On Two Player Simultaneous Combinatorial Auctions
Lecture notes – 12/08
Scribe: Hrishikesh Khandeparker

These are rough lecture notes for a presentation on the paper [BMW17] given by Hrishikesh Khandeparker for the final presentation in COS597D - Lower Bounds in Computational Models Seminar, Fall 2017.

Contents

1 The Problem 2
  1.1 Setting ........................................................................................................ 2
  1.2 Valuation function types ............................................................................... 3
  1.3 Results of the paper ..................................................................................... 3

2 Motivation, and a brief history of how we got here 4
  2.1 What is the price of truthfulness? ................................................................. 4
  2.2 Seeming contradiction? ............................................................................... 5
  2.3 Justification for the communication setting and a reduction to simultaneous communication protocols ................................................................................... 5

3 Main results and some proofs 6
  3.1 Warmup ....................................................................................................... 6
  3.2 $k, \alpha$-summaries ..................................................................................... 10

4 Summary of results 11
  4.1 Lower bounds for communication .............................................................. 11

5 Recap 12
  5.1 Main Results ............................................................................................... 13
  5.2 Open Problems ............................................................................................ 13
1 The Problem

We first look at a quick summary of the setting of the main problem at hand and what the “simultaneous” in the title means.

1.1 Setting

The general setting for the problem of welfare maximization is as follows (we only define the 2 player case).

**Definition 1.1** (Valuation function). A valuation function \( v : 2^{|m|} \rightarrow \mathbb{R}_+ \), is a mapping from subsets of \([m]\) to positive real numbers.

Given two valuations \( v_a, v_b \) we define “social welfare” as follows.

**Definition 1.2** (Social Welfares). The social welfare of a partition \( S, \bar{S} \) is \( SW(S) := v_a(S) + v_b(\bar{S}) \). The optimal social welfare achievable is \( SW_{OPT}(v_a, v_b) := \max_{S \subseteq [m]} SW(S) \).

Knowing \( SW(v_a, v_b) \) allows a solution to decision problem of “is the optimal social welfare atleast X”. Also, the optimal allocation is also desirable and is defined as

**Definition 1.3** (Optimal Allocation). \( S_{OPT} := \arg \max_{S \subseteq [m]} v_a(S) + v_b(\bar{S}) \).

This allows a solution to the search problem of “what partition of the set \([m]\) leads to the maximum welfare”.

Thus, the problem is defined as

**Definition 1.4** (Welfare maximization problem). Given poly-time oracle access to value queries for two valuation functions \( v_a(\cdot), v_b(\cdot) \), output \( S_{OPT} \) (for search problem) and/or \( SW_{OPT}(v_a, v_b) \) (for decision problem).

The approximation/gap versions of the decision and search problems are defined naturally as

**Definition 1.5** (Decision (approximation)). \( SW_{OPT}(v_A, v_B) \geq C \) or \( SW_{OPT}(v_A, v_B) < \alpha C \)?

**Definition 1.6** (Allocation (approximation)). Return \( S \) such that \( SW(S) \geq \alpha SW(v_A, v_B) \)

For general valuation functions, the problem of outputting the exact solutions of the decision and search problem can be easily shown to be NP-Hard.
1.2 Valuation function types

We quickly also define some restricted class of functions without comment on the motivation.

**Definition 1.7** (Submodular). A valuation function \( f \) is submodular if for every \( X, Y \subseteq [m] \) with \( X \subseteq Y \) and every \( x \in [m] \setminus Y \) we have that
\[
f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y).
\]

Submodular functions are usually defined on graphs, but are good for modeling the concept of “decreasing marginal utilities” as the above definition demonstrates.

**Definition 1.8** (Additive). A valuation function \( f \) is additive if \( f(S) = \sum_{i \in S} v(\{i\}) \). The valuation function is binary additive if \( f \) is additive and \( f(\{i\}) \in \{0, 1\} \) for all items \( i \).

It is useful to think of binary additive as a set, referring to \( f(X) = |X \cap A| = |\{i|i \in A, \in X\}| \) and naturally \( A = \{i|v(\{i\}) = 1\} \).

**Definition 1.9** (General XOS). A valuation function \( f \) is general XOS if \( f \) if there exists additive valuations (which we call clauses) \( \{f_i\}_{i=1}^n \) such that \( f(X) = \max_i f_i(X) \)

**Definition 1.10** (Binary-XOS). A valuation function \( f \) is binary XOS if \( f \) is XOS and all \( f \)’s clauses (or “sets”) are binary additive valuations.

Again, it is simpler to refer to \( f \)’s clauses as sets to make it more natural to talk about unions/intersections/etc. Specifically, \( f(X) = \max_{A_i} |X \cap A_i| \)

1.3 Results of the paper

The following are the results proved in the paper. Note that the key interplay is between theorem 1.7 and 1.8 which provides a strange result – an instance of a setting where the allocation or search problem is provably easier than the decision problem!

**Theorem 1.11.** There exists a randomized, simultaneous protocol with poly\((m)\) communication that obtains a \( 3/4 \)-approximation for the BXOS allocation problem. This is the best possible, as even randomized, interactive protocols require \( 2^{\Omega(m)} \) communication to do better.

This means that one needs polytime to \( 3/4 \)-approximate the allocation problem in the BXOS case

**Theorem 1.12.** There exists a deterministic simultaneous protocol with poly\((m)\) communication that obtains a \( (23/32 - 1/k) \) approximation to the 2-party XOS allocation problem and the 2-party XOS decision problem. It is not known if this approximation ratio is the best possible.

This means that one needs polytime to \( 23/32 \)-approximate the allocation and decision problem in the general XOS case
Theorem 1.13. For all $\epsilon > 0$, any randomized, simultaneous protocol that obtains a $(3/4 - 1/108 + \epsilon)$-approximation for the BXOS decision problem with probability larger than $1/2 + 1/poly(m)$ requires $2^{\Omega(m)}$ communication.

This means that at least exp-time is needed to $3/4$-approximate the decision problem in the BXOS case (notice the complementing Theorem 1.9).

2 Motivation, and a brief history of how we got here

Before we talk about the proofs of the result, it is useful to stop and wonder why such an “artificial” seeming setting is interesting to study. Indeed, simultaneous protocols are interesting in their own right, but there is a rather long line of work that leads one to look for lower bounds for the problem of simultaneous 2-play combinatorial auctions, and it is worth a quick glance to appreciate the work. The main motivation for this problem stems from mechanism design.

Mechanism design deals with looking at truthful protocols and how they compare to each other. A truthful protocol for a combinatorial auction is a protocol which, given that every other player is following the protocol, leaves a player at least or strictly worse off by deviating from the protocol. One can think of this as the “oracles” for the value queries have the option to lie, but given the structure of the protocol, are worse off lying.

2.1 What is the price of truthfulness?

The problem of whether truthful protocols are as powerful as algorithms (an algorithm simply solves the welfare maximization problem given true oracle access to value queries; a protocol doesn’t need players to be truthful) is a central problem in mechanism design. Intuitively, it seems like incentivizing “truthfulness” must have a price. However, surprisingly enough, the VCG Mechanism ([Vic61],[Cla71],[Gro73]) accomplishes this goal of a truthful mechanism. In other words, for very general valuations, the VCG mechanism finds the optimal allocation while incentivizing the players to report their valuations truthful. The welfare maximization problem itself is NP hard even with oracle access, so as expected, the VCG mechanism also takes exponential time.

Now, since the welfare maximization problem itself is NP Hard, it takes at least exp-time to even get a better than $O(\sqrt{m})$ approximation, and this bound was matched by [BKV05]. Interestingly, [LS11] came up with a truthful mechanism with the same approximation guarantee. So far, no gap was found. Truthful protocols can still do as well as algorithms.

In order to find a gap, we restrict ourselves to a class of monotone functions, like sub-modular functions. Here, the best known approximation by algorithms (poly-time oracle valuation queries or poly-time in the description complexity of the valuation functions) can achieve a $(1 - 1/e)$ approximation (refer to [DV12a]). However, [DV12b] showed that truthful mechanisms cannot do better than a $O(m)$ approximation (m is number of items) in poly time (again, poly-time oracle valuation queries or poly-time in the description complexity of the valuation functions). Thus, a separation was found, provably! One has to pay the price for incentivizing truthfulness – indeed, an $O(m)$ approximation is much worse than a constant $(1 - 1/e)$ approximation!
2.2 Seeming contradiction?

However, a seemingly contradictory result was known long before [DV12b] – a “natural looking” mechanism by [DNS06] (the same author!) achieved an $O(\log m)$ approximation, and a subsequent improvement by [Dob16a] improved this mechanism to a $O(\sqrt{\log m})$ (this was for XOS functions which are defined later, but XOS functions are submodular). This seems like a contradiction – while the previous results we just saw said no truthful mechanism can do better than $O(m)$ approximation in polytime, here is one that does a $O(\sqrt{\log m})$ approximation in polytime. What’s the catch?

The catch is that this “natural looking” mechanism is what is called a “posted price” mechanism. Without getting into the specific details of the mechanism, this mechanism essentially offers a price $p_j$ for every item $j$ and lets the bidders pick their favorite subsets. This is different from the value queries, for which the $O(m)$ bound was proved, because computing the best subset for a valuation function (even if its XOS) can itself be an NP-hard problem.

Thus, in some sense, this mechanism was “outsourcing” some of the computation to the bidders themselves.

2.3 Justification for the communication setting and a reduction to simultaneous communication protocols

This brings us to realize that maybe the model of value queries isn’t quite the correct model to prove lower bounds in. For the sake of separating all “natural looking” truthful mechanisms, the lower bounds must be proved in the setting of communication complexity. This makes sense, because every “natural” mechanisms must involve some sort of communication, so lower bounding the communication for truthful protocols would lower bound the complexity of any mechanism.

However, this was thought to be hard for a long – all the known lower bounds for truthful mechanisms were in fact just lower bounds for (not necessarily truthful) communication protocols. This means that the bounds were the powerful bounds that not just rule out truthful mechanisms, but also algorithms. This wasn’t great and most of the progress was stalled until a recent work by [Dob16b]. Again, without getting into the details of the proof, [Dob16b] shows that

\begin{theorem}
any truthful protocol with poly-time communication that achieves an approximation ratio of $\alpha$ with 2 players would imply a simultaneous (not necessarily truthful) protocol for achieving the same approximation ratio $\alpha$
\end{theorem}

Thus the following important corollary

\begin{corollary}
A bound of the type ‘any simultaneous protocol needs super poly-time for a approximation ratio of better than $\alpha$’ would imply that no truthful mechanism can achieve better than an $\alpha$ approximation ratio with polynomial communication."
\end{corollary}

\footnote{Although a lot of this line of work is thanks to Dobzinski, my citations are misleading and a lot of the work leading up to these results is credited to several other people! Please refer to the original [BMW17] paper for a complete treatment of this topic.}
provides a way to bound the communication complexity, and therefore the runtime of truthful mechanisms!

Note: The result that a 23/34 approximation exists for the XOS case is a little surprising, because it’s known that any sketching scheme for XOS valuations that allows for evaluation of value queries to be accurate within a \( o(m) \)-factor requires \( \text{superpoly}(m) \) size. So, paraphrasing the authors of the paper,

“... somehow a \((1/(2 - \epsilon))\)-approximation could be guaranteed with a \( \text{poly}(m) \)-communication simultaneous protocol, it is not because enough information is transmitted to evaluate value queries within any non-trivial error.”

Trivia: Indeed, the authors conjectured at the onset of this work that it was not possible to better than 1/2.

3 Main results and some proofs

We first start off by a quick warmup that will motivate the definitions and ideas of the full proof. Then we will introduce the ideas necessary for the proof and we will sketch a proof for the binary case.

3.1 Warmup

Recall the problem and the setting

- 2 Players, Alice and Bob, and an auctioneer
- Alice and Bob have valuation functions \( v_A, v_B : 2^m \to \mathbb{R}_+ \)
- Valuation functions are Binary XOS (BXOS)
- Another way to think of BXOS is that Alice is given sets \( \{A_i\} \) and her value for the set \( S \) is \( \max_i |A_i \cap S| \)
- Goal is find \( S \) to maximize \( SW(S) = v_A(S) + v_B(\bar{S}) \)

Approach

- Alice and Bob simultaneously announce some information. Auctioneer uses information to allocate \( S, \bar{S} \) to Alice and Bob resp.

Let’s start in a restricted setting where Alice and Bob announce only 1 set.

**Question:** If Alice and Bob could only announce 1 set each, which would they pick?

**Answer:** The largest set!
Thus, we think of the following protocol.

**Protocol I**

- Alice picks her largest clause $A_1$
- Bob picks his largest clause $B_1$
- Both send $A_1, B_1$ to auctioneer, who flips a coin, then gives all the items in $A_1$ to Alice and rest to Bob or vice versa.

**Claim I:** This above protocol gives a 1/2 approximation

**Proof of Claim I:**

\[
|A_{\text{max}}| = \max_i |A_i \cap [m]| = v_A([m]) \geq v_A(S)
\]
\[
|B_{\text{max}}| = \max_i |B_i \cap [m]| = v_B([m]) \geq v_B(\bar{S})
\]
\[
\text{Output} = 1/2(SW(A) + SW(\bar{B}))
\]
\[
\geq 1/2(v_A([m]) + v_B([m]))
\]
\[
\geq 1/2(v_A(S) + v_B(\bar{S}))
\]
\[
= 1/2(SW(v_A, v_B))
\]

This protocol can be easily de-randomized by giving all the items to the player with the larger reported clause.

New question – what if Alice and Bob could both report 2 clauses? What would be their optimal choice. Maybe it makes sense to report the 2 clauses with the largest union. Intuitively, this is covering most of the set $[m]$ that the player wants. At the same time, somehow the players also want to not represent the each element too much ie. not have the intersection of these clauses to be too large. So to understand the balance between covering a large section of the $[m]$ and minimizing the intersection, we look at the restricted case where Alice and Bob can each send 3 clauses.

**Protocol II**

- Alice picks $A_i$ which maximizes $|A_i|$ and $A_{j1}, A_{j2}$ which maximize $SW(A_{j1}, A_{j2}) = |A_{j1} \cup A_{j2}|$. Let the clauses picked be $A_1, A_2, A_3$.
- Alice sends one of them randomly.
Auctioneer allocates all items in sent clause $A_i$ to Alice and the rest to Bob

Claim II: The above protocol gives a $2/3$ approximation to the allocation problem

Observation: If Alice’s valuation function is just $v_A(X) = |X \cap A^*|$ and Bob has a BXOS valuation $v_B$ then $SW(A^*) = OPT(v_A, v_B)$.

Proof Sketch of observation: Basically, Alice only wants items in $X$, so giving items from $X$ to Bob cannot increase the welfare.

In other words, optimal welfare can be achieved by just giving Alice everything she wants

Proof of Claim II:
Recall that

- $A_1 = \max |A_i|$. $A_2, A_3 = \arg \max |A_i \cup A_j|
- Auctioneer allocates all items in randomly sent clause $A_i$ to Alice and the rest to Bob

Notation

- $S, \bar{S}$ is any real OPT partition for $v_A, v_B$. $A_{OPT}, B_{OPT}$ are the participating clauses.
- $OPT(v_A, v_B) = |A_{OPT} \cup B_{OPT}| = SW(A_{OPT}) = SW(B_{OPT}) = SW(S)$
- $OPT(v_A, v_A) = |A_2 \cup A_3| = |A_2| + |\bar{A}_2 \cap A_3| \geq |S \cap A_{OPT}| + |\bar{S} \cap A_1|$

Thus, the algorithm gets $1/3$rd of $\sum_{i=1}^{3} SW(A_i, v_B)$

$$\begin{align*}
&= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3)) \\
&\geq (|A_1 \cap S| + v_B(\bar{S}) + |A_2| + v_B(\bar{A}_2) + |\bar{A}_2 \cap A_3| + v_B(A_2)) \\
&\geq (|A_1 \cap S| + v_B(\bar{S}) + |S \cap A_{OPT}| + v_B(\bar{A}_2) + |\bar{S} \cap A_1| + v_B(A_2)) \\
&\geq (|A_1| + |B_{OPT}| + v_B(\bar{S}) + |S \cap A_{OPT}|) \\
&= (|A_1| + |B_{OPT}| + |B_{OPT} \cap \bar{S}| + |S \cap A_{OPT}|) \\
&\geq (|A_{OPT} \cap B_{OPT}| + |B_{OPT}| + |B_{OPT} \cap \bar{S}| + |S \cap A_{OPT}|) \\
&= SW(\bar{B}_{OPT}) + SW(S) \\
&= 2SW_{OPT}(v_A, v_B)
\end{align*}$$
Here the key takeaways are

- Notice that although the protocol guarantees a 2/3 approximation, the auctioneer has no idea what the actual welfare achieved is.
- This is because the auctioneer knows nothing about Bob’s valuation function at all!
- So this protocol I gives a 2/3 approximation to the allocation problem. With 1 more round, Bob could send his fav clause and that would also solve the decision problem.

**Question:** What happens if we take Bobs input too in a similar manner? Can we get a similar allocation guarantee while knowing the social welfare that it leads to?

The following protocol does exactly that

**Protocol III**

- Alice picks $A_i$ which maximizes $|A_i|$ and $A_{j1}, A_{j2}$ which maximize $SW(A_{j1}, A_{j2}) (= |A_{j1} \cup A_{j2}|)$. Let the clauses picked be $A_1, A_2, A_3$.
- Bob does the same. Let the clauses picked be $B_1, B_2, B_3$.
- Both send everything to auctioneer
- Auctioneer picks max $SW(A_i, B_j)(= \max |A_i \cup B_j|)$ of whatever they received and divide items accordingly

We provide the following claim without proof (refer to [BMW17] for complete proofs)

**Claim:** Protocol III is a 3/5 approximation for the decision and allocation problem

**Observation:** Auctioneer knows the value of the proposed allocation because they calculated it. Approximation guarantee takes care of the rest.

**Note:** Turns out, we can remove the “binary” assumption and just use max $SW(X,Y)$ and still get the same guarantee for general XOS.

Thus we see two main themes here – somehow reporting sets with large unions is important, but its also important to not have elements represented in too main reported clauses (intersections should be small too). Perhaps its not clear immediately how to generalize these definitions, but the follow section provides the required tools to generalize this intuition.
Another theme we see is that when we only used Alice’s input, and randomly selected an outcome, it in fact gave a better approximation guarantee for the allocation than deterministically picking an outcome who’s social welfare we knew \((2/3 > 3/5)\). This general pattern of Alice-only protocols having better approximation guarantees for the allocation problem continues in the following sections.

### 3.2 \(k, \alpha\)-summaries

How does one generalize this notion of “largest” clause as well as “clauses that cover the largest section”. We observe the following.

- Want to pick large clauses
- Want to pick clauses that don’t have too much overlap
- Not clear which is more important. Somehow need a balance..

Suppose Alice were to send \(k\) clauses \(\{A_i\}_{i=1}^k\). Let \(x_i = \frac{1}{k} \sum_{j=1}^k 1_{i \in A_j}\). Intuitively, \(x_i\) captures how much importance the \(k\) clauses give the element \(i\).

More observations follow:

- Maximize \(\sum x_i\) to get large unions. (Sanity check: what is the answer when \(k = 1\)?)
- Minimize \(\sum x_i^2\) to get small intersections (not immediately obvious that this is the right way to do things.)
- Somehow we need a balance. Maybe maximize \(\sum x_i - \alpha x_i^2\)?

**Definition 3.1.** A \((k, \alpha)\)-summary of a BXOS valuation \(\{A_i\}_{i=1}^l\) is the set of \(k\) clauses defined as

\[
(A_1, A_2, \ldots A_k) = \arg \max_{(C_1, C_2, \ldots C_k \in \mathcal{V}_A)} \sum x_i - \alpha x_i^2
\]

and \(x_i = \frac{1}{k} \sum_{j=1}^k 1_{i \in C_j}\) is as defined above.

For a sanity check we can see that

- A \((1,1/2)\)-summary selects the largest clause. This was our first strategy!
- A \((2, 2/3)\)-summary selects the 2 clauses with the largest union. This was our second strategy!

The simple proofs for these sanity checks are skipped (but fun to try out!)
4 Summary of results

Thus, not all our efforts can culminate into finally understanding the results of the paper. Recall that:

- for 2/3-BXOS Alice sent a random clause decided in a “summary-like manner” and auctioneer assigned all that Alice wanted to her.
- for 3/5-XOS Bob and Alice both sent all the clauses decided in a “summary-like manner” and auctioneer assigned the best allocation.
- for 3/5-XOS, the approximation didn’t make use of binary assumption.
- for 3/5-XOS auctioneer knows the welfare of proposed allocation while for the 2/3-BXOS, auctioneer did not know the value.

The results in the paper follow a similar flavor. The proofs are routine and are thus skipped. The reader can refer to the original paper for complete proofs!

**Theorem 4.1.** The following protocols achieve the following guarantees:

<table>
<thead>
<tr>
<th>Alice’s summary</th>
<th>Bob’s summary</th>
<th>Wrap-up</th>
<th>Approximation</th>
<th>Problem</th>
<th>Valuations</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k, 1/2)</td>
<td>⊥</td>
<td>Alice-Only</td>
<td>3/4 − 1/k</td>
<td>Allocation</td>
<td>BXOS</td>
</tr>
<tr>
<td>(k, 1/3)</td>
<td>(k, 1/3)</td>
<td>Best Known Allocation</td>
<td>23/32 − 1/k</td>
<td>Allocation</td>
<td>XOS</td>
</tr>
<tr>
<td>(k, 1/3)</td>
<td>(k, 1/3)</td>
<td>Best Known Decision</td>
<td>23/32 − 1/k</td>
<td>Decision</td>
<td>XOS</td>
</tr>
</tbody>
</table>

As usual, we follow up with some quick observations:

- The 3/4-approximation guaranteed by the protocol in the first row is tight: randomized, interactive protocols require exponential communication to beat a 3/4-approximation.
- The second and third protocols also work for general XOS (naturally extend definition of (k, α)-summary)
- Still open whether it is possible to beat 23/32 with a deterministic protocol for the allocation problem!
- 23/32 is optimal for any protocol using the Best Known Allocation after Alice and Bob each report a (k, α)-summary (skipped proof)
- No simultaneous, randomized protocol can do better than (3/4-1/108) approximation with less than exp(m) communication (WHP)

4.1 Lower bounds for communication

Finally, we sketch a lower bound for the communication. Recall the theorem

**Theorem 4.1** (Also Thm. 1.13). For all ε > 0, any randomized, simultaneous protocol that
obtains a \((3/4 - 1/108 + \epsilon)\)-approximation for the BXOS decision problem with probability larger than \(1/2 + 1/poly(m)\) requires \(2^{\Omega(m)}\) communication.

So we want to show that to do as good as 3/4 for the decision problem, one needs exponential communication at least. The idea is to create exponentially many sets that all “look very similar” and give them to Alice, and do the same for Bob. Amongst these sets, hide 2 sets that also look similar, but are complements of each other. Then, the problem of finding a 3/4 approximation reduces to the problem of finding these hidden sets or the auctioneer using randomness to find these hidden sets. A bit more formally ....

- First sample a \(S\) and a \(T\) uniformly at random from all sets such that \(|S \cap T| = m/6\) and \(|S| = |T| = m/6\)
- Sample exp many sets \(A_i\) such that \(|A_i \cap S| = m/3\) and \(|A_i \cap \bar{S}| = m/6\) (\(|A_i| = m/2\))
- Sample exp many sets \(B_i\) such that \(|B_i \cap T| = m/3\) and \(|B_i \cap \bar{T}| = m/6\) (\(|B_i| = m/2\))
- Pick a set \(A^*\) such that \(|A^* \cap S \cap T| = m/6\), \(|A^* \cap S \cap \bar{T}| = m/6\) and \(|S \cap \bar{S} \cap T| = 0\). Basically, this set satisfies the conditions of the sampling in step 2. Let \(B^* = \bar{A}^*\). \(B^*\) satisfies similar conditions.
- Give Alice \(A_i\)s and Bob \(B_i\)s
- Flip a coin – if heads then hide \(A^*\) in Alice’s valuation and \(B^*\) in Bob’s valuations, else do nothing

Case 1: The max welfare is \(m\)
Case 2: The max welfare is \(3/4 - 1/108m\) (constant from sampling and concentration bounds)

---

**Very high level proof of Theorem 4.1:** To be able to do better than 3/4 at decision algorithm needs to tell Case 1 and Case 2 apart.
But the special set \(A^*\) looks exactly like the other exponential \(A_i\) to Alice (same for Bob).
So they need to either send exponentially large information, or the auctioneer needs to get lucky and that happens with exponentially small probability. One can prove this with a bound using mutual information! between coin toss and information sent to auctioneer by Bob and Alice!

### 5 Recap

Thus, to quickly recap, we saw the following things [Dob16b] reduction says that a lower bound for simultaneous communication complexity of welfare maximization \(\Rightarrow\) lower bound on communication for truthful welfare maximization protocol.

Goal: Find lower bound for simultaneous communication complexity of XOS/BXOS valuations.
5.1 Main Results

- 3/4 Approximation for the BXOS problem (sketched)
- 23/32 Approximation for the XOS problem (similar to sketch)
- Lower bound for the decision problem (sketched)
- 23/32 upper bound for approximation ratio of algorithms that does 23/32 for XOS problem (counter example is Binary!) (not shown in lecture)

Additional results that were not mentioned in lecture but are written for the sake of completeness:

- Interesting new type of truthful protocol for BXOS that gets 3/4 ratio
- Truthful protocol for BXOS and more than 2 parties that gets 1/2 ratio

However, there is some bad news.

In case you didn’t notice, no lower bound of the type that was actually needed was proved :( . Dobzinski’s reduction requires a lower bound for the “allocation problem” but the lower bound proved was for the “decision problem”. The lower bound for the decision version of BXOS is a lower bound on the fully general communication complexity of the problem, so is rather trivial. For the general XOS case, the authors showed a result which had an approximation of 23/32, but this doesn’t rule out a better approximation! Infact, the authors also proved that their method of k-summarizes, has its limitation and cannot do better than 23/32. Unfortunately, this means we still don’t know that it’s not better to do better than a 23/32 for general XOS case. But this is an open problem!

5.2 Open Problems

So, to end these notes, we list some open problems in this direction.

- Deterministic 3/4 approximation for BXOS (interesting because would them hope to imply det. protocol)
- < 3/4 approximation lower bound for deterministic case? (Lower bound from paper is for this specific type) (interesting because can use Dobzinski’s reduction to imply lower bound for truthful auctions )
- > 23/32 approximation for general XOS? (interesting because then allocation easier than decision in full generality!)
- < 3/4 lower bound for general XOS (3/4 is best known). (Again, can use Dob.’s reductions)
I want to thank Prof. Ran Raz for a very engaging, clear, and illuminating course (and also for the opportunity to present this paper). Also, thanks to the authors Prof. Weinberg, and Jieming for talking at length about the contents of their papers. Lastly, thanks to all the students in the course for listening!

References


