Applications of Communication Complexity for Game Theory

Lecturers: Jaiganesh Balasundaram and Tyrone Strangway

This document is divided into two sections. The first section deals with the communication complexity of reaching Nash equilibrium. The second section focuses on the applications of communication complexity to algorithmic mechanism design.

1 Nash Equilibrium

1.1 Introduction

1.1.1 An example of a 2 player game

Imagine the following scenario. Two prisoners, prisoner A and prisoner B, who are in the prison for the same crime, are being interrogated simultaneously in separate rooms. They are each given 2 options: (a) Either cooperate with the other prisoner by staying silent or (b) Betray the other prisoner by snitching him out. They are each given deals (reduced prison sentences) for betraying the other prisoner. However, if it turns out that both of them had betrayed each other, then they will both serve longer sentences than if they had both stayed silent. So, there are four possible scenarios: (a) Both stay silent (b) Both betray each other (c) Prisoner A remains silent and prisoner B betrays and (d) Prisoner B remains silent and prisoner A betrays. Below is a matrix representing the "payoffs" for each prisoner for the four different scenarios (By "payoffs", we mean the quantification of preferences for different scenarios, depending on the prison sentences they each get for each scenario).

	Cooperate	**Betray**
Cooperate	**3,3**	**0,5**
Betray	**5,0**	**2,2**

The first entries of the matrix represent the payoffs for prisoner A, and the second entries of the matrix represent the payoffs for prisoner B. Lower numbers should be interpreted as higher prison sentences and higher numbers as reduced prison sentences. Now, both prisoners are better off by both staying silent, than by both betraying each other (because in the first case both get a payoff of 3, whereas in the latter case both get a payoff of 2). However, the scenario where both prisoners betray each other is called the *equilibrium* point for this 2 person game because of the following reason. In this scenario, both of them have no incentive to deviate from their actions because, for each prisoner, deviating from his action while the other person doesn't (that is, choosing to cooperate while the other person still betrays) puts him in a worse position. It is easy to see that the same cannot be said for the remaining three scenarios. This is actually a quite famous example called the "Prisoners' Dilemma". The two prisoners are the *players* of this game, where each player has 2 possible *actions* or *strategies* forming their action set: {cooperate, betray}. Each of the 4 possible scenario is termed as a *joint action.* The entries of the matrix form the *payoff or utility* *function* for each player, which as we mentioned earlier, is nothing but the quantification of their preferences for the different possible scenarios. Finally, the joint action (betray,betray), which we described above as the *equilibrium* point of this game, is formally termed the "Nash equilibrium" of the game. We will be formalizing these notions in the next section.

1.1.2 The Problem

In this paper, we will be discussing the problem of complexity of *reaching* equilibrium. More specifically, we will be looking at the communication model (which we will be formally defining in the next section), in which players initially only know their own payoff functions [1], and analyse how much information must be transferred between them to jointly compute the equilibrium point. We will also assume that the players follow a predetermined protocol, that is, we abstract away the incentives of each player and are only concerned about the amount of information that needs to be transferred between them to attain equilibrium. The motivation for this model has been well studied [3].

We will be discussing results from two papers. First, we will be discussing results from the paper "How long to equilibrium? The communication complexity of uncoupled equilibrium procedures" by Hart and Mansour [8], where will be looking at the communication complexity lower bounds for reaching *pure* Nash equilibrium. Then, we will be discussing results from the paper "On the communication complexity of approximate Nash equilibria" by Goldberg and Pastink [6], where we will be discussing the communication complexity results for *mixed* Nash equilibrium.

Now, we present some of the results regarding reaching pure Nash equilibrium. We will be using the notations and definitions of [8].

1.2 The Setting for Pure Action Games

There are $n \geq 2$ players. Each player i, $1 \leq i \leq n$, has a finite set of actions (or strategies) A_i, with $|A_i| \geq 2$. For the analysis of pure action games, we will be only considering binary action games, that is, for each i, $A_i = \{0, 1\}$. Let $A = \prod_{i=1}^{n} A_i$ denote the joint action space (for binary action games, $A = \{0, 1\}^n$, the set of all $n-$ bit strings). Each player has a private payoff (or utility) function $u_i : A \to \{0, 1\}^n$ (For simplicity we are assuming that the utilities are finitely represented, that is, are rational numbers with $n-$ bit numerator and denominator). Let's denote the game by $G = (n, \{A_i\}, \{u_i\})$.

For a joint action $a = (a_1, ..., a_n) \in A$ (for binary action games, a joint action is an $n-$ bit binary string), let a^{-i} denote the joint action of all players except player i. For each player i (with utility function u_i), the *best response* to a joint action a^{-i}, denoted by $BR(a^{-i}, u_i)$ is $arg\,max_{a_i \in A_i} u_i(a_i, a^{-i})$. A joint action $a \in A$ is a **Pure Nash Equilibrium** if $u_i(a) \geq u_i(b_i, a^{-i})$ for *every* player i and any action $b_i \in A_i$ (equivalently, $a_i \in BR(a^{-i}, u_i)$ for every i).

[1] This is usually called the "uncoupled" setting.

1.3 The Communication Model: Pure Nash Equilibrium Procedures

A Pure Nash Equilibrium Procedure is a dynamic process by which the players reach a Pure Nash Equilibrium of the game (if one exists). Recall that a game is denoted by $G = (n, \{A_i\}, \{u_i\})$. Fix n and the action spaces, so a game is identified by its payoff function set $(u_1, ..., u_n)$. Let $\mathcal{G}$ be a family of games to which the procedure should apply. The input to the procedure is a game $G = (u_1, ..., u_n)$ in the family $\mathcal{G}$. Initially, each player i has access only to his own payoff function u_i. In each round $t = 1, 2, ...$, every player outputs a bit (say writes the bit on a blackboard). This bit is a function of the the player's payoff function and what he has seen so far in the procedure. At the end of round t, all the players observe the joint output.

In a Pure Nash Equilibrium Procedure Π for a game G, the "output" of player i is either

- a pure action $a_i \in A_i$ (which is a bit) or
- a declaration of "no pure nash equilibrium"

In the first case, the joint output $(a_1, ..., a_n) \in A$ is a pure Nash equilibrium of G. In the second case, G has no pure Nash equilibrium.

The communication complexity of a Pure Nash equilibrium procedure Π applied to a game G, denoted by $CC(\Pi, G)$, is the number of bits communicated until Π terminates when the input is G. Given a family of games $\mathcal{G}$, the communication complexity of a Pure Nash equilibrium procedure Π for a family $\mathcal{G}$, denoted by $CC(\Pi, \mathcal{G})$, is the worst case communication complexity of Π over all games $G \in \mathcal{G}$, that is, $CC(\Pi, \mathcal{G}) = max_{G \in \mathcal{G}} CC(\Pi, G)$. The communication complexity of pure Nash equilibrium procedures for a family of games $\mathcal{G}$, denoted by $CC(\mathcal{G})$, is the minimal communication complexity of any Pure Nash equilibrium procedure Π for the family $\mathcal{G}$, that is, $CC(\mathcal{G}) = min_{\Pi} CC(\Pi, \mathcal{G})$. Finally, when the games in $\mathcal{G}$ are chosen according to a probability distribution $\mathbf{P}$, then the expected communication complexity of Pure Nash Equilibrium Procedures for the family $\mathcal{G}$ is $min_{\Pi} \mathbf{E}[CC(\Pi, \mathcal{G})]$, where the expectation is taken with respect to $\mathbf{P}$.

1.4 Communication Complexity of Pure Nash Equilibrium Procedures

We are interested in the asymptotic behaviour of the communication complexity of Nash equilibrium procedures as the number of player n increases, while the size of the action sets is fixed. Specifically, let Γ_2^n be the family of all $n-$ person binary action games. We prove the following theorem.

Theorem 1 *Any Nash equilibrium procedure for the family Γ_2^n has communication complexity $\Omega(2^n)$. That is,*

$$CC(\Gamma_2^n) = \Omega(2^n)$$

Note that since the communication complexity is defined as the worst case over all games, $\Omega(2^n)$ is also a lower bound for the communication complexity of any Nash equilibrium procedure for the family Γ_s^n for every $s \geq 2$.

Proof We give a simple reduction from set disjointness. Say n is even and divide the player set $\{1, 2, ..., n\}$ into two sets T_1 and T_2, each of size $n/2$, where $T_1 = \{1, ..., n/2\}$ and $T_2 =$

$\{n/2+1,...,n\}$. Also for notational convenience, each player in set T_l is referred to as player (l,i) where $l \in \{1,2\}$ and $i \in \{1,2,...,n/2\}$ [2]. Using similar notations, let us denote a $n-$ bit joint action a as $(a_{11},...,a_{1n/2},a_{21},...,a_{2n/2})$, where each a_{li}, $l \in \{1,2\}$ and $i \in \{1,2,...,n/2\}$, is a bit.

Let $S = \{0,1\}^n$, the set of all $n-$ bit strings [3]. For any two sets $S_1, S_2 \subseteq S$, the input to the set disjointness problem, we define a binary action game G in Γ_2^n, such that $S_1 \cap S_2 \neq \emptyset$ iff G has a pure Nash equilibrium, and for every $a \in A$, the value of the payoff function of player (l,i) on a, that is, $u_{l,i}(a)$, is computable, by a finite algorithm, from a, S_l, i. So, we will have that if the players reach a pure Nash equilibrium in G, then the sets S_1 and S_2 are not disjoint, and if they do not reach a pure Nash equilibrium then the sets are disjoint. Given a pure Nash equilibrium procedure Π_{NE}, we will be able to generate a protocol Π_D for the set disjointness problem, with the same communication complexity [4].

Now we define the payoff function for each player (l,i) where $l \in \{1,2\}$ and $i \in \{1,2,...,n/2\}$. The idea is that, for each player (l,i) and each joint action $a \in A$, the the payoff $u_{l,i}(a)$ will be "high" if a lies in the set S_l and "low" if it doesn't. Also, the payoff functions for players $i \geq 3$ in both sets T_1, T_2 would be identical, and for players $i = 1,2$ in both sets T_1, T_2, the payoff functions would be slightly different than the rest of the players. The reason we are singling out these 4 players $(1,1),(1,2),(2,1)$ and $(2,2)$ is because, we would like to show that if the sets are disjoint, then there exists a player (among these four) who would benefit by deviating on any joint action set (and hence we would show that there is no pure Nash equilibrium for the game).

Formally, for $l = 1,2$ and $i \geq 3$, the payoff functions are

$$u_{l,i}(a) = \begin{cases} 2, & \text{if } a \in S_l \\ 0, & \text{if } a \notin S_l \end{cases}$$

That is, for a player (l,i) where $l = 1,2$ and $i \geq 3$, the value of $u_{l,i}(a)$ is just a function of whether a lies in S_l or not.

We give the payoff functions for players $(1,1)$ and $(1,2)$. The payoff functions for players $(2,1)$ and $(2,2)$ are similar [5] (Recall that the $n-$ bit joint action is represented as $a = (a_{11},...,a_{1n/2},a_{21},...,a_{2n/2})$).

$$u_{1,1}(a) = \begin{cases} 2, & \text{if } a \in S_1 \\ 1, & \text{if } a \notin S_1 \text{ and } a_{11} = a_{12} \\ 0, & \text{if } a \notin S_1 \text{ and } a_{11} \neq a_{12} \end{cases}$$

$$u_{1,2}(a) = \begin{cases} 2, & \text{if } a \in S_1 \\ 1, & \text{if } a \notin S_1 \text{ and } a_{11} \neq a_{12} \\ 0, & \text{if } a \notin S_1 \text{ and } a_{11} = a_{12} \end{cases}$$

[2] So for example, the first player in T_1 is referred to as player $(1,1)$, and last player in T_2 is referred to as player $(2,n/2)$

[3] Equivalently a 2^n bit string with all 1s. So any subset of S is also a set of strings (or equivalently a 2^n bit string with 1s representing the strings in the subset).

[4] More specifically, we will have Alice simulate all players in T_1 and Bob simulate all players in T_2.

[5] Just replace S_1 by S_2, a_{11} by a_{21}, and a_{12} by a_{22}.

For player $(1, 1)$, the value $u_{1,1}(a)$ is not just a function of whether a lies in S_1, but is also a function of what the first two bits of a are (Similar thing holds for the other three players). Now, we prove that S_1 and S_2 are disjoint if and only if the game $G = \{u_{l,i}\}_{l,i}$ has no pure Nash equilibrium.

Say S_1 and S_2 are not disjoint. Let $a \in S_1 \cap S_2$. We claim that a is indeed a pure Nash equilibrium for the game G. This is because, since $a \in S_1 \cap S_2$, by definition, the payoff for each player on a is 2, which is the maximal payoff. So no player has any reason to deviate from the joint action a.

Say S_1 and S_2 are disjoint. For every $a \in A$, if $a \notin S_l$ for some $l \in \{1, 2\}$, then either $(l, 1)$ or $(l, 2)$ benefits by deviating. To see this, consider the case that $a \in S_2$, but $a \notin S_1$. Now since $a \in S_2$, by definition, the payoff of all the players in the set T_2 on the joint action a is 2. Also, since $a \notin S_1$, by definition, the payoff of every player in the set T_1, except players $(1, 1)$ and $(1, 2)$, is zero on the joint action a. Now the payoff of players $(1, 1)$ and $(1, 2)$ depends on the the first two bits of a, namely a_{11}, a_{12} [6]. There are two cases. If $a_{11} = a_{12}$, the the payoff of player $(1, 1)$ is 1 and the payoff of player $(1, 2)$ is zero. In this case, a is not a pure Nash equilibrium because player $(1, 2)$ benefits by negating his action, since for this modified joint action say a' (which is a, but with the second bit negated), the player $(1, 2)$ gets a payoff of at least [7] 1 (since $a'_{11} \neq a'_{12}$). A similar argument can be made for the case $a_{11} \neq a_{12}$. So, we have proved that if S_1 and S_2 are disjoint, then no pure Nash equilibrium exists for the game G. This completes the proof of the theorem.

□

1.5 Average Case Communication Complexity of Pure Nash Equilibrium Procedures

While the communication complexity of Pure Nash Equilibrium procedures is exponential (in the number of players) in the worst case, it is possible that the *expected* communication complexity (when the games are chosen according to a probability distribution) will be smaller. However, we will show a simple probability distribution over Γ_2^n, such that the expected communication complexity of pure Nash equilibrium procedures (where the expectation is over the choice of games in Γ_2^n according to the probability distribution) is also exponential in the number of players. More specifically, we will prove the following theorem.

Theorem 2 *There exists a probability distribution* $\boldsymbol{P}$ *over* Γ_2^n *such that*

$$min_{\Pi}\boldsymbol{E}(CC(\Pi, \Gamma_2^n)) = \Omega(2^n)$$

Note that Theorem 2 implies Theorem 1. We will provide a direct proof here, rather than applying a reduction. Before proceeding with the proof, we will define what we mean by a *combinatorial rectangle* in this setting.

Definition [Combinatorial Rectangle] A *combinatorial rectangle* is a set $U = U_1 \times ... \times U_n$ where each U_i is a set of payoff functions of player i.

[6] For this case, we can think of players $(1, 1)$ and $(1, 2)$ as playing the "matching pennies game", which is the binary action equivalent of the popular "Rocks, Papers and Scissors" game, where the two pennies are represented by the bits a_{11} and a_{12}. Indeed, [8] term this reduction as the "matching pennies reduction".

[7] We say "at least" because a' might now be in S_1 and not S_2, in which case, the payoff will be 2.

Let V_i represent the set of all possible payoff functions of player i. Then, we have that a combinatorial rectangle is a set $U \subseteq V_1 \times ... \times V_n$ [8], such that $U = U_1 \times ... \times U_n$, where for each i, $U_i \subseteq V_i$.

Definition [Monochromatic Combinatorial Rectangle] A *monochromatic* rectangle is labelled by either

- a pure joint action $a \in A$, where a is the pure Nash Equilibrium for every game $(u_1, ..., u_n) \in U$ or
- "No pure Nash equilibrium" (when no game $(u_1, ..., u_n) \in U$ has a pure Nash equilibrium)

Now, we proceed to prove Theorem 2.

Proof The outline of the proof is as follows. The idea is that it will be "hard" for the players to agree that there is no pure Nash equilibrium. Specifically, we will construct a probability distribution over Γ_2^n such that

- The probability that there is no pure Nash equilibrium is bounded *away* from 0 as n increases.
- Any combinatorial rectangle that is labelled "No pure Nash equilibrium" has a low probability.

Now, we give the simple probability distribution $\mathbf{P}$, over Γ_2^n. The payoff function u_i of player i is selected randomly as follows. For each $a^{-i} \in \{0,1\}^{n-1}$ (recall that a^{-i} denotes the joint action of all players except player i), we toss a fair coin. Let c be the outcome of the coin toss. We let $u_i(0, a^{-i}) = c$ and $u_i(1, a^{-i}) = 1 - c$. The coin tosses are independent over all a^{-i} and over all i. More formally, for every $a^{-i} \in \{0,1\}^{n-1}$, with probability 1/2 we let
$u_i(0, a^{-i}) = 0$ and
$u_i(1, a^{-i}) = 1$,
and with probability 1/2 we let
$u_i(0, a^{-i}) = 1$ and
$u_i(1, a^{-i}) = 0$,
These choices are made independently over all a^{-i} and over all i. It is easy to that, for all $a \in \{0,1\}^n$, for all i, we have that

$$Pr[u_i \ : \ a_i \in BR(a^{-i}, u_i)] = Pr[u_i \ : \ a_i \notin BR(a^{-i}, u_i)] = 1/2$$

where a_i is the action of player i (which is a bit). Also, recall that for each player i with utility u_i, $BR(a^{-i}, u_i)$ is the best response to a joint action a^{-i}, and is $arg\,max_{a_i \in A_i} u_i(a_i, a^{-i})$.

Suppose we sample from the set Γ_2^n according to the distribution $\mathbf{P}$. Let $(u_1, ..., u_n)$ represent the corresponding random variable.

Now we proceed to prove that the probability that there is no pure Nash equilibrium for $(u_1, ..., u_n)$ is bounded *away* from 0 as n increases. Formally,

Lemma 3 *There exists a constant $\alpha > 0$ such that*

$$Pr[(u_1, ..., u_n) \textit{ has no pure Nash equilibira}] \geq \alpha$$

for all $n \geq 2$.

[8]This is actually Γ_2^n.

Proof of Lemma 3 Let N be the random variable representing the number of pure Nash equilibria for $(u_1, ..., u_n)$. We have to prove that the probability $Pr[N = 0]$ is bounded away from zero. To this end, we calculate the expectation of the random variable N. First, it is easy to see that, for every joint action $a \in \{0, 1\}^n$, the probability that a is a pure Nash equilibrium for G is $1/2^n$. This is because, for a joint action a to be a pure Nash equilibrium, we should have that for each i, the payoff function u_i is such that $a_i \in BR(a^{-i}, u_i)$. As we saw above, this happens with probability $1/2$, independently, for each player i. So, the probability that a is a pure Nash equilibrium is $1/2^n$. So, we have that, each joint action $a \in \{0, 1\}^n$ is a pure Nash equilibria with probability $1/2^n$, and since there are 2^n possible joint actions, we have that $\mathbf{E}(N) = 1$.

Now since $\mathbf{E}(N) = \sum_{k\geq 1} kPr[N = k]$, we get that

$$\begin{aligned} E[N] &\geq Pr[N = 1] + 2Pr[N \geq 2] \\ &= (1 - Pr(N = 0) - Pr(N \geq 2)) + 2Pr(N \geq 2) \\ &= 1 - Pr(N = 0) + Pr(N \geq 2) \end{aligned}$$

Since we have that $\mathbf{E}(N) = 1$, we get

$$\begin{aligned} Pr[N = 0] &\geq Pr[N \geq 2] \\ &\geq Pr[N = 2] \end{aligned}$$

So, we just need a lower bound on the probability that the $(u_1, ..., u_n)$ has exactly 2 pure Nash equilibria. It turns out that this probability is tricky to calculate. The reason is as follows. Consider the joint actions $a = 1^n$ and $b = 1^{n-1}0$. The probability that a is a pure Nash equilibrium is $1/2^n$. However, given that a is a pure Nash equilibrium, that probability that b is a pure Nash equilibrium is 0. This is because of the definition of the probability distribution $\mathbf{P}$. So, we have that for every pair of distinct strings a, b in $\{0, 1\}^n$, it is not necessarily the case that the event that a is a pure Nash equilibrium is independent of the event that b is a pure Nash equilibrium. We would like to get around this problem.

To this end, consider the set $Z = \{a \in \{0, 1\}^n \ : \ a \text{ has an even number of 1s}\}$. Now, we claim that for every pair of distinct strings $a, b \in Z$, it *is* the case that the event that a is a pure Nash equilibrium is independent of the event that b is a pure Nash equilibrium. To see this, consider two arbitrary but distinct strings $a, b \in Z$. It must be the case that a, b must differ in at least two different positions. This is because, if they differ in only one position, then one of the two strings would end up containing an odd number of 1s, which is a contradiction, since $a, b \in Z$. Hence, for each player i, we have that $a^{-i} \neq b^{-i}$ (Note that this wouldn't have been necessarily true, if a, b were arbitrary but distinct strings from $\{0, 1\}^n$). Now it is easy to calculate the probability γ, that exactly 2 elements of Z are pure Nash equilibria for $(u_1, ..., u_n)$. Since the size of Z is 2^{n-1}, we have that

$$\gamma = \binom{2^{n-1}}{2} (1/2^n)^2 (1 - 1/2^n)^{2^{n-1}-2}$$

which is positive for all $n \geq 2$. Moreover, we have that $\gamma \to (1/8)e^{-1/2} > 0$ as $n \to \infty$. Since, $Z \subset \{0, 1\}^n$, we have that $Pr[N = 0] \geq Pr[N = 2] \geq \gamma$. Hence we conclude that the probability that there is no pure Nash equilibrium for $(u_1, ..., u_n)$ is bounded away from 0. This completes the proof of the lemma.

$\square$

Now we proceed to prove that the probability that $(u_1, .., u_n)$ belongs to a rectangle labelled "no pure Nash equilibrium" is small. Formally,

Lemma 4 *Let $U = U_1 \times ... \times U_n$ be a monochromatic combinatorial rectangle labelled "no pure Nash equilibria". Then,*

$$Pr[(u_1, ..., u_n) \in U] \leq 2^{-2^{n-1}}$$

Proof of Lemma 4 First, we claim that for every $a \in \{0,1\}^n$, there exists a player i such that, $a_i \notin BR(a^{-i}, u_i)$ for every $u_i \in U_i$. To see this, let us assume this is not the case. That is, there exists an $a \in \{0,1\}^n$ such that, for every player i, $a_i \in BR(a^{-i}, u'_i)$, for some $u'_i \in U_i$. Then this would mean that a is a pure Nash equilibrium of the game $(u'_1, ..., u'_n)$, where $(u'_1, ..., u'_n) \in U$. But this is a contradiction, since U is a monochromatic combinatorial rectangle labelled "no pure Nash equilibrium".

Now let us consider the set $Z = \{a \in \{0,1\}^n \ : \ a \text{ has an even number of 1s}\}$ (for a similar reason which will be clear shortly). Since for every $a \in \{0,1\}^n$, there exists a player i such that, $a_i \notin BR(a^{-i}, u_i)$ for every $u_i \in U_i$, the same is true for every $a \in Z$. Let us divide the set Z based on i. That is, Z_i will contain all the $a \in Z$ such that $a_i \notin BR(a^{-i}, u_i)$ for all $u_i \in U_i$. More formally, let

$$Z_i = \{a \in Z \ : \ a_i \notin BR(a^{-i}, u_i) \ \forall u_i \in U_i\}$$

We have that $Z = \cup_{i=1}^{n} Z_i$ (Note that a single $a \in \{0,1\}^n$ can be in multiple Z_is). Now consider the following set

$$\{u_i \ : \ a_i \notin BR(a^{-i}, u_i) \ \forall a \in Z_i\}$$

That is, this set is the set of all payoff functions u_i of player i such that, for every $a \in Z_i$, we have that $a_i \notin BR(a^{-i}, u_i)$. But since the definition of the set Z_i is the set of all $a \in Z$ such that $a_i \notin BR(a^{-i}, u_i)$ for all $u_i \in U_i$, it is not difficult to see that

$$U_i \subseteq \{u_i \ : \ a_i \notin BR(a^{-i}, u_i) \ \forall a \in Z_i\}$$

So, we have that if a payoff function of player i, $u_i \in U_i$, then $u_i \in \{u_i \ : \ a_i \notin BR(a^{-i}, u_i) \ \forall a \in Z_i\}$. Now, for every i, for every distinct pair $(a, b) \in Z_i$, we have that the event $\{u_i \ : \ a_i \notin BR(a^{-i}, u_i)\}$ is independent of the event that $\{u_i \ : \ b_i \notin BR(b^{-i}, u_i)\}$ (for the same reason as discussed in the previous lemma) and each event has a probability of 1/2. So, it is easy to see that

$$Pr[u_i \in U_i] \leq \prod_{a \in Z_i} Pr[u_i \ : \ a_i \notin BR(a^{-i}, u_i)] = 1/2^{|Z_i|}$$

So, since the coin tosses for choosing u_i are independent over i, we have that

$$Pr[(u_1, ..., u_n) \in U] \leq \prod_{i=1}^{n} Pr[u_i \in U_i] \leq 1/2^{\sum_{i=1}^{n} |Z_i|}$$

Since we have that $Z = \cup_{i=1}^{n} Z_i$, it follows that $\sum_{i=1}^{n} |Z_i| \geq |Z| = 2^{n-1}$. Hence, we have that,

$$Pr[(u_1, ..., u_n) \in U] \leq 1/2^{2^{n-1}}$$

This completes the proof of the lemma.

□

Now we complete the proof of Theorem 2 as follows. By Lemma 3 we have that the probability that $(u_1, ..., u_n)$ has no pure Nash equilibrium is bounded from below by $\alpha > 0$. However, by Lemma 4, we have that the probability that $(u_1, ..., u_n)$ belongs to a monochromatic combinatorial rectangle labelled "no pure Nash equilibrium" is small, namely at most $1/2^{2^{n-1}}$. So the number of such monochromatic rectangles should be $\geq \alpha 2^{2^{n-1}}$. This gives us a lower bound of $\log(\alpha 2^{2^{n-1}}) = \Omega(2^n)$ on the expected communication complexity. This completes the proof of Theorem 2.

□

1.6 Approximate Nash Equilibrium

1.6.1 Setup

Here we will present some results from [6]. The setup to this problem is slightly different from the previous section.
We will only focus on the case of two agents, let us call them Row and Column. Furthermore each agent will have a strategy set of n actions, because of this we can represent the payoffs to Row and Column as n by n matrices R and C respectively. Assume that all the entries of R and C are normalised to lie in [0,1]. As before Row and Column know their own payoff functions but not the other players payoff functions.
The communication protocol proceeds as follows, Row will send a message, based on R, to Column. Column sends a message based on C and Rows message back to Row. They alternate in this fashion, each player sending a message based on his matrix and the previous messages, until each player knows a strategy he wishes to play. At this point they will each play their strategy on the payoff matrix [R,C], where $[R, C]_{i,j} = (R_{i,j}, C_{i,j})$. We stress that the player only need to know the strategy they wish to play and not the strategy his counterpart wishes to play.
We start by showing deciding if a pure Nash equilibrium exists in our setup is still hard, this proof can also be found on Noam Nissan's blog [11].

Theorem 5 *The communication complexity of finding a pure Nash equilibrium for two players with large strategy sets is near maximal (in terms of the size of their strategy sets). This is true even if the payoffs are binary.*

Proof We will show this via a set disjointness reduction. Say Row has input string x and Column y, each of length n. Assume that n is a perfect square, if it is not we can just add 0's to the end of x and y until they are of appropriate length. We will form payoff matrices X, from x and Y, from y, such that dim(X)=dim(Y)=($\sqrt{n}+2$,$\sqrt{n}+2$), We will be playing our game on the matrix [X,Y]. Here we show [X,Y] for n=4.

$$\begin{pmatrix} (x_1, y_1) & (x_2, y_2) & (0,1) & (0,1) \\ (x_3, y_3) & (x_4, y_4) & (0,1) & (0,1) \\ (1,0) & (1,0) & (0,1) & (1,0) \\ (1,0) & (1,0) & (1,0) & (0,1) \end{pmatrix}$$

The bits of x and y constitute the elements of the first $\sqrt{n}$ columns and rows of [X,Y] in order. If x and y do intersect about some bit i we see the corresponding element in the matrix must be (1,1) a Nash equilibrium since each player is receiving maximum payoff. The reason the last 2 rows and columns include items of the form (0,1) and (1,0) is to ensure if there is no intersection there will be no Nash equilibrium. Observe if we are currently in a position where Row has 0 payoff he can deviate to one of the last two Rows and get 1 payoff. Column has a similar option. Thus if x and y do not intersect, there is no (1,1) element in the matrix, so the players will never converge to equilibrium. So we require $\Omega(n)$ communication where the player inputs are of size $(\sqrt{n}+2)^2$.

□

1.6.2 Mixed Strategies

We can think of a mixed strategy for a player as a probability distribution over his action set. We can represent a mixed strategy as a length n vector.

Definition A mixed strategy is a vector $[p_1, ..., p_n]$, s.t. $\sum_i p_i = 1$. The intuition is we play strategy i with probability p_i.

Definition The utility of a mixed strategy pair x played by Row and y played by Column under payoff matrix [R,C], are $x^T Ry$ to Row and $x^T Cy$ to Column. That is they aim to maximise their expected payoff.

Nash proved that any game has a mixed strategy Nash equilibrium. Still finding one, in the setting of several agents with binary strategy sets, requires near maximal communication [8].

1.6.3 ϵ-Nash Equilibrium

We will now weaken the requirements of finding an equilibrium by only requiring we find an approximate equilibrium. We say an agent will only deviate if by doing so he can gain a factor of at least ϵ $(0 \leq \epsilon \leq 1)$ to his expected utility. We need the following definitions.

Definition If Column is playing strategy y, the best response of Row is a strategy e_{best} such that for any strategy x, $e_{best}^T Ry \geq x^T Ry$. We can define the best response of Column similarly. It is easy to see that a best response is always a pure strategy.

Definition The regret of Row playing x and Column playing y is the difference between between his expected utility and the expected utility of his best response. That is Row's regret is $e_{best}^T Ry - x^T Ry$. We can define Column's regret in a similar way.

Definition If both Column and Row have regret of at most ϵ we say they are at a ϵ-Nash Equilibrium.

1.6.4 Mini-Max

We will often use the Mini-Max result in our proofs. This result was first proven by Von-Neumann.

Theorem 6 *Consider the zero sum game [A,-A] for some square matrix A. Say [A,-A] has a Nash equilibrium strategy pair (x,y) for Row and Column, and expected payoff of v for Row and -v for column.*

By playing y Column limits Rows payoffs to at most v, equivalently Column guarantees himself a payment of at least -v. Similarly by playing x Row limits Columns payoffs to at most -v, and guarantees himself a payoff of at least v.

The Mini-Max name comes from fact players try to minimise the maximum payment given to the other player, equivalently they minimise their maximum losses.

1.6.5 No communication

Theorem 7 *It is possible to achieve a 0.75-Nash Equilibrium using no communication.*

Proof For Row let his i'th strategy be the best response to Column taking column 1 of C. For Column let his j'th strategy be the best response to Row taking row 1 of R.
Row will now play row 1 with probability 0.5 and row i with probability 0.5. Column is defined similarly for column 1 and j.
Let us now look at the regret of Row. Let his best response to Column be row b.
Regret of Row is:
$(\frac{1}{2}R_{b,1} + \frac{1}{2}R_{b,j}) - (\frac{1}{4}R_{1,1} + \frac{1}{4}R_{1,j} + \frac{1}{4}R_{i,1} + \frac{1}{4}R_{i,j})$
Since i is Row's best response to Column taking 1 it must be that $R_{b,1} \leq R_{i,1}$, we get the above value must be at most:
$(\frac{1}{4}R_{b,1} + \frac{1}{2}R_{b,j}) - (\frac{1}{4}R_{1,1} + \frac{1}{4}R_{1,j} + \frac{1}{4}R_{i,j})$
Furthermore since each of the entries of R is in [0,1] the term on the right of the difference can't be negative. Also the term on the left is maximised when each of the matrix entries is 1. Thus we get it must be at most:
$(\frac{1}{4}1 + \frac{1}{2}1) = \frac{3}{4}$, which completes the proof.

□

Theorem 8 $\forall\delta > 0$ *it is impossible to achieve a (0.5-δ)-Nash Equilibrium using no communication.*

Proof The reader is referred to [6] for the proof.

1.6.6 0.438-Nash Equilibrium using little communication

Theorem 9 *There is a protocol using $O(log^2(n))$ 2-way communication that finds a 0.438-Nash Equilibrium.*

Proof
We will first show how to find an α-equilibrium and later see the minimum value of α is 0.438.
First Row will find a solution (x_r^*, y_r^*) to the zero sum game [R,-R]. Column will find a solution (x_c^*, y_c^*) to [-C,C], also a zero sum game. Let us say the values specified by mini-max to these games are v_r and v_c for Row and Column respectively. We have 2 cases to consider.

Case 1 *If v_r, $v_c \leq \alpha$:*

Row will communicate to Column y_r^*, Column will communicate to Row x_c^*. We will now play the game with strategy pair (x_c^*, y_r^*). We will show how to communicate these strategies in the next section.
The intuition is each player is playing a strategy that mini-max guarantees will limit his counterpart to a payoff of most v_r or v_c which are both less than α. Since they can't get better than an α payoff their regret must also be at most α. Thus we are at an α-NE

Case 2 *If at least one of $v_r, v_c > \alpha$ (Say w.l.o.g. $v_r > \alpha$):*

Row sends to Column x_r^*. Column will send back to Row his best response, a pure strategy, e_c to x_r^*. If we were to play the game using (x_r^*,e_c), we see that Column would have no regret since he is playing his best response. Row may have regret larger than α so we need to modify his strategy a bit.
Say that Row's best response to e_c is e_b with payoff β. Take $\frac{1}{2}\alpha$ probability from the pure strategy that has the worst payoff and give it to e_b. If the worst payoff strategy did not have enough probability repeat the process until we shift enough probability. The intuition is we are still getting at least $(1 - \frac{1}{2}\alpha)\alpha$ payoff from Row's non best response strategies and $(\frac{1}{2}\alpha)\beta$ more payoff from his best response.
At worst the regret of column is now α. This is because Row shifting $\frac{1}{2}\alpha$ probability at most made his current strategy appear $\frac{1}{2}\alpha$ worse and some other strategy appear at most $\frac{1}{2}\alpha$ better. It turns out the minimum α where we can guarantee Row will not deviate is α=$\frac{1}{2}(5 - \sqrt{17})$, which is roughly 0.438.

□

1.6.7 How to communicate information for the algorithm

By sending one bit each we can identify which of the two cases we are in. To communicate a pure strategy we can just send $log(n)$ bits as the index for the pure strategy. Instead of communicating a mixed strategy we will instead communicate an approximation of it with some nice properties. We will get a 0.438+δ approximation for any $\delta > 0$.
Let k=$\frac{ln(n)}{\delta^2}$ Let's say that we wish to send mixed strategy x. Let A be a multiset of pure strategies formed by sampling from x k times according to the probability distribution x. Now we will approximate x with x'. For each pure strategy in our action set let us play it with probability $\frac{c}{k}$ where c is the number of times that pure strategy appears in A.
We want the expected utility of x' to be not too far from x. Let us consider any mixed strategy y and payoff matrix R. Let the event $\phi = \{|x'^T Ry - x^T Ry| \leq \delta\}$. As in [10] using some probability theory, and applying a tail inequality, we get $\Pr[\phi] \geq 1 - \frac{1}{2^n}$. Thus if our estimation is too far off we can just re-sample until we succeed.
To send x' we can just send every element of A to our counterpart and let him reconstruct x'. Each element takes log(n) bits to specify and $|A| \in O(log(n))$. Thus we need to use $O(log^2(n))$ bits.
In the first case where each zero sum game had a solution value of at most 0.438, when we communicate our limiting strategy to our counterpart at worst this strategy limits us to a payoff of 0.438+δ. Thus our regret is at most 0.438+δ. In the second case where Row's expected payoff was greater than 0.438 we still achieve a 0.438 equilibrium. This is because we just relied on Row

playing x_r^* having a guaranteed expected payoff of at least 0.438, so we can always find an x' such that it gets at least 0.438 payoff by taking a small enough δ.

1.6.8 Nash Equilibrium Conclusion

In the approximate Nash equilibrium setting the biggest open question is how good are the factors of ϵ. In the communication free setting it would be interesting to see if one could increase the 0.5 lower bound or find a protocol that beats 0.75. As for the 2-way communication model no known lower bounds exist.

The 0.438 upper bound is inspired by an algorithm for the uncoupled model that starts with a solution to the zero sum game [R-C,C-R]. Here it is possible to achieve a 0.36395 approximation [1] in polynomial time. It would be interesting to see if a algorithm that is not based on a polynomial time algorithm could achieve a better ϵ than 0.438.

2 Algorithmic Mechanism Design

Before we talk about algorithmic mechanism design it will be helpful to introduce a problem we wish to study, so our first section will focus on the combinatorial auctions problem.

2.1 Combinatorial Auctions

In the Combinatorial Auctions problem there is a set of m items M, and n agents N. There is also a valuation profile $v = (v_1, ..., v_n)$ where agent i is given valuation function v_i which is only known to him. Here for each i, $1 \leq i \leq n$, $v_i : 2^{[m]} \to \mathbb{R}^{\geq 0}$. We require that v_i is submodular and monotone (see definition below). Also, we assume that for every $i \leq n$, $v_i(\emptyset) = 0$. We restrict our attention to submodular functions since they are widely studied in economics and they capture the notion of diminishing marginal returns.
We wish to pick a partition $A = A_1, ..., A_n$ of M (which we call an *allocation*) s.t. $\sum_i v_i(A_i)$ is maximised, that is, we think of A_i as a set of items allocated to agent i, and our goal is to maximize the social welfare $\sum_i v_i(A_i)$.

Definition A valuation function $v_i : 2^{[m]} \to \mathbb{R}^{\geq 0}$ is submodular if $\forall S, T \subseteq M$ such that $S \subseteq T$, and all $x \in M \backslash T$, $v_i(x \cup T) - v_i(T) \leq v_i(x \cup S) - v_i(S)$. It is monotone if $v_i(S) \leq v_i(T)$, for all $S, T \subseteq M$ such that $S \subseteq T$.

2.2 The Mechanism

For a Combinatorial Auction problem with item set M, agent set N, and valuation profile $v = (v_1, ..., v_n)$, we would like to design a centralized algorithm, with certain properties (which we will state shortly). This algorithm, called the mechanism $\mathcal{M}$, will communicate with the agents over a series of rounds in an effort to learn about their underlying valuation functions. At the conclusion of this process, $\mathcal{M}$ will pick a partition $A = A_1, ..., A_n$ of M, where agent i is assigned set A_i, in an effort to maximise $\sum_i v_i(A_i)$.
The game theoretic aspect to our problem is each agent i wants to be allocated a subset A_i s.t. $v_i(A_i)$ is maximised. Thus the agents will provide incorrect information to the mechanism,

regarding their valuation function, if they will receive a better subset at the conclusion of the protocol.
Thus we would like our mechanism to satisfy three criteria, quality of solution, efficient to compute and incentive compatible (immune to agent manipulation). We formally define these criteria in the next three sections.

2.3 Quality of Solution

We define the approximation ratio of the mechanism as the ratio between what it returns and the optimum.

Definition $\mathcal{M}$ is a α approximation if $\forall$ valuation profiles $v = (v_1, ..., v_n)$ if the mechanism returns partition $A = (A_1, ..., A_n)$ and the optimal partition is $O = (O_1, ..., O_n)$ then $\frac{\sum_i v_i(O_i)}{\sum_i v_i(A_i)} \leq \alpha$.

2.4 Efficiency

Our notion of efficiency will be in the communication model, where we place no restrictions on the computational powers of the mechanism and the agents, and are only interested in the amount of communication that occurs between them. We will now define the model more formally. Recall that M is the set of items, N is the set of agents, and each agent, as input, is given a private valuation function v_i as defined previously.
The communication will proceed in a series of rounds. In each round the mechanism is allowed to query one of the agents for information about v_i, we allow the query to be based on the previous queries and responses to them. We allow the agents' reply, which will be seen by all other agents and the mechanism, to be based on the query, his underlying valuation function v_i and the previous queries and responses of any agent.
Formally if agent i is queried in round l his response is some function $r_i(v_i, Q_l, R_{l-1})$, where v_i is his valuation function, Q_l is the set of queries asked so far (including the current one) and R_{l-1} is the responses to the queries (up to the last one).
At the conclusion of this protocol the mechanism will announce a partition $A = A_1, ..., A_n$ of M, based on the responses to his queries, where agent i gets set A_i.

- One of the most common queries that appears in the literature is the valuation query, where the mechanism asks agent i, the value of $v_i(S)$, for some $S \subseteq M$.
- In our lower bound proofs we allow the queries to be anything.

Definition The communication complexity of a mechanism $\mathcal{M}$, with agent set N and item set M, on valuation profile $v = (v_1, ..., v_n)$ is the number of bits communicated over the execution of the protocol. The number of bits is a function of $|N|$ and $|M|$.
The communication complexity of a mechanism is the maximum number of bits communicated over all valuation profiles $v = (v_1, ..., v_n)$.

The goal in algorithmic game theory is to design mechanisms whose communication complexity is polynomial in $|N|$ and $|M|$.

One may question why we even study this from a communication standpoint. In fact a divergent avenue of research assumes the agents, using some succinct representation, hand their entire valuation functions to the mechanism. In this non communication model the mechanism must run in time $poly(|M|, |N|)$.
There are two main reasons for studying the communication model. First the valuation functions can be exponentially large in $|M|$, even if we only consider submodular functions. It would be unreasonable to assume that the agents must send these large objects to the mechanism. Also the agents may be fine revealing a bit of their private valuations but not all of it, this is especially true in settings where knowledge of their valuations could prove advantageous to the mechanism or other agents in future games.

2.5 Truthfulness

In order to motivate the agents to be honest with their valuations we allow the mechanism to charge payments to the agents at the end of communication protocol. That is, in addition to deciding upon an allocation $A = (A_1, ..., A_n)$, the mechanism $\mathcal{M}$ will also pick a payment scheme $p = (p_1, ..., p_n)$ where agent i is charged $p_i \in \mathbb{R}^{\geq 0}$. Each agent now has the goal of maximising his valuation utility minus his payment, that is agent i wishes to maximise $u_i(A_i) = v_i(A_i) - p_i$.
Now we will define incentive compatibility, our notion of truthfulness.

Definition A mechanism $\mathcal{M}$ is incentive compatible if for each valuation profile $v = (v_1, ..., v_n)$ the following holds. For each agent i, his utility (u_i) is maximized if he answers all his queries using his actual valuation function v_i.

Intuitively, if $\mathcal{M}$ is incentive compatible, then for any agent, it is in his best interest to respond to the queries using his true valuation function, regardless of the other agents' valuation functions.

2.6 VCG, VCG based mechanisms, and MIR algorithms

From the previous sections, it is clear that a mechanism can be thought of as an (*allocation algorithm, payment scheme*) pair, where the allocation algorithm picks an allocation (which is a partition of M into n sets) from the set of all possible allocations (by communicating with the agents), and the payment scheme charges payments to each agent, based on the allocation picked by the allocation algorithm.

A major achievement in the field of mechanism design was the introduction of a general method of construction of incentive compatible mechanisms called the Vickrey-Clarke-Groves (VCG) method [15] [2] [7]. VCG method has a specific payment rule, that when applied to the output of an allocation algorithm that computes the optimal allocation (allocation which maximizes social welfare), guarantees incentive compatibility. A little more formally, a VCG mechanism is an (*allocation algorithm, payment scheme*) pair, (K, P), where for every valuation profile $v = (v_1, ..., v_n)$, the allocation algorithm K returns the allocation that maximizes the social welfare (that is, for every $v = (v_1, ..., v_n)$, K on v returns $arg\,max_{A \in O} \sum_i v_i(A_i)$, where O is the set of all possible allocations), and the payment scheme $P = (P_1, ..., P_n)$ is as follows: For each i, $P_i(v) = \sum_{j \neq i} v_j(A_j) + h_i(v_{-i})$ (where $A = (A_1, ..., A_n)$ is the allocation returned by the allocation algorithm K on the valuation profile v, and h_i is an arbitrary function of v_{-i}, where $v_{-i} = (v_1, ..., v_{i-1}, v_{i+1}, ...v_n)$). We won't be going into detail about the payment scheme P, but the intuition is that each agent is being charged the "social cost" of him receiving A_i, that is incurred by the rest of the agents, and it is not difficult to prove that VCG mechanisms are indeed incentive compatible [7].

However, the problem is apparent when we apply this scheme to combinatorial auctions: The allocation algorithm of a VCG mechanism is required to compute *optimal allocation*, and hence is not efficient, since the range of possible allocations for a combinatorial auction problem is huge, and we should maximize the social welfare over this huge range (the allocation algorithm is not efficient in the sense that it is not feasible to compute the optimal allocation with, say polynomial (in $|M|, |N|$) communication).

So, a natural idea is to retain the VCG payment scheme and replace the optimal algorithm by an algorithm that is approximately optimal but efficient (say that requires only polynomial (in $|M|, |N|$) communication). This idea was introduced in [12] and they termed such mechanisms as *VCG-based* mechanisms. [12] showed that if we replace the optimal algorithm by an arbitrary sub-optimal (but efficient) algorithm (and apply the VCG payment scheme to the output of this algorithm), then the resulting mechanism is not necessarily incentive compatible. However, they proved that the following family of sub optimal allocation algorithms, called the Maximal in Range (MIR) algorithms, do yield incentive compatible VCG-based mechanisms.

Definition An allocation algorithm is called Maximal In Range (MIR), if there exists a subset $R \subseteq O$ (where O is the set of all possible allocations), such that for every $v = (v_1, ..., v_n)$, the algorithm outputs an allocation that has the maximum social welfare, among all the allocations in R. That is, for all $v = (v_1, ..., v_n)$, the algorithm outputs $arg\, max_{A \in R} \sum_i v_i(A_i)$. R is called the range of the MIR algorithm.

So, an MIR algorithm maximizes the social welfare over some subset $R \subseteq O$. It is not difficult to see that the VCG-based mechanisms where the allocation algorithm is an MIR algorithm, are indeed incentive compatible [12], however, the main result of [12] is that this is essentially it, no other VCG-based mechanisms are incentive compatible. Note that the algorithm that outputs the optimal allocation is an MIR algorithm with range $R = O$ (so the approximation ratio is 1, however the algorithm is not efficient). The algorithm which always chooses only one allocation, say the allocation where we allocate all items to agent 1, is also an MIR algorithm, with the range R containing only that allocation. Though such an algorithm is very efficient, it provides a very poor approximation ratio.

2.7 Limitations of VCG based Mechanisms

In this section, we will present the results from the paper "Limitations of VCG-based mechanisms" by Dobzinski and Nisan [4]. The main result of this paper provides a lower bound for the approximation factor that can be achieved by incentive compatible VCG-based mechanisms (that make a sub-exponential number of queries to the bidders) for combinatorial auctions with submodular valuations. The best known mechanism for this case is the $O(\sqrt{m})$-approximation mechanism of [5]. [4] show that this is almost the best approximation one can get by showing a $m^{1/6}$ lower bound. We first state their theorem,

Theorem 10 *Every incentive compatible VCG-based mechanism for approximating the social welfare in combinatorial auctions with submodular bidders that uses a sub-exponential number of queries to the bidders achieves an approximation factor of* $\Omega(min(n, m^{1/6}))$.

The proof proceeds by combinatorially analysing MIR algorithms (this is sufficient because of the main Theorem of [12]). The main idea is, if the range is "small", then we get a bad approximation. So the range should be "large". However, if the range is "large", then optimizing over it needs exponential communication. A little more formally, we define two complexity measures for the range R of an MIR algorithm A, called $cover(R)$ and $intersect(R)$, and adopt the following outline.

1. Show, using probabilistic method, that if $cover(R)$ is "small", then there exists an instance such that A fails to provide a good approximation. Therefore the range must be "large".
2. Show that if the $cover(R)$ is large, then $intersect(R)$ is exponential.
3. Show that $intersect(R)$ serves as a lower bound to the communication complexity of A.

$cover(R)$ roughly corresponds to the size of the range R [9]. We will shortly define what $intersect(R)$ means. We first state the lemma corresponding to (1) of the proof outline, without the proof.

Lemma 11 *Let A be an MIR algorithm with range R. If $cover(R) < e^{m/300n}$, then there is an instance in which A provides no more than $1.01/n$-fraction of the welfare.*

The high level idea of this proof is to construct a random instance of combinatorial auctions and prove that the value of optimal solution in the random instance is exactly m, however with non-negligible probability, the welfare provided by the MIR algorithm A with range R (with small cover number), is only $1.01m/n$. We direct the readers to [4] for the full proof.

Next, we state the lemma corresponding to (2) of the proof outline, without the proof.

Lemma 12 *Let R be a set of allocations with $intersect(R) \leq d$. Then, we have that*

$$cover(R) < (8d)^{m^{3n/5}} m^{4m^{2/5}n^2}$$

This is a pure combinatorial lemma, which relates the two measures defined on the range R of an MIR algorithm. This lemma is rather difficult to prove, and we direct the readers to [4] for the proof.

We now proceed to state and prove the lemma corresponding to (3) of the proof outline, which is the relevant result to us from a communication complexity perspective. Before that, we state some definitions, including the definition of the intersection number.

Definition We say that a set of allocations R is *regular* if there exists constants $s_1, ..., s_n$ such that for all $S \in R$ and for all $1 \leq i \leq n$, it holds that $|S_i| = s_i$.

[9] It is not exactly the size of R. We exclude certain "degenerate" allocations which don't affect the guaranteed approximation ration of the MIR algorithm. However, we won't be formally defining $cover(R)$, so for the purpose of this document, $cover(R)$ can be thought of as representing the size of the range R

So, if the allocations are chosen from a set of allocations R which is regular, then no matter what the allocation is, for each i, agent i ends up getting the same number of items, namely s_i.

Now, we proceed to define what be mean by an intersection set and the intersection number of a set of allocations.

Definition A set of allocations $\mathcal{D}$ is called an (i, j) intersection set if for each $D, D' \in \mathcal{D}$ such that $D \neq D'$, it holds that $D_i \cap D'_j \neq \emptyset$.

Definition Define the intersection number of a set of allocations R, denoted by $intersect(R)$ to be the maximum cardinality regular (i, j) intersecting set which is a subset of R, where the maximum is taken over all pairs of bidders i and j.

So, for a set of allocations R, we consider the power set of R. From this power set, we only consider the sets (which are nothing but the subsets of R) which are regular. Out of these, for each distinct i, j, $1 \leq i, j \leq n$, we consider the sets (which now are subsets of R that are regular), that are (i, j) intersection sets, and find the set the has maximum size among these (where the maximum is taken over all distinct i, j), and the size of this set is the intersection number of R [10].

Now, we proceed to prove that the intersection number of the range of a MIR algorithm serves as a lower bound to the communication complexity of the algorithm.

Lemma 13 *Let A be a MIR algorithm for combinatorial auctions with submodular bidders, with range R. Let $intersect(R) = d$. Then the communication complexity of A is $\Omega(d)$.*

Proof The outline of the proof is as follows. We reduce from set disjointness. Alice and Bob each hold a $d-$ bit string, $a_1, ..., a_d$ and $b_1, .., b_d$ respectively (where d is the intersection number of the range R of the MIR algorithm A), the input to the set disjointness problem. Given, the input to the set disjointness problem, we construct a combinatorial auction with m items and n submodular bidders such that, the MIR algorithm A with range R (with intersection number d) outputs a specific fixed value (which we will specify shortly) for this auction if there exists k such that $a_k = b_k = 1$ (that is, if the sets represented by the inputs to the set disjointness problem are not disjoint) and outputs a value strictly less than the specific fixed value if the sets represented by the inputs to the set disjointness problem are disjoint. More formally,

Let $\mathcal{D} = \{D^1, ..., D^d\}$ be an (i, j) intersection set of the range R (we know such a set exists for some distinct i, j, $1 \leq i, j \leq n$, since the intersection number of R is d). By definition, $\mathcal{D}$ is regular. So, for each bidder t, there exists a constant s_t such that $|D_t| = s_t$ for every $D \in \mathcal{D}$.

We will construct a combinatorial auction with m items as follows. Alice will play the role of bidder i. Bob will play the role of rest of the bidders, especially bidder j. We now define the valuation functions for each of the bidders. We specify the valuation function of bidder i played by Alice.

$$v_i(S) = \begin{cases} |S|, & \text{if } |S| \leq s_i - 1 \\ s_i, & \text{if } |S| \geq s_i \text{ and } \exists k \text{ s.t. } D^k \subseteq S \text{ and } a_k = 1 \\ s_i - 1/2^{|S| - s_i + 1}, & \text{otherwise} \end{cases}$$

[10] We acknowledge that this measure looks somewhat artificial, and unlike the cover number, there is no clear intuition as to what characteristic/property of the range this measure exactly captures.

The valuation v_j is defined in an analogous way. The valuations of the rest of the bidders is zero on any input.

First, we claim that the valuations are submodular. This is not difficult to see, and we just give the outline of the proof. To say that v_i is submodular, we prove that for every $S \subseteq M$, and $x_1, x_2 \in M \backslash S$,

$$v_i(S \cup \{x_1, x_2\}) + v_i(S) \leq v_i(S \cup \{x_1\}) + v_i(S \cup \{x_2\})$$

We select an arbitrary S, and we do case analysis based on the size of S to show that the above equation is true for each of the cases [11].

Now we claim that if there exists k such that $a_k = b_k = 1$, then the optimal welfare is exactly $s_i + s_j$, otherwise the optimal welfare is strictly less than $s_i + s_j$. To see this, note that if there exists k such that $a_k = b_k = 1$, then the valuation v_i on any superset of D_i^k is s_i and the valuation v_j on any superset of D_j^k is s_j. This is by definition of the valuation functions. So, in particular, the allocation D^k yields the social welfare of $s_i + s_j$, which is indeed optimal [12].

Now we prove the other direction. Observe that on an allocation, if bidder i gains s_i (say his allocation is S_1) and bidder j gets s_j [13] (say his allocation is S_2), then by definition there exists k, k' such that $D_i^k \subseteq S_1$ and $a_k = 1$, and $D_j^{k'} \subseteq S_2$ and $b_{k'} = 1$. However since $\mathcal{D}$ is an intersection set, we have that $D_i^k \cap D_j^{k'} \neq \emptyset$. So it must be the case that $S_1 \cap S_2 \neq \emptyset$, so the allocation is not a valid allocation, and hence k must equal k'.

Finally, by construction, if the optimal welfare is $s_i + s_j$, then it can be achieved by an allocation in R. Since A is an MIR algorithm with range R, A must return $s_i + s_j$ in this case (and obviously A returns value strictly less than $s_i + s_j$ if the optimal welfare is strictly less than $s_i + s_j$). So, we will be able to decide if there if there exists k such that $a_k = b_k = 1$ based on the value returned by the MIR algorithm A. Hence, we conclude that the communication complexity of A is at least as that of the set disjointness problem, which is $\Omega(d)$. This concludes the proof of Lemma 13.

□

It is not difficult to see that we can combine Lemmas 11, 12, 13 in a straightforward way to prove Theorem 10.

2.8 The Combinatorial Public Projects Problem (CPPP)

We now present the main result from [13] showing that no mechanism for the CPPP that satisfies all three of our criteria exists. The setup to the CPPP is similar to the Combinatorial Auction problem.

We have a set N of n players, and a set M of m resources with a constant k. We also have a valuation profile $v = (v_1, ..., v_n)$, player i is given valuation v_i which is only known to him. Here for each i, $v_i : 2^{[m]} \rightarrow \mathbb{R}^{\geq 0}$.

[11] Based on the definition of the v_i and the definition of submodularity we chose to use, we can think of four different cases, $|S| < s_i - 2$, $|S| = s_i - 2$, $|S| = s_i - 1$, and $|S| > s_i - 1$.

[12] By definition, the maximum value that v_i can evaluate to is s_i, and for v_j it is s_j and zero for rest of the bidders

[13] That is, the optimal welfare is $s_i + s_j$.

We require that each v_i is submodular. That is $\forall S, T \subseteq M$ where $S \subseteq T$, and for all $x \in M \backslash T$, $v_i(x \cup T) - v_i(T) \leq v_i(x \cup S) - v_i(S)$. We also require that each v_i is monotone, that is $v_i(S) \leq v_i(T)$, for all $S, T \subseteq M$ such that $S \subseteq T$. Finally we require that each v_i is normalized, that is $v_i(\emptyset) = 0$.
The goal of the mechanism $\mathcal{M}$ is to pick a set of items $A \subseteq M$ s.t. $|A| = k$ and maximise $\sum_{i=1}^{n} v_i(A)$.
The way to perceive this problem is we have a set of potential resources M, say potential hospital locations, and the budget to build k of them that will be shared amongst the agents of N. We wish to maximise the social welfare of the agents, that is the sum of their valuations on the shared subset of M.
We define the quality of the solution in the same way as before, the ratio between the optimal and the algorithm for any valuation profile.
The communication model, and our notion of communication complexity, is exactly the same as in the combinatorial auction setting. That is the mechanism will make a series of queries to the agents regarding their valuation functions. After some number number of rounds the mechanism will pick $A \subseteq M$ s.t. $|A| = k$ as the solution.
As in the combinatorial auction case to ensure his queries are answered honestly the mechanism will also create a payment scheme $p = (p_1, ..., p_n)$ where each agent is charged their corresponding payment. As before each agent i aims to maximise $v_i(A) - p_i$.

Theorem 14 *Any incentive compatible mechanism that achieves a $m^{\frac{1}{2}-\epsilon}$ approximation for any $\epsilon > 0$ requires communication $\Omega(2^{m^\epsilon})$*[14]

Proof The proof proceeds in three steps. We will only outline the first two steps since they are not very interesting from a communication standpoint. See [13] for the full proof. We first need to define the following two concepts.

Definition The range of a mechanism $\mathcal{M}$ is $R_{\mathcal{M}}$=$\{A \subseteq M : |A| = k$ and $\exists$ a valuation profile $v = (v_1, ..., v_n)$ such that the mechanism $\mathcal{M}$ on v outputs A $\}$.

Definition A mechanism $\mathcal{M}$ is an affine maximiser if for every valuation profile $v = (v_1, ..., v_n)$, it pick $argmax_{A \in R_{\mathcal{M}}}(\sum_i w_i v_i(A) + C_A)$.

For $1 \leq i \leq n$, w_i is a non negative constant associated with player i. We require at least one of them is non zero.
For every $A \in R_{\mathcal{M}}$, C_A is a non negative constant associated with outcome A. We do not require at least one of them to be non zero.
The definition of affine maximiser mechanism (as described above) can be seen as a weighted version of maximal in range algorithms (defined in the Section 2.6).

Lemma 15 *(Affine Maximiser) If mechanism $\mathcal{M}$ is incentive compatible for CPPP, $\mathcal{M}$ is an affine maximiser.*

Proof The proof of the lemma is a modification of the proof of Roberts' theorem [15], which proved the same results for general problems and valuation functions.

[14]There is a simple greedy incentive compatible mechanism that achieves an $O(\sqrt{m})$ approximation for the CPPP, thus this lower bound is tight [14].
[15]See [9] for two recent proofs of Roberts' theorem.

Lemma 16 *(Large Range) Any affine maximiser that achieves an approximation ratio of* $m^{\frac{1}{2}-\epsilon}$ *requires a range of size* $\Omega(2^{m^\epsilon})$*. For this lemma we will fix* k *to be* $\sqrt{m}$*.*

Proof This will be done via a probabilistic argument. Informally we construct a set $V \subset M$, by adding each element $a \in M$ to V independently with probability $m^{-\frac{1}{2}+\epsilon}$. We then consider the game consisting of 1 agent with a valuation function $v(S) = |V \cap S|$. Using Chernoff bounds we get that with high probability no solution contained in $R_{\mathcal{M}}$ will serve as a good approximation to the optimal solution, unless $|R_{\mathcal{M}}| \in \Omega(2^{m^\epsilon})$.

Now we can prove that for 2 players we require communication of size $\Omega(|R_{\mathcal{M}}|)$ via a reduction from 2 player set disjointness.
Let us say each player has a private set of outcomes he prefers $R_i \subset R_{\mathcal{M}}$ for $i \in \{1, 2\}$. Define his valuation as $v_i(T) = \beta_i(\prod_{S \in R_i} |S| - \prod_{S \in R_i}(|S| - |S \cap T|))$, where β_i is some constant. To show this function is submodular we can perform induction on $|R_i|$.
First we note for the affine maximiser can assume that the agent constants $w_1 = w_2 = 1$. If this were not the case we could just scale the β_i value of the agents. Furthermore we can assume that each of the outcome constants C_j is 0 since we can just scale up each β_i to make C_j negligible.
For each $T \in R_i$, v_i is maximised. This is because the first product is independent of the solution set T so it is just a constant for the game. On the other hand since $|T \cap T| = |S|$ the second product, a non positive value, will become 0. If $T \notin R_i$ we note the value to agent i must be smaller. Again because in the second product all of the item in it will have value at least 1, thus it must be negative.
Thus if $|R_1 \cap R_2| > 0$ the set the affine maximiser picks must lie in the intersection of the two, and the sum of the 2 utilities must be $k^{|R_1|} + k^{|R_2|}$ where k is the number of elements we activate (each S is of size k since it is an outcome).
Now the reduction from set disjointness is obvious. If Alice gets x and Bob gets y each of size $|R_{\mathcal{M}}|$. They form sets R_1 where we include the i^{th} outcome iff x has a 1 in the i^{th} bit, and do the same for R_2 and y. They both simulate the mechanism asking the queries it would ask of each other until they find a solution. If both Alice and Bob have achieved an optimal value for their set functions they conclude that their sets must intersect, otherwise they conclude they must not intersect.
Thus we see we need $\Omega(|R_{\mathcal{M}}|)$ communication. Assuming $\mathcal{M}$ achieves an approximation ratio of $m^{\frac{1}{2}-\epsilon}$ we require $\Omega(2^{m^\epsilon})$ communication.

$\square$

2.8.1 Mechanism Design Conclusion

The biggest open question now is are the techniques of [13] applicable to different problems. It would be interesting to see if we can use similar arguments to get strong lower bounds on incentive compatible mechanisms for the combinatorial auctions problem, not just VCG based ones.

References

[1] H. Bosse, J. Byrka, and E. Markakis, *New algorithms for approximate nash equilibria in bimatrix games*, Theoretical Computer Science, 411 (2010), pp. 164 – 173.

[2] E. H. Clarke, *Multipart pricing of public goods*, Public Choice, 2 (1971), pp. 19–33.

[3] V. Conitzer and T. Sandholm, *Communication complexity as a lower bound for learning in games*, in IN PROCEEDINGS OF THE 21ST INTERNATIONAL CONFERENCE ON MACHINE LEARNING, Springer, 2004, pp. 185–192.

[4] S. Dobzinski and N. Nisan, *Limitations of vcg-based mechanisms*, in Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing, STOC '07, New York, NY, USA, 2007, ACM, pp. 338–344.

[5] S. Dobzinski, N. Nisan, and M. Schapira, *Approximation algorithms for combinatorial auctions with complement-free bidders.*, in STOC, H. N. Gabow and R. Fagin, eds., ACM, 2005, pp. 610–618.

[6] P. Goldberg and A. Pastink, *On the communication complexity of approximate nash equilibria*, in Algorithmic Game Theory, M. Serna, ed., Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 192–203.

[7] T. Groves, *Incentives in Teams*, Econometrica, 41 (1973), pp. 617–631.

[8] S. Hart and Y. Mansour, *How long to equilibrium? the communication complexity of uncoupled equilibrium procedures.*, Games and Economic Behavior, 69 (2010), pp. 107–126.

[9] R. Lavi and N. Nisan, *Two simplified proofs for roberts theorem.*

[10] R. J. Lipton, E. Markakis, and A. Mehta, *Playing large games using simple strategies*, in Proceedings of the 4th ACM Conference on Electronic Commerce, EC '03, New York, NY, USA, 2003, ACM, pp. 36–41.

[11] N. Nisan, *Communication complexity of reaching equilibrium.* `http://agtb.wordpress.com/2009/08/18/communication-complexity-of-reaching-equilibrium/`, 2009.

[12] N. Nisan and A. Ronen, *Computationally feasible vcg mechanisms*, in EC '00: Proceedings of the 2nd ACM conference on Electronic commerce, New York, NY, USA, 2000, ACM Press, pp. 242–252.

[13] C. H. Papadimitriou, M. Schapira, and Y. Singer, *On the hardness of being truthful.*, in FOCS, IEEE Computer Society, 2008, pp. 250–259.

[14] M. Schapira and Y. Singer, *Inapproximability of combinatorial public projects*, in Internet and Network Economics, C. Papadimitriou and S. Zhang, eds., vol. 5385 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 351–361.

[15] W. Vickrey, *Counterspeculation, Auctions and Competitive Sealed Tenders*, Journal of Finance, (1961), pp. 8–37.