*

$$\Pr[\bar{X} - \mathbf{E}[\bar{X}] \geq t] \leq \exp(-\tfrac{mt^2}{2C^2}).$$

In our case, the variables satisfy $|X_j| \leq M$ and $\mathbf{E}[X_j] = \frac{3}{4}$, and $\bar{X}$ is the payoff of playing strategy $i > n$ (when the other player still plays $x = y$). We thus obtain

$$\Pr[\bar{X} \geq 1] \leq \exp(-\tfrac{k}{32M^2}).$$

By a union bound over all strategies $i > n$ we have that the probability there exists a strategy $i > n$ with payoff larger than 1 is at most $(N-n) \cdot e^{-\frac{k}{32M^2}} \leq n^{c_1 - \frac{\hat{c}}{32M^2}} = 1/n^2$, where the last inequality follows by our choice of c_1. This completes the proof of Lemma 4.1. ∎

5 Soundness

To complete the proof of Theorem 1.1, we shall show that with high probability, every $\hat{\varepsilon}$-approximate equilibrium with value $\geq 1 - \hat{\varepsilon}$ in the game can be used to find the hidden clique efficiently. We do this in three

steps, using the three lemmas below. We note that the second and main step (Lemma 5.2) can be made to hold without any assumption on the value of the equilibrium (details omitted in this abstract).

For our purpose, a bipartite subgraph is two equal-size subsets $V_1, V_2 \subseteq V(G)$ (not necessarily disjoint); its density is the probability that random $v_1 \in V_1, v_2 \in V_2$ are connected by an edge, in the input graph with self-loops added.

LEMMA 5.1. *With high probability, given an $\hat{\varepsilon}$-equilibrium in the game with value $\geq 1 - \hat{\varepsilon}$, we can efficiently compute a $(4M\hat{\varepsilon})$-equilibrium that is supported only on A and has value $\geq 1 - \hat{\varepsilon}$.*

LEMMA 5.2. (MAIN) *With high probability, given a $(4M\hat{\varepsilon})$-equilibrium supported only over the matrix A and with value $\geq 1 - \hat{\varepsilon}$, we can efficiently find a bipartite graph of size $c_2 \log n$ and density $\frac{3}{5}$ in the input graph.*

LEMMA 5.3. *With high probability, given a bipartite subgraph of size $c_2 \log n$ and density $\geq \frac{3}{5}$ in the input graph, we can efficiently find the entire planted hidden k-clique.*

5.1 Proof of Lemma 5.1
The following two claims are stated with a general parameter $\delta > 0$, although we will later use them only with a specific value $\delta = \hat{\varepsilon}$.

CLAIM 5.4. *In every pair of mixed strategies achieving value $\geq 1 - \delta$, at most δ of the probability mass of each player resides on strategies not in $[n]$.*

Proof. The contribution to the value of the equilibrium from outside the upper left block is 0, because over there the game is zero-sum. Inside that block the two players receive identical payoffs, which are according to A and thus upper bounded by 1. Thus,

$$1 - \delta \qquad \leq \tfrac{1}{2}x^\top(R+C)y$$
$$= \sum_{i,j \in [n]} x_i y_j A_{ij} \quad \leq \Big(\sum_{i \in [n]} x_i\Big)\Big(\sum_{j \in [n]} y_j\Big),$$

and it immediately follows that both $\sum_{i \in [n]} x_i$ and $\sum_{j \in [n]} y_j$ are at least $1 - \delta$. ∎

CLAIM 5.5. *Given an $\hat{\varepsilon}$-equilibrium where at most δ of each player's probability mass reside on strategies not in $[n]$, we can find an $(\hat{\varepsilon} + 3M\delta)$-equilibrium that is supported only on A and whose value is at least as large.*

Proof. Given an $\hat{\varepsilon}$-equilibrium (x, y), we obtain a new equilibrium $(\tilde{x}, \tilde{y})$ by restricting each player's support to $[n]$, i.e. removing strategies not in $[n]$ and scaling to obtain a probability distribution. Since the game

is zero-sum outside of A, removing strategies not in A does not change the value, and since the entries in A are nonnegative, the scaling operation can only increase the value, i.e. $\tilde{x}^\top(R+C)\tilde{y} \geq x^\top(R+C)y$.

To bound defections, consider without loss of generality the row player. First, her payoff when defecting to strategy $i \in [N]$ does not change much, i.e. $|e_i^\top R\tilde{y} - e_i^\top Ry| \leq M\delta$, because the total mass of y moved around is at most δ, and different entries in R differ by at most M. Furthermore, her payoff in the new equilibrium does not change much, i.e. $|\tilde{x}^\top R\tilde{y} - x^\top Ry| \leq 2M\delta$, because at most 2δ of the total probability mass of x and y was moved. $\blacksquare$

Lemma 5.2 now follows immediately from the two claims above by setting $\delta = \hat{\varepsilon}$.

5.2 Proof of Lemma 5.2 To prove this lemma, we first need the following structural claim.

CLAIM 5.6. *With high probability, in every $1/2$-approximate equilibrium supported only over the matrix A, the total probability mass every player places on every set of $c_2 \log n$ pure strategies is $\leq 2/M$.*

Proof. For convenience, denote $d = c_2 \log n$. Suppose that one of the players, say the column player, has measure of more than $\frac{2}{M}$ on a support of size $\leq d$. Let us compute the probability that there exists a row in B, in which the d corresponding entries all have a value of M. If this event happens, then we do not have an ε-equilibrium, since the row player can defect to this particular row, to obtain payoff $\geq \frac{2}{M} \cdot M = 2$, while her current payoff is ≤ 1. The probability this event happens for a single row is obviously p^d for $p = \frac{3}{4M}$, and very small. But we have $N - n$ rows, and they are independent. Thus, the probability that no row has a streak of M's in the particular d columns is at most

$$
\begin{aligned}
(1 - p^d)^{N-n} &\leq \exp(-p^d N/2) \\
&= \exp(-n^{c_1 - c_2 \log \frac{4M}{3}}/2) \\
&\leq \exp(-n^2/2)
\end{aligned}
$$

where the last inequality is by our choice of c_1. Hence with probability $1 - e^{-n^2}$ there is such a row, and hence this cannot be an equilibrium.

We now need to rule out all possible subsets sets of size d. There are $\binom{n}{d} \leq n^d$ such subsets, and each one cannot be an equilibrium with probability $\geq 1 - e^{-n^2/2}$. We can rule out all of them by a union bound, since $n^d \cdot e^{-n^2/2} \leq e^{-\Omega(n^2)}$. $\blacksquare$

Proof. [Proof of Lemma 5.2] Let (x, y) be such an $(4M\hat{\varepsilon})$-equilibrium. Define $T = \{j \in \mathrm{supp}(y) :$

$x^\top Ae_j \geq \frac{4}{5}\}$, where e_j is the j-th standard basis vector. Observe that T is nonempty, since $x^\top Ay \geq 1 - \hat{\varepsilon}$. Furthermore, the total probability mass outside of T is $\sum_{j \notin T} y_j \leq \frac{1}{2}$, or otherwise moving this probability mass into T would increase the column player's payoff by more than $\frac{1}{2}(1 - \hat{\varepsilon} - \frac{4}{5}) > \frac{1}{12} > 4M\hat{\varepsilon}$, by our choice of $\hat{\varepsilon}$, contradicting our assumption of an approximate equilibrium (specifically Proposition 2.1). Since $\sum_{j \in T} y_j \geq \frac{1}{2} > \frac{2}{M}$ and $4M\hat{\varepsilon} \leq 1/2$, we have by Claim 5.6 that $|T| \geq c_2 \log n$. To get size exactly $c_2 \log n$, we can just take an arbitrary subset of T. Thus, denoting by u_T the uniform distribution on T, the pair (x, u_T) satisfies

$$x^\top Au_T \geq \frac{4}{5}.$$

Now define $S = \{i \in \mathrm{supp}(x) : e_i^\top Au_T \geq \frac{3}{5}\}$. Its total probability mass must be $\sum_{i \in S} x_i > \frac{2}{M}$, as otherwise $x^\top Au_T \leq \frac{2}{M} \cdot 1 + (1 - \frac{2}{M}) \cdot \frac{3}{5} < \frac{4}{5}$, by our choice of $M \geq 10$. By claim 5.6 we then have $|S| > c_2 \log n$. Let u_S be the uniform distribution over the set S. Then $u_S^\top Au_T \geq \frac{3}{5}$, i.e. S, T define a bipartite subgraph of size $\geq c_2 \log n$ and density $\geq \frac{3}{5}$. To get size exactly $c_2 \log n$, we can just take an arbitrary subset of S. $\blacksquare$

5.3 Proof of Lemma 5.3 To prove this lemma we separate out first how to extract a clique of logarithmic size.

CLAIM 5.7. *With high probability, given a bipartite subgraph of size $c_2 \log n$ and density $\geq \frac{3}{5}$ in the input graph, we can efficiently find a clique of size $100 \log n$.*

Proof. Let $S, T \subset V(G)$ be the two sets forming the bipartite subgraph, and let $W \subset V(G)$ denote the vertices of the planted clique, i.e. $|W| = k$. A straightforward union bound shows that with high probability at least $\frac{1}{20}$ of the vertices in S must lie in the planted clique. Indeed, the number of sets $S, T \subset V(G)$ with $|S| = |T| = c_2 \log n$ and $|S \cap W| < \frac{1}{20}|S|$ is at most $\binom{n}{|S|}|T| < n^{2c_2 \log n}$. For each of them, the density between $S \setminus W$ and T is essentially the average of $\Theta(c_2 \log n)^2$ Bernoulli random variables with expectation $1/2$ (except that some variables may be included twice, and except for at most $c_2 \log n$ terms corresponding to self loops). Thus, by the Chernoff bound above, $\Pr[\mathrm{density}(S \cap W, T) \geq 0.55] \leq \exp(-\Omega(c_2 \log n)^2)$. and we get that with high probability,

$$
\begin{aligned}
\mathrm{density}(S, T) &= \\
&= \frac{|S \cap W|}{|S|}\mathrm{density}(S \cap W, T) + \frac{S \setminus W}{|S|}\mathrm{density}(S \setminus W, T) \\
&< \frac{3}{5}.
\end{aligned}
$$

Furthermore, we can take a union bound over all choices of such S, T by choosing c_2 as a sufficiently large constant.

Thus, given S, T we can try all subsets of S of size $\frac{1}{20}c_2 \log n$ by exhaustive search (the number of such sets is $n^{O(c_2)} = n^{O(1)}$), and find the largest subset S' that forms a clique in G. By the above analysis, with high probability $|S'| \geq \frac{c_2}{20}\log n = 100 \log n$. ∎

Lemma 5.3 now follows by combining the above claim with Lemma 2.2. This completes also the proof of Theorem 1.1.

References

[AAK$^+$07] N. Alon, A. Andoni, T. Kaufman, K. Matulef, R. Rubinfeld, and N. Xie. Testing k-wise and almost k-wise independence. In *39th Annual ACM Symposium on Theory of Computing*, pages 496–505, New York, NY, USA, 2007. ACM.

[AKS98] N. Alon, M. Krivelevich, and B. Sudakov. Finding a large hidden clique in a random graph. *Random Structures Algorithms*, 13(3-4):457–466, 1998.

[BBM07] H. Bosse, J. Byrka, and E. Markakis. New algorithms for approximate nash equilibria in bimatrix games. In *WINE 2007*, volume 4858 of *Lecture Notes in Computer Science*, pages 17–29. Springer, 2007.

[CDT06] X. Chen, X. Deng, and S.-H. Teng. Computing nash equilibria: Approximation and smoothed complexity. In *47th Annual IEEE Symposium on Foundations of Computer Science*, pages 603–612, Washington, DC, USA, 2006. IEEE Computer Society.

[CS03] V. Conitzer and T. Sandholm. Complexity results about nash equilibria. In *18th International Joint Conference on Artificial Intelligence*, pages 765–771, 2003.

[DGP06] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a nash equilibrium. In *38th Annual ACM Symposium on Theory of computing*, pages 71–78, New York, NY, USA, 2006. ACM.

[DMP06] C. Daskalakis, A. Mehta, and C. H. Papadimitriou. A note on approximate nash equilibria. In *WINE 2006*, volume 4286 of *Lecture Notes in Computer Science*, pages 297–306. Springer, 2006.

[DMP07] C. Daskalakis, A. Mehta, and C. Papadimitriou. Progress in approximate nash equilibria. In *8th ACM conference on Electronic commerce*, pages 355–358, New York, NY, USA, 2007. ACM.

[DP08] C. Daskalakis and C. H. Papadimitriou. Discretized multinomial distributions, covers, and nash equilibria in anonymous games. To appear in FOCS, 2008.

[EGG07] E. Elkind, L. A. Golberg, and P. W. Goldberg. Computing good nash equilibria in graphical games. In *8th ACM conference on Electronic commerce*, pages 162–171, New York, NY, USA, 2007. ACM.

[Fei02] U. Feige. Relations between average case complexity and approximation complexity. In *34th annual ACM Symposium on Theory of Computing*, pages 534–543. ACM, 2002.

[FK00] U. Feige and R. Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures Algorithms*, 16(2):195–208, 2000.

[FK03] U. Feige and R. Krauthgamer. The probable value of the Lovász-Schrijver relaxations for maximum independent set. *SIAM J. Comput.*, 32(2):345–370, 2003.

[FNS07] T. Feder, H. Nazerzadeh, and A. Saberi. Approximating nash equilibria using small-support strategies. In *8th ACM conference on Electronic commerce*, pages 352–354, New York, NY, USA, 2007. ACM.

[GZ89] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

[HH05] E. Halperin and E. Hazan. HAPLOFREQ - estimating haplotype frequencies efficiently. In *9th RECOMB*, volume 3500 of *Lecture Notes in Computer Science*, pages 553–568. Springer, 2005.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[Jer92] M. Jerrum. Large cliques elude the Metropolis process. *Random Structures Algorithms*, 3(4):347–359, 1992.

[JP00] A. Juels and M. Peinado. Hiding cliques for cryptographic security. *Des. Codes Cryptogr.*, 20(3):269–280, 2000.

[KLS01] M. J. Kearns, M. L. Littman, and S. P. Singh. Graphical models for game theory. In *17th Conference in Uncertainty in Artificial Intelligence*, pages 253–260. Morgan Kaufmann Publishers Inc., 2001.

[KPS06] S. C. Kontogiannis, P. N. Panagopoulou, and P. G. Spirakis. Polynomial algorithms for approximating nash equilibria of bimatrix games. In *WINE 2006*, volume 4286 of *Lecture Notes in Computer Science*, pages 286–296. Springer, 2006.

[Kuč95] L. Kučera. Expected complexity of graph partitioning problems. *Discrete Appl. Math.*, 57(2-3):193–212, 1995.

[KV02] M. Krivelevich and V. H. Vu. Approximating the independence number and the chromatic number in expected polynomial time. *J. Comb. Optim.*, 6(2):143–155, 2002.

[LMM03] R. J. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In *4th ACM conference on Electronic commerce*, pages 36–41, New York, NY, USA, 2003. ACM.

[McS01] F. McSherry. Spectral partitioning of random graphs. In *42nd IEEE symposium on Foundations of Computer Science*, page 529, Washington, DC, USA, 2001. IEEE Computer Society.

[Pap07] C. H. Papadimitriou. The complexity of finding nash equilibria. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 2, pages 29–51. Cambridge University Press, 2007.

[Pap08] C. H. Papadimitriou. The search for equilibrium concepts. In *SAGT*, volume 4997 of *Lecture Notes in Computer Science*, pages 1–3. Springer, 2008.

[PY96] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the v-c dimension. *J. Comput. Syst. Sci.*, 53(2):161–170, 1996.

[Rou08] T. Roughgarden. An algorithmic game theory primer. To appear in TCS, invited survey, 2008.

[Spi08] P. G. Spirakis. Approximate equilibria for strategic two person games. In *SAGT 2008*, volume 4997 of *Lecture Notes in Computer Science*, pages 5–21. Springer, 2008.

[TS07] H. Tsaknakis and P. G. Spirakis. An optimization approach for approximate nash equilibria. In *WINE 2007*, volume 4858 of *Lecture Notes in Computer Science*, pages 42–56. Springer, 2007.

A Deferred Proof from Section 2

Proof. [Proof of Lemma 2.2] Suppose there exists a polynomial time algorithm $\mathcal{A}^*$ that, given the hidden clique problem with $k \geq c_1 \log n$, finds a clique of size $100 \log n$. We prove that there exists a (randomized) polynomial time algorithm that solves the hidden clique problem exactly for every $k \geq c_0 \log n$, where $c_0 = 2tc_1$ for a sufficiently large t to be determined later. The algorithm is composed of two stages.

Stage 1. Randomly partition the graph vertices into t parts. In each part, the expected number of vertices from the planted clique vertices is $k/t \geq \frac{c_0 \log n}{t} = 2c_1 \log n$. Furthermore, using Chernoff bounds it is easy to show that with probability $> 7/8$, every part contains at least $c_1 \log n$ vertices from the hidden clique. In our analysis we shall assume henceforth that this event does not occur.

In each part separately, first complete it into an instance of hidden clique of size exactly n, by adding $n - n/t$ vertices and connecting all new potential edges with probability $\frac{1}{2}$. Then apply the polynomial time algorithm $\mathcal{A}^*$. Observe that each part is distributed exactly as a hidden clique instance, and by our assumption its hidden clique size is large enough that algorithm $\mathcal{A}^*$ succeeds, with probability $\geq 1/2$, in finding a clique of size at least $100 \log n$. Since the different parts are independent, the probability that $\mathcal{A}^*$ succeeds in one or less parts is $\leq (t+1)2^{-t} < 1/8$ for, say, $t = 6$. In our analysis we shall assume henceforth that this event does not occur, i.e. $\mathcal{A}^*$ succeeds in at least two parts.

In each part where $\mathcal{A}^*$ succeeds, we may assume that the clique size is exactly $100 \log n$ by removing arbitrary vertices from it. Since even in the entire graph the maximum clique size in the random portion (i.e. not not using the planted clique) is with high probability roughly $2 \log n$. In our analysis we shall assume henceforth that this event occurs, in every part where $\mathcal{A}^*$ succeeds, at least $97 \log n$ among the $100 \log n$ vertices of the clique found belong to the planted clique.

Stage 2. In each part i apply the following. Identify *another* part $j \neq i$ where $\mathcal{A}^*$ succeeded in finding a clique of size $100 \log n$. Select the vertices in the part i whose degree towards the clique found in part j is at least $97 \log n$. Call these vertices Q_i and report all selected vertices, i.e. $Q = \cup_i Q_i$. (If $\mathcal{A}^*$ did not succeed in at least two parts, report fail.)

To analyze this stage, observe that a vertex v from part i that belongs to the planted clique must have degree at least $97 \log n$ towards the clique found in another part j, and thus belongs to Q_i. On the other hand, for a vertex v in part i that does not belong to the planted clique, the expected degree towards any fixed subset of $100 \log n$ vertices in part j is $50 \log n$. Notice that the chosen part j and the clique found in it are completely independent of the edges connecting different parts, because they are determined only by the edges internal to the different parts (possibly in a complicated way, e.g. how to break ties if $\mathcal{A}^*$ succeeds in more than two parts). Thus, the probability that v has degree at least $97 \log n$ towards the corresponding clique in part j is, using the Chernoff bound, $\leq 1/n^2$. By a union bound over all vertices we get that with high probability all such vertices do not belong to Q_i. Combining this with the events mentioned earlier, we get by a union bound that $Q = \cup_i Q_i$ contains exactly all the hidden clique vertices with probability at least $2/3$. ∎