On the Communication Complexity of Approximate Nash Equilibria

Paul W. Goldberg[1,⋆] and Arnoud Pastink[2]

[1] Department of Computer Science, University of Liverpool, U.K.
P.W.Goldberg@liverpool.ac.uk
[2] Utrecht University, Department of Information and Computing Science,
P.O. Box 80089, 3508TB Utrecht, The Netherlands
arnoudpastink@gmail.com

Abstract. We study the problem of computing approximate Nash equilibria, in a setting where players initially know their own payoffs but not the payoffs of other players. In order for a solution of reasonable quality to be found, some amount of communication needs to take place between the players. We are interested in algorithms where the communication is substantially less than the contents of a payoff matrix, for example logarithmic in the size of the matrix. At one extreme is the case where the players do not communicate at all; for this case (with 2 players having $n \times n$ matrices) ϵ-Nash equilibria can be computed for $\epsilon = 3/4$, while there is a lower bound of slightly more than 1/2 on the lowest ϵ achievable. When the communication is polylogarithmic in n, we show how to obtain $\epsilon = 0.438$. For one-way communication we show that $\epsilon = 1/2$ is the exact answer.

1 Introduction

Algorithmic game theory is concerned not just with properties of a solution concept, but also how that solution can be obtained. It is considered desirable that the outcome of a game should be "easy to compute", and in that respect the PPAD-completeness results of [6,2] are interpreted as a "complexity-theoretic critique" of Nash equilibrium. Following those results, a line of work addressed the problem of computing ϵ-Nash equilibrium, where $\epsilon > 0$ is a parameter that bounds a player's incentive to deviate, in a solution. Thus, ϵ-Nash equilibrium imposes a weaker constraint on how players are assumed to behave, and an exact Nash equilibrium is obtained for $\epsilon = 0$.

Besides the existence of a fast algorithm, it is also desirable that a solution should be obtained by a process that is simple and decentralised, since that is likely to be a better model for how players in a game may eventually reach a solution. In that respect, most of the known efficient algorithms for computing ϵ-Nash equilibria are not entirely satisfying. They take as input the payoff matrices

⋆ Supported by EPSRC Grant EP/G069239/1 "Efficient Decentralised Approaches in Algorithmic Game Theory".

M. Serna (Ed.): SAGT 2012, LNCS 7615, pp. 192–203, 2012.

and output the approximate Nash equilibrium. If we try to translate such an algorithm into real life, it would correspond to a process where the players pass their payoffs to a central authority, which returns to them some mixed strategies that have the "low incentive to deviate" guarantee. In this paper we try to model a setting where players perform individual computations and exchange some limited information.

There are various ways in which one can try to model the notion of a decentralised algorithm; here we consider a general approach that has previously been studied in [4,9] in the context of computing exact Nash equilibria. The players begin with knowledge of their own payoffs but not the payoffs of the other players. An algorithm involves communication in addition to computation; to reach an approximate equilibrium, a player usually has to know something about the other players' matrices, but hopefully not all of that information. We study the computation of ϵ-Nash equilibria in this setting, and the general topic is the trade-off between the amount of communication that takes place, and the value of ϵ that can be obtained.

1.1 Definitions

We consider 2-player games, with a *row player* and a *column player*, who both have *n pure strategies.* The game (R, C) is defined by two $n \times n$ *payoff matrices*, R for the row player, and C for the column player. The pure strategies for the row player are his rows and the pure strategies of the column player are her columns. If the row player plays row i and the column player plays column j, the *payoff* for the row player is R_{ij}, and C_{ij} for the column player. For the row player a *mixed strategy* is a probability distribution $\mathbf{x}$ over the rows, and a mixed strategy for the column player is a probability distribution $\mathbf{y}$ over the columns, where $\mathbf{x}$ and $\mathbf{y}$ are column vectors and $(\mathbf{x}, \mathbf{y})$ is a *mixed strategy profile.* The payoffs resulting from these mixed strategies $\mathbf{x}$ and $\mathbf{y}$ are $\mathbf{x}^T R\mathbf{y}$ for the row player and $\mathbf{x}^T C\mathbf{y}$ for the column player.

A *Nash equilibrium* is a pair of mixed strategies $(\mathbf{x}^*, \mathbf{y}^*)$ where neither player can get a higher payoff by playing another strategy assuming the other player does not change his strategy. Because of the linearity of a mixed strategy, the largest gain can be achieved by defecting to a pure strategy. Let $\mathbf{e}_i$ be the vector with a 1 at the ith position and a 0 at every other position. Thus a Nash equilibrium $(\mathbf{x}^*, \mathbf{y}^*)$ satisfies

$$\forall i = 1 \cdots n \quad \mathbf{e}_i^T R\mathbf{y}^* \leq (\mathbf{x}^*)^T R\mathbf{y}^* \text{ and } (\mathbf{x}^*)^T C\mathbf{e}_i \leq (\mathbf{x}^*)^T C\mathbf{y}^*$$

We assume that the payoffs of R and C are between 0 and 1, which can be achieved by rescaling. An *ϵ-approximate Nash equilibrium* (or, ϵ-Nash equilibrium) is a strategy pair $(\mathbf{x}^*, \mathbf{y}^*)$ such that each player can gain at most ϵ by unilaterally deviating to a different strategy. Thus, it is $(\mathbf{x}^*, \mathbf{y}^*)$ satisfying

$$\forall i = 1 \cdots n \quad \mathbf{e}_i^T R\mathbf{y}^* \leq (\mathbf{x}^*)^T R\mathbf{y}^* + \epsilon \text{ and } (\mathbf{x}^*)^T C\mathbf{e}_i \leq (\mathbf{x}^*)^T C\mathbf{y}^* + \epsilon$$

We say that the *regret* of a player is the difference between his payoff and the payoff of his best response.

The *support* of a mixed strategy $\mathbf{x}$, denoted by Supp($\mathbf{x}$), is the set of pure strategies that are played with non-zero probability by $\mathbf{x}$.

The communication model: Each player $p \in \{r, c\}$ has an algorithm $\mathcal{A}_p$ whose initial input data is p's $n \times n$ payoff matrix. Communication proceeds in a number of rounds, where in each round, each player may send a single bit of information to the other player. During each round, each player may also carry out a polynomial (in n) amount of computation. (One could alternatively omit the restriction to polynomial computation. Our lower bounds on communication requirement do not depend on computational limits.) At the end, each player p outputs a mixed strategy $\mathbf{x}_p$. We aim to design (pairs of) algorithms $(\mathcal{A}_r, \mathcal{A}_c)$ that output ϵ-Nash strategy profiles $(\mathbf{x}_r, \mathbf{x}_c)$, and are economical with the number of rounds of communication.

Notice that given $\Theta(n^2)$ rounds of communication, we can apply any centralised algorithm $\mathcal{A}$ by getting (say) the row player to pass additive approximations of all his payoffs to the column player, who applies $\mathcal{A}$ and passes to the row player the mixed strategy obtained by $\mathcal{A}$ for the row player. (The quality of the ϵ-Nash equilibrium is proportional to the quality of of the additive approximations used.) For this reason we focus on algorithms with many fewer rounds, and we obtain results for logarithmic or polylogarithmic (in n) rounds.

We also consider a restriction to *one-way communication*, where one player may send but not receive information.

1.2 Related Work

Algorithms for Approximate Equilibria. In recent years a number of algorithms have been developed that compute (in polynomial time) ϵ-Nash equilibria for various values of ϵ. This is not a complete overview of all existing algorithms. The algorithm with the best approximation that is known, gives a 0.3393-approximate Nash equilibrium [17]. However, here we mainly use ideas from certain earlier algorithms.

DMP-algorithm: The DMP-algorithm [7] works as follows to achieve a 0.5-approximate Nash equilibrium. The algorithm picks a arbitrary row for the row player, say row i. Let $j \in \text{argmax}_{j'} C_{ij'}$. Let $k \in \text{argmax}_{k'} R_{k'j}$. So j is a pure-strategy best response for the column player to row i and k is a best response strategy for the row player to column j. The strategy pair $(\mathbf{x}^*, \mathbf{y}^*)$ will now be $\mathbf{x}^* = \frac{1}{2}\mathbf{e}_i + \frac{1}{2}\mathbf{e}_k$ and $\mathbf{y}^* = \mathbf{e}_j$. With this strategy pair the row player plays a best response with probability $\frac{1}{2}$ to a pure strategy of the column player and the column player has a pure strategy that is with probability $\frac{1}{2}$ a best response.

The DMP-algorithm is well-adapted to the limited-communication setting. Suppose the row player uses $i = 1$ as his initial choice of row. The column player needs to tell the row player his value of j, a communication of $O(\log n)$ bits. No further communication is needed. Notice moreover that the communication is all one-way; the row player does not need to tell the column player anything.

Subsequent algorithms for computing ϵ-Nash equilibria cannot so easily be adapted to a limited-communication setting, but we can use some of the ideas they develop, to obtain values of ϵ below $\frac{1}{2}$ in this setting.

An algorithm of Bosse et al. [1]: The algorithm presented in [1] can be seen as a modification of the DMP-algorithm and achieves a 0.38197-approximate Nash equilibrium. Instead of a player playing a pure strategy with some positive probability, the algorithm starts with the row player allocating some probability to the row-player strategy $\mathbf{x}$ belonging to the Nash equilibrium of the zero-sum game $(R - C, C - R)$. In solving the zero-sum game efficiently we apply the connection of zero-sum games with linear programming [15,5,11]. If the (mixed) strategy profile $(\mathbf{x}, \mathbf{y})$ that is a Nash equilibrium of $(R-C, C-R)$ gives a 0.38197-approximate Nash equilibrium for (R, C), this solution is used. Otherwise, the column player plays a best response $\mathbf{e}_j$ to $\mathbf{x}$ and the row player plays a mixture of $\mathbf{x}$ and $\mathbf{e}_i$, where $\mathbf{e}_i$ is a best response to the strategy $\mathbf{e}_j$ of the column player. ([1] goes on to improve the worst-case performance to a 0.36395-approximate Nash equilibrium.)

Notice that this algorithm cannot be adapted in a straightforward way to our communication-bounded setup, since it requires a computation using knowledge of both matrices.

Communication Complexity. The "classical" setting of communication complexity is based on the model introduced by Yao in [18]. We will follow the representation in [12]. We have two agents[1], one holding an input $\mathbf{x} \in \{0, 1\}^n$ and the other holding an input $\mathbf{y} \in \{0, 1\}^n$. The objective is to compute $f(\mathbf{x}, \mathbf{y}) \in \{0, 1\}$, a joint function of their inputs. The computation of $f(\mathbf{x}, \mathbf{y})$ is done via a communication protocol $\mathcal{P}$. During the execution of the protocol, the agents send messages to each other. While the protocol has not terminated, the protocol specifies what message the sender should send next, based on the input of the protocol and the communication so far. If the protocol terminates, it will output the value $f(\mathbf{x}, \mathbf{y})$. A communication protocol $\mathcal{P}$ computes f if for every input pair $(\mathbf{x}, \mathbf{y}) \in \{0, 1\}^n \times \{0, 1\}^n$, it terminates with the value $f(\mathbf{x}, \mathbf{y})$ as output.

The communication complexity of a communication protocol $\mathcal{P}$ for computing $f(\mathbf{x}, \mathbf{y})$ is the number of bits sent during the execution of $\mathcal{P}$, which we denote by $CC(\mathcal{P}, f, \mathbf{x}, \mathbf{y})$. The communication complexity of a protocol $\mathcal{P}$ for a function f is defined as the worst case communication complexity over all possible inputs for $(\mathbf{x}, \mathbf{y}) \in \{0, 1\}^n \times \{0, 1\}^n$, which we denote by $CC(\mathcal{P}, f)$:

$$CC(\mathcal{P}, f) = \max_{(\mathbf{x},\mathbf{y}) \in \{0,1\}^n \times \{0,1\}^n} CC(\mathcal{P}, f, \mathbf{x}, \mathbf{y})$$

The communication complexity of a function f is the minimum over all possible protocols:

$$CC(f) = \min_{\mathcal{P}} CC(\mathcal{P}, f)$$

[1] We use agents instead of players to avoid confusion, the communication does not have to be between the players of the game.

Existing Results on Communication Complexity of Nash Equilibria. There are a few results concerning the communication complexity of Nash equilibria. In [4] it is shown that a lower bound on the communication complexity for 2-player games of finding a pure Nash equilibrium is $\Omega(n^2)$, where n is the number of pure strategies for each player. They also show a simple algorithm that finds a pure Nash equilibrium (if it exists) in $O(n^2)$. They do not extend their analysis to mixed Nash equilibria; their method is about finding out whether there exists a pure Nash equilibrium, in contrast with the existence of a mixed Nash equilibrium, which is guaranteed [14].

In [9] the communication complexity of uncoupled equilibrium procedures is studied. They show that for reaching a pure Nash equilibrium, reaching a pure Nash equilibrium in a Bayesian setting and for reaching a mixed Nash equilibrium, a lower bound on the communication complexity is $\Omega(2^s)$, where s is the number of players. To show that reaching this equilibrium is not just due to the complexity of the input, they also show that you can reach a correlated equilibrium in a polynomial number of steps. The methods they use cannot be extended to analysing the communication complexity of ϵ-approximate Nash equilibria. For pure Nash equilibria, their analysis is based on games that might not have a Nash equilibrium and for mixed strategy Nash equilibrium the analysis is based on equilibria that require a large description. Approximate Nash equilibria always exist and can have small descriptions, so the developed techniques do not work for ϵ-approximate Nash equilibria.

1.3 Overview of Our Results

For general $n \times n$ games we show the following bounds on the approximate Nash equilibrium if we fix the amount of communication allowed. We start by considering a version where no communication is allowed. Theorem 1 gives a simple way to find a $\frac{3}{4}$-Nash equilibrium, in this setting. Theorem 3 identifies a contrasting lower bound of slightly more than $\frac{1}{2}$. For one-way communication we exhibit (Theorem 2) a lower bound of $0.5 - o(\frac{1}{\sqrt{n}})$. The DMP-algorithm can be implemented as a algorithm with one-way communication and gives a 0.5-approximate Nash equilibrium. Therefore the constant $\frac{1}{2}$ in the lower bound of Theorem 2 is tight, in this context. In Section 3 we show how to compute a 0.438-Nash equilibrium using polylogarithmic communication.

2 Computing Approximate Nash Equilibria with No Communication

The simplest version of our model is one where there is no communication between the players.[2] That means that for each player $p \in \{r, c\}$, we must find a

[2] This is to some extent inspired by earlier work of the first author [8] that studied an approach to pattern classification in which the set of observations of each class must be processed by an algorithm that proceeds independently of the corresponding algorithms that receive members of the other classes.

function f_p from p's payoff matrix to a mixed strategy, such that for all pairs of matrices (R, C), we have that $(f_r(R), f_c(C))$ is an ϵ-Nash equilibrium.

Theorem 1. *It is possible to guarantee a $\frac{3}{4}$-approximate Nash equilibrium, with no communication between the players.*

Proof. Each player allocates probability $\frac{1}{2}$ to his first pure strategy, and $\frac{1}{2}$ to his best response to the other player's first pure strategy. In detail, let $i \in \arg\max_{i'} R_{i'1}$ and let $j \in \arg\max_{j'} C_{1j'}$. The approximate Nash equilibrium will be $\mathbf{x}^* = \frac{1}{2}\mathbf{e}_1 + \frac{1}{2}\mathbf{e}_i$ and $\mathbf{y}^* = \frac{1}{2}\mathbf{e}_1 + \frac{1}{2}\mathbf{e}_j$.

Let i' be a best pure strategy response of the row player to $\mathbf{y}^*$. Then his incentive to deviate is

$$(\tfrac{1}{2}R_{i'1} + \tfrac{1}{2}R_{i'j}) - (\tfrac{1}{4}R_{11} + \tfrac{1}{4}R_{1j} + \tfrac{1}{4}R_{i1} + \tfrac{1}{4}R_{ij})$$
$$\leq (\tfrac{1}{4}R_{i'1} + \tfrac{1}{2}R_{i'j}) - (\tfrac{1}{4}R_{11} + \tfrac{1}{4}R_{1j} + \tfrac{1}{4}R_{ij}) \leq \tfrac{1}{4}R_{i'1} + \tfrac{1}{2}R_{i'j} \leq \tfrac{1}{4} + \tfrac{1}{2} = \tfrac{3}{4}$$

where the first inequality holds because i was a best response to column 1 (so $R_{i1} \geq R_{i'1}$) and the next inequalities hold because payoffs lie in $[0, 1]$. The same kind of argument holds for the column player. This proves the theorem. □

The following result gives a lower bound of $\frac{1}{2}$; in fact it provides a stronger result saying that $\frac{1}{2}$ is a lower bound for any amount of *one-way communication*, where one player (say, the row player) may send but not receive information about payoffs. Since the DMP-algorithm uses one-way communication, our result shows that it is optimal, in this context.

Theorem 2. *With one-way communication, it is impossibly to guarantee to find an ϵ-Nash equilibrium, for any constant $\epsilon < \frac{1}{2}$.*

Proof. We define a game $G = (R, C)$, where R and C are payoff matrices with dimensions $\binom{n}{k} \times n$, with $k \approx \sqrt{n}$. Consider the following set of column player payoff matrices $C^1, \ldots, C^n$, where C^ℓ has a payoff of 1 for every entry in the ℓth column and a 0 in every other place:

$$\forall i, j : \quad C^\ell_{ij} = 1 \text{ if } j = \ell;\ 0 \text{ otherwise}$$

The row player has matrix R with $\binom{n}{k}$ rows, where a row consists of k 1's and $(n-k)$ 0's. Every row is a different combination, so the $\binom{n}{k}$ rows are all distinct combinations of k 1's in a row of length n.

Let D^r be the strategy of the row player, resulting from matrix R. Let D^c_ℓ be the strategy of the column player resulting from matrices R and C^ℓ; note that with unlimited one-way communication we can assume that the row player sends all of R to the column player.

We will show that for this class of games, one cannot do better than a $(\frac{1}{2} - o(\frac{1}{\sqrt{n}}))$-approximate Nash equilibrium. This implies for large values of n approximately a $\frac{1}{2}$-approximate Nash equilibrium.

During the proof we will search for a lower bound of $\frac{1}{2} - z$, where the value of z is to be determined.

First observe that a best response for the column player having matrix C^ℓ is $\mathbf{e}_\ell$, the pure strategy of column ℓ. It has payoff 1 and other columns have payoff 0. So to reach a $(\frac{1}{2} - z)$-approximate Nash equilibrium, D^c_ℓ must allocate a probability at least $(\frac{1}{2} + z)$ to column ℓ.

The row player has one matrix R with all different combinations of k 1's in a row of length n. Now consider the columns of R. By construction each column of R consists of $\frac{k}{n} \cdot \binom{n}{k}$ 1's and $(1 - \frac{k}{n}) \cdot \binom{n}{k}$ 0's.

D^r assigns a probability to each row of R. Define an unnormalised probability distribution Φ over the columns as follows. Φ assigns to each column j a value $\Phi(j)$, which gives the probability that a 1 will be in this column given a row sampled from D^r. This value $\Phi(j)$ will be at most 1, when every row that is played with positive probability has a 1 in column j. Because every row contains k 1's, the sum of over all values will sum to k: $\sum_{j=1}^{n} \Phi(j) = k$.

We define column m to be one with a lowest value of Φ: $m \in \operatorname{argmin}_j \Phi(j)$. Suppose the column player has payoff matrix C^m. Note that the sum over all values $\Phi(j)$ is k and there are n columns. This means that $\Phi(m)$ is at most $\frac{k}{n}$. This means that column m, which is played at least $\frac{1}{2} + z$ of the time by the column player, gives a payoff of 0 with a probability of at least $1 - \frac{k}{n}$.

We now consider the row player's strategy D^r and construct an improved response D^* —that is supposed to be an improvement of at most $\frac{1}{2} - z$— as follows. D^* will differ from D^r in the following way. For every row i we see if there is a 1 on the mth entry. If this is the case, we do not change anything. If there is a 0 on the mth entry we do the following: look at the positions where there is a 1 in row i. Of all the entries where there is a 1, we select the entry to which the column player gives the lowest probability, say entry a. Now we move all the probability allocated by D^r this row, to the row of R that instead has a 0 on entry a and a 1 on entry m, and is otherwise the same as i.

The probability on entry a is defined as the smallest of all the entries where this row has a 1. We can bound the probability that was given to this entry by the column player. A probability at least $\frac{1}{2} + z$ is given to column m, so a probability of $\frac{1}{2} - z$ can be distributed over the remaining columns. The column belonging to entry a has the smallest probability of at least k columns, so the probability given to column a is at most $\frac{1/2 - z}{k}$.

The result of this construction of D^* from D^r is that every row that is played with positive probability by D^* will have a 1 on the mth entry. There is a probability at least $(1 - \frac{k}{n})$ that a row sampled from D^r did not have a 1 on the mth entry. This means that the increase in payoff from replacing D^r with D^* is at least

$$\left(1 - \frac{k}{n}\right) \cdot \left(\frac{1}{2} + z\right) - \left(1 - \frac{k}{n}\right) \cdot \frac{1/2 - z}{k} = \left(1 - \frac{k}{n}\right) \cdot \left(\frac{1}{2} + z - \frac{\frac{1}{2} - z}{k}\right)$$

We will show that this increase in payoff is close to $\frac{1}{2}$ for well chosen k and z. Assume that z is chosen such that $z = \frac{(1/2) - z}{k}$. Equivalently, $z = 1/(2k + 2)$.

This will make the difference in payoff between D^r and D^* at least

$$\left(1-\frac{k}{n}\right)\cdot\left(\frac{1}{2}+z-z\right)=\frac{1}{2}-\frac{k}{2n}.$$

So if the column player has a regret (as defined in Section 1.1) of $\leq \frac{1}{2}-z$, the row player has a regret of at least $\frac{1}{2}-\frac{k}{2n}$, and we put $z=\frac{1}{2k+2}$. We can use these two observations to find the value of k such that the regrets are the same for the row player and column player:

$$\begin{aligned}\tfrac{1}{2}-\tfrac{k}{2n} &= \tfrac{1}{2}-\tfrac{1}{2(k+1)}\\ \tfrac{k}{2n} &= \tfrac{1}{2(k+1)}\\ k=\tfrac{1}{2}(\sqrt{4n+1}-1) &\vee k=\tfrac{1}{2}(-\sqrt{4n+1}-1)\end{aligned}$$

Since k should be greater than 0, only the first solution is feasible. So we have $k=\frac{1}{2}(\sqrt{4n+1}-1)$ and $z=\frac{\frac{1}{2}(\sqrt{4n+1}-1)}{2n}$, which is $o(\frac{1}{\sqrt{n}})$. We have proven now that for general games with one-way communication one cannot do better than a $(\frac{1}{2}-o(\frac{1}{\sqrt{n}}))$-approximate Nash equilibrium. □

Theorem 3. *It is impossible to guarantee a* 0.501*-Nash equilibrium, with no communication between the players.*

As we noted, the previous Theorem 2 already shows a lower bound of $\frac{1}{2}$ in this setting. Theorem 3 rules out the possibility that $\frac{1}{2}$ is the correct answer, as it was for one-way communication.

Proof. (sketch) For $p \in \{r, c\}$, let Ω^p be the set of (mixed) strategies p may use (the image of f_p). Let $\mathbf{c}^p$ be a distribution over $[n]$ that minimises the maximum variation distance $d_{\max}$ from $\mathbf{c}^p$ to elements of Ω^p; $\mathbf{c}^p$ is called the *centre strategy* for p, and p's *commitment* (denoted τ^p) is $1-d_{\max}$. Thus $\tau^p \in [0,1]$ and is high when p must choose a strategy close to some $\mathbf{c}^p$.

The proof is by case analysis on the values τ^r and τ^c. If either value (say τ^c) is ≥ 0.501, then c's matrix C is chosen to be C^ℓ as in the proof of Theorem 2 where column ℓ receives low probability from $\mathbf{c}^c$. c's high commitment prevents c from deviating sufficiently far from $\mathbf{c}^c$ to make a good enough response.

If either value (say τ^c) is ≤ 0.05 then c has 3 strategies s_1, s_2, s_3 that are all very far apart in variation distance. Design a matrix for r where row i is a very good response to s_i but a poor response to $s_j \neq s_i$. The row player has no strategy that is sure to fall short of optimal by ≤ 0.501.

If $\tau^r, \tau^c \in [0.05, 0.501]$, assume $\tau^r \geq \tau^c$, and design a matrix R such that r's commitment forces him to allocate nearly 0.05 of his probability to rows that have zero payoff. The remaining rows $S \subset [n]$ have payoff 1 against "most" columns (w.r.t. measure $\mathbf{c}^c$). Each row in S is a good response to one of the remaining columns, associated with that row alone, but gets payoff 0 against others. The column player can be forced by matrix C to allocate probability ≥ 0.499 to one of those columns. r loses 0.05 due to having to allocate ≥ 0.05 to rows outside S, and a further ~ 0.49 due to not knowing which row in S is the best one to use, for a total regret > 0.501. □

3 A 0.438-Approximate Nash Equilibrium with Limited Communication

This section provides a 0.438-approximate Nash equilibrium where the amount of communication between the players is polylogarithmic in n. We present the algorithm as an α-approximate Nash equilibrium first and then optimize α. At various points the algorithm uses the operation of communicating a mixed strategy (a probability distribution over $[n]$) from one player to the other; the details of this operation are given in Section 3.1. The general idea is to send a sample of size $O(\log n)$ from the distribution and argue that the corresponding empirical distribution is a good enough estimate for our purposes.

First the row player finds a Nash equilibrium for the zero-sum game $(R, -R)$ and the column player computes a Nash equilibrium for the zero-sum game $(-C, C)$. Since both games are zero-sum, we know that the payoff values for their Nash equilibria will be unique. Both players compare this payoff value with α. We distinguish two cases, the Nash equilibrium of both players is lower than α (Case 1) or at least one of the players has a value equal to or higher than α for his Nash equilibrium (Case 2). With $O(1)$ communication, the case that holds can be identified.

Case 1:
Both players have a Nash equilibrium with value smaller than α. The row player finds a strategy pair $(\mathbf{x}_r^*, \mathbf{y}_r^*)$ and the column player a strategy pair $(\mathbf{x}_c^*, \mathbf{y}_c^*)$. The row player communicates $\mathbf{y}_r^*$ to the column player (as described in Section 3.1) and the column player sends $\mathbf{x}_c^*$ to the row player. They now play the game with the strategy pair $(\mathbf{x}_c^*, \mathbf{y}_r^*)$. Since $\mathbf{y}_r^*$ was a Nash equilibrium strategy in the zero-sum game $(R, -R)$ and the row player still plays with payoff matrix R, by definition of a Nash equilibrium, the row player has no strategy that can give him a payoff of α or higher. The row player has a best response with a value of at most α, so his regret is also at most α. This leads to an α-approximate Nash equilibrium for the row player. The strategy $\mathbf{x}_c^*$ was a Nash equilibrium strategy in the zero-sum game $(-C, C)$ and the column player still has payoff matrix C. So we can use the same argument for the column player to argue that when the row player has strategy $\mathbf{x}_c^*$, the column player has a α-approximate Nash equilibrium. This concludes Case 1.

Case 2:
If at least one of the players has a value of at least α for his zero-sum game, he can get a payoff of at least α if he plays this strategy, regardless the strategy of the other player. Assume w.l.o.g. that it is the row player who has a payoff of at least α in his zero-sum game. He communicates this strategy $\mathbf{x}_r^*$ to the column player (again, as described in Section 3.1). The column player identifies a pure strategy best response $\mathbf{e}_j$ to the strategy of the row player and communicates this strategy to the row player (using $\log n$ bits).

At this point in the algorithm we have the strategy pair $(\mathbf{x}_r^*, \mathbf{e}_j)$. The column player has a best response strategy, so at this point his strategy is a 0-approximate Nash equilibrium. The row player can guarantee a payoff of α.

Let $\beta \leq 1$ be the value of his best response to $\mathbf{e}_j$. So at this point the row player has a $\beta - \alpha$-approximate Nash equilibrium. We next deal with the possibility that $\beta - \alpha > \alpha$.

At this stage the column player has a 0-approximate Nash equilibrium while we are only looking for a α-approximate Nash equilibrium; meanwhile the row player has a strategy that might not be good enough for a α-approximate Nash equilibrium. To change this, we use a method used in [3] (Lemma 3.2), which allows the row player to shift some of his probability to his best response to $\mathbf{e}_j$. By shifting some of his probability, it could be that $\mathbf{e}_j$ no longer is a best response strategy for the column player. This is allowed, as long as the column player's regret while playing $\mathbf{e}_j$ is at most α. Suppose the row player shifts $\frac{1}{2}\alpha$ of his probability to a best response strategy. The payoff the column player gets could be $\frac{1}{2}\alpha$ lower because of this move. The payoff of some other strategy could go as much as $\frac{1}{2}\alpha$ higher because of this shift. The strategy $\mathbf{e}_j$ was a 0-approximate Nash equilibrium, so by the shift of $\frac{1}{2}\alpha$ of the row player's probability, the regret of the column player is at most $\frac{1}{2}\alpha + \frac{1}{2}\alpha = \alpha$, which constitutes an α-approximate Nash equilibrium, for the column player.

The row player is allowed to change the allocation of $\frac{1}{2}\alpha$ of his probability with the worst payoff. Since we rearrange the worst part of the row player, the remainder of his probability, $1 - \frac{1}{2}\alpha$ had already at least a payoff of α. The probability is shifted to his best response with a value of β, with $\alpha \leq \beta \leq 1$. This leads to the following inequality:

$$(1 - \frac{1}{2}\alpha)\alpha + \frac{1}{2}\alpha\beta \geq \beta - \alpha \ , \quad 0 \leq \alpha \leq \beta \leq 1$$

The solutions to this inequality are

$$\begin{array}{rl} 0 < \alpha \leq \frac{1}{2}(5 - \sqrt{17}) & \alpha \leq \beta \leq \frac{\alpha^2 - 4\alpha}{\alpha - 2} \\ \frac{1}{2}(5 - \sqrt{17}) < \alpha < 1 & \alpha \leq \beta \leq 1 \\ \alpha = 0 \quad \beta = 0 & \alpha = 1 \quad \beta = 1 \end{array}$$

where it holds that if $\alpha = \frac{1}{2}(5 - \sqrt{17})$ then $f(\alpha) = \frac{\alpha^2 - 4\alpha}{\alpha - 2} = 1$ and for $0 \leq \alpha \leq 1$ this function is monotone increasing. This procedure will give an α-approximate Nash equilibrium, so α should be as low as possible. Next to this it should also hold for every β with $\alpha \leq \beta \leq 1$. The lowest α such that this condition hold is when $f(\alpha) = 1$, thus $\alpha = \frac{1}{2}(5 - \sqrt{17}) \approx 0.438$.

So if the row player rearranges $\frac{1}{2} \cdot 0.438 = 0.219$ of his probability to his best response row, both players have a strategy that guarantees them a 0.438-approximate Nash equilibrium.

3.1 Communicating Mixed Strategies

We describe how to communicate an approximation of the mixed strategies that are computed, using $O(\log^2 n)$ bits. We ultimately obtain an ϵ of $0.438 + \delta$, for any $\delta > 0$.

We first look at the case where one of the players, assume w.l.o.g. the row player, has a payoff higher than α in the Nash equilibrium of his zero-sum game

$(R, -R)$. The column player plays a pure best response to the strategy of the row player, regardless of the support of the strategy of the row player. So we mainly consider the row player.

The zero-sum game $(R, -R)$ gives a strategy pair $(\mathbf{x}^*, \mathbf{y}^*)$. Fix $k = \frac{\ln n}{\delta^2}$ and form a multiset A by sampling k times from the set of pure strategies of the row player, independently at random according to the distribution $\mathbf{x}^*$. Let $\mathbf{x}'$ be the mixed strategy for the row player with a probability of $\frac{1}{k}$ for every member of A. We want the distribution $\mathbf{x}'$ to have a payoff close to the payoff of $\mathbf{x}^*$. This corresponds to the following event:

$$\phi = \{((\mathbf{x}')^T R\mathbf{y}^*) - ((\mathbf{x}^*)^T R\mathbf{y}^*) < -\delta\}$$

As noted in [13] the expression $((\mathbf{x}')^T R\mathbf{y}^*)$ is essentially a sum of k independent random variables each of expected value $((\mathbf{x}^*)^T R\mathbf{y}^*)$, where every random variable has a value between 0 and 1. This means we can bound the probability that ϕ does not hold, which we will call ϕ^c. When we apply a standard tail inequality [10] to bound the probability of ϕ^c, we get:

$$\Pr[\phi^c] \leq e^{-2k\delta^2}$$

With $k = \frac{\ln n}{\delta^2}$, this gives $\Pr[\phi^c] \leq \frac{1}{n^2}$ and $\Pr[\phi] \geq 1 - \frac{1}{n^2}$. If $\mathbf{x}'$ does not give payoffs close enough to $\mathbf{x}^*$, we sample again.

The strategy $\mathbf{x}'$ has a guaranteed payoff of $0.438 + \delta - \delta = 0.438$. This strategy is communicated to the column player. The support of this strategy is logarithmic and all probabilities are rational (multiples of $\frac{1}{k}$). Communication of one pure strategy has a communication complexity of $O(\log n)$. This will give a communication complexity for $\mathbf{x}'$ of $O(\log^2 n)$.

The column player computes a pure strategy best response to $\mathbf{x}'$ and communicates this strategy in $O(\log n)$ to the row player. The strategy of the row player might not yet lead to a 0.438-approximate Nash equilibrium, his payoff could be too low. As we have seen before, if the row player redistributes at most 0.219 of his probability, he is guaranteed to have a strategy that leads to a 0.438-approximate Nash equilibrium.

This change in strategy of the row player can decrease the payoff of the column player by as much as 0.219 and increase another pure strategy by as much as 0.219. His strategy was a best response, a 0-approximate Nash equilibrium, and the improvement to another pure strategy is maximal $0.219 + 0.219 = 0.438$, this leads to a 0.438-approximate Nash equilibrium.

In the alternative case, where both players have a low ($< \alpha$) payoff in their zero-sum games, the technique is essentially the same: each player samples k times from the opposing distribution, checks that it limits his own payoff to at most $\alpha + \delta$, re-samples as necessary, and communicates the k-sample.

4 Conclusions

The general topic of the communication complexity of approximate Nash equilibrium, seems to be a rich source of research questions. [16] considers some related ones, including the communication required for approximate *well-supported* equilibria, as well as games of fixed size. It may be that future work should address the issue of communication protocols where the players have an incentive to report their information truthfully.

Acknowledgements. The first author thanks Sergiu Hart for useful discussions during the iAGT workshop in May 2011.

References

1. Bosse, H., Byrka, J., Markakis, E.: New Algorithms for Approximate Nash Equilibria in Bimatrix Games. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 17–29. Springer, Heidelberg (2007)
2. Chen, X., Deng, X.: Settling the complexity of two-player Nash equilibrium. In: Procs. of the 47th FOCS Symposium, pp. 261–272. IEEE (2006)
3. Chen, X., Deng, X., Teng, S.H.: Settling the complexity of computing two-player Nash equilibria. J. ACM 56, 14:1–14:57 (2009)
4. Conitzer, V., Sandholm, T.: Communication complexity as a lower bound for learning in games. In: Proceedings of the 21st ICML, pp. 24–32 (2004)
5. Dantzig, G.B.: Linear Programming and Extensions. Princeton Univ. Press (1963)
6. Daskalakis, C., Goldberg, P.W., Papadimitriou, C.H.: The complexity of computing a Nash equilibrium. SIAM J. Comput. 39(1), 195–259 (2009)
7. Daskalakis, C., Mehta, A., Papadimitriou, C.: A Note on Approximate Nash Equilibria. In: Spirakis, P.G., Mavronicolas, M., Kontogiannis, S.C. (eds.) WINE 2006. LNCS, vol. 4286, pp. 297–306. Springer, Heidelberg (2006)
8. Goldberg, P.W.: Some discriminant-based PAC algorithms. Journal of Machine Learning Research 7, 283–306 (2006)
9. Hart, S., Mansour, Y.: How long to equilibrium? the communication complexity of uncoupled equilibrium procedures. GEB 69(1), 107–126 (2010)
10. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association 58(301), 13–30 (1963)
11. Karmarkar, N.: A new polynomial-time algorithm for linear programming. In: 16th STOC, pp. 302–311. ACM (1984)
12. Kushilevitz, E.: Communication complexity. Advances in Computers 44, 331–360 (1997)
13. Lipton, R.J., Markakis, E., Mehta, A.: Playing large games using simple strategies. In: Procs. of the 4th ACM-EC, EC 2003, pp. 36–41 (2003)
14. Nash, J.: Non-cooperative games. Ann. Math. 54(2), 286–295 (1951)
15. von Neumann, J.: Zur theorie der gesellschaftsspiele. Mathematische Annalen 100, 295–320 (1928)
16. Pastink, A.: Aspects of communication complexity for approximating Nash equilibria. MSc dissertation, Utrecht University (2012)
17. Tsaknakis, H., Spirakis, P.G.: An Optimization Approach for Approximate Nash Equilibria. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 42–56. Springer, Heidelberg (2007)
18. Yao, A.C.C.: Some complexity questions related to distributive computing (preliminary report). In: 11th STOC, pp. 209–213. ACM (1979)