On simultaneous two player combinatorial auctions

Braverman, Mao, Weinberg

December 6, 2017
Table of Contents

1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
   • Lower bound

4 Recap
Table of Contents

1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
   • Lower bound

4 Recap
Table of Contents

1 Problem
   - Setting
   - Results

2 Some History
   - Mechanism Design 101
   - The price of truthfulness
   - Strong lower bounds
   - Something strange
   - A reduction to simultaneous protocols

3 Approximation results
   - Warmup
   - k-summary
   - Lower bound

4 Recap
The Problem

Input

Two players Alice and Bob each with a "valuation function" $v_a$:

$$v_a : 2^{[m]} \rightarrow \mathbb{R}^+$$

Bob has $v_b$:

$$v_b : 2^{[m]} \rightarrow \mathbb{R}^+$$

(Also, implicitly, the set $[m]$ of "items" – motivation coming soon)

Output

Partition of $[m]$ into $S, \bar{S}$ such that

$$v_a(S) + v_b(\bar{S})$$

is maximized

Definition

$$SW(S) = v_A(S) + v_B(\bar{S})$$

is the social welfare

$$OPT(v_A, v_B) = \max_S SW(S)$$

is the optimal social welfare.
The Problem

Input

- Two players Alice and Bob each with a "valuation function" \( v \)

Output

Partition of \( \{m\} \) into \( S, \bar{S} \) such that \( v_a(S) + v_b(\bar{S}) \) is maximized

Definition

\[ SW(S) = v_A(S) + v_B(\bar{S}) \]

is the social welfare

\[ \text{OPT}(v_A, v_B) = \max_S SW(S) \]

is the optimal social welfare.
The Problem

Input

- Two players Alice and Bob each with a "valuation function" $v$
- Alice has $v_a : 2^m \rightarrow \mathbb{R}_+$
- Bob has $v_b : 2^m \rightarrow \mathbb{R}_+$
The Problem

Input

- Two players Alice and Bob each with a "valuation function" $\nu$
  - Alice has $\nu_a : 2^m \to \mathbb{R}_+$
  - Bob has $\nu_b : 2^m \to \mathbb{R}_+$

(Also, implicitly, the set $[m]$ of "items" – motivation coming soon)
The Problem

Input
- Two players Alice and Bob each with a "valuation function" $\nu$
- Alice has $\nu_a : 2^m \to \mathbb{R}_+$
- Bob has $\nu_b : 2^m \to \mathbb{R}_+$

(Also, implicitly, the set $[m]$ of "items" – motivation coming soon)

Output
The Problem

Input

- Two players Alice and Bob each with a "valuation function" \( v \)
- Alice has \( v_a : 2^m \rightarrow \mathbb{R}_+ \)
- Bob has \( v_b : 2^m \rightarrow \mathbb{R}_+ \)

(Also, implicitly, the set \([m]\) of "items" — motivation coming soon)

Output

- Partition of \([m]\) into \( S, \bar{S} \) such that \( v_a(S) + v_b(\bar{S}) \) is maximized

Definition

\( SW(S) = v_A(S) + v_B(\bar{S}) \) is the social welfare of the allocation \( S \)

\( OPT(v_A, v_B) = \max_S SW(S) \) is the optimal social welfare.
Formally..

Decision Problem (is it possible to get at least C welfare)

∃ \mathcal{S} \subseteq \mathcal{Y} \text{ such that } \mathcal{S} \subseteq \mathcal{Y} \Rightarrow C \iff \text{OPT}(v_A, v_B) > C

Allocation Problem (maximum welfare allocation)

Return \mathcal{S} = \text{argmax}_X SW(X)

Sanity check: \mathcal{S} \subseteq \mathcal{Y} = \text{OPT}(v_A, v_B)

However, usually we don't hope for exact answers so we work approximation guarantees and algorithms.
Formally,

### Decision Problem (is it possible to get at least C welfare)

\[
\exists S \subseteq [m] \text{ such that } SW(S) > C \iff OPT(v_A, v_B) > C
\]

### Allocation Problem (maximum welfare allocation)

Return \( S = \arg\max X SW(X) \)

Sanity check:

\[ SW(S) = OPT(v_A, v_B) \]

However, usually we don't hope for exact answers so we work approximation guarantees and algorithms.
Formally..

**Decision Problem (is it possible to get at least C welfare)\)**

\[ \exists S \subseteq [m] \text{ such that } SW(S) > C \iff OPT(v_A, v_B) > C \]

**Allocation Problem (maximum welfare allocation)**

\[ \text{Sanity check: } SW(S) = OPT(v_A, v_B) \]

However, usually we don't hope for exact answers so we work approximation guarantees and algorithms.
Formally..

Decision Problem (is it possible to get at least C welfare)
\[ \exists S \subseteq [m] \text{ such that } SW(S) > C \iff OPT(v_A, v_B) > C \]

Allocation Problem (maximum welfare allocation)
- Return \( S = \arg \max_X SW(X) \)
Formally..

**Decision Problem (is it possible to get at least C welfare)**

\[ \exists S \subseteq [m] \text{ such that } SW(S) > C \iff OPT(v_A, v_B) > C \]

**Allocation Problem (maximum welfare allocation)**

- Return \( S = \arg\max_X SW(X) \)

Sanity check: \( SW(S) = OPT(v_A, v_B) \)

However, usually we don’t hope for exact answers so we work approximation guarantees and algorithms.
Formally and approximately

Decision Problem (approximation version)
Formally and approximately

Decision Problem (approximation version)

\[ \exists S \subseteq [m] \text{ such that } SW(S) \geq C \text{? or } \forall S, SW(S) < \alpha C \]
Formally and approximately

<table>
<thead>
<tr>
<th>Decision Problem (approximation version)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\exists S \subseteq [m]$ such that $SW(S) \geq C$? or $\forall S, SW(S) &lt; \alpha C$</td>
</tr>
<tr>
<td>$\iff$ $OPT(v_A, v_B) \geq C$ or $OPT(v_A, v_B) &lt; \alpha C$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Allocation Problem (approximation version)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return $S$ such that $SW(S) \geq \alpha OPT(v_A, v_B)$</td>
</tr>
</tbody>
</table>

Braverman, Mao, Weinberg
On simultaneous two player combinatorial auctions
December 6, 2017
Formally and approximately

<table>
<thead>
<tr>
<th>Decision Problem (approximation version)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \exists S \subseteq [m] \text{ such that } SW(S) \geq C ) or ( \forall S, SW(S) &lt; \alpha C )</td>
</tr>
<tr>
<td>( \iff ) ( \text{OPT}(v_A, v_B) \geq C ) or ( \text{OPT}(v_A, v_B) &lt; \alpha C )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Allocation Problem (approximation version)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return ( S ) such that ( SW(S) \geq \alpha \text{OPT}(v_A, v_B) )</td>
</tr>
</tbody>
</table>
Simultaneous setting

Definition (Simultaneous CC)

- 2 players and a Viewer
- Player 1 gets $x$, Player 2 gets $y$
- Players simultaneously output information $I_1$, $I_2$
- Using $I_1$, $I_2$, Viewer computes $f(x, y)$ (in some settings without bounds on computational power)

The catch is that Players must decide what information to announce without knowing anything about Players’ input. In some sense they must "summarize" their input.
Generally we don’t separate decision problems ($\exists OPT > C$) from search problems ($OPT$ which is max). This is because we know that the decision problem reduces to the search problem.
Generally we don’t separate decision problems ($\exists OPT > C$) from search problems ($OPT$ which is max). This is because we know that the decision problem reduces to the search problem.

However, we consider the framework of *simultaneous* communication complexity.
Table of Contents

1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
   • Lower bound

4 Recap
Results (presented)

For 2 players Alice and Bob, in the simultaneous setting

Result: 3/4 approximation for allocation problem

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

Result: "3/4" approximation for decision problem is hard

Any simultaneous protocol for 3/4 - 1/108 approximation for binary XOS valuations* requires exp(m) communication.

*binary XOS will be defined soon
Results (presented)

For 2 players Alice and Bob, in the simultaneous setting

Result: 3/4 approximation for allocation problem

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*.

Result: “3/4” approximation for decision problem is hard.

Any simultaneous protocol for 3/4 - 1/108 approximation for binary XOS valuations* requires exp(m) communication.

*binary XOS will be defined soon.
Results (presented)

For 2 players Alice and Bob, in the simultaneous setting

**Result: 3/4 approximation for allocation problem**

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

*binary XOS will be defined soon
Results (presented)

For 2 players Alice and Bob, in the simultaneous setting

Result: 3/4 approximation for *allocation problem*
Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

Result: "3/4" approximation for *decision problem* is hard

*binary XOS will be defined soon
Results (presented)

For 2 players Alice and Bob, in the simultaneous setting

Result: 3/4 approximation for allocation problem

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

Result: "3/4" approximation for decision problem is hard

Any simultaneous protocol for 3/4 -1/108 approximation for binary XOS valuations* requires exp(m) communication.
Results (presented)

For 2 players Alice and Bob, in the simultaneous setting

**Result: 3/4 approximation for allocation problem**

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

**Result: "3/4" approximation for decision problem is hard**

Any simultaneous protocol for 3/4 -1/108 approximation for binary XOS valuations* requires exp(m) communication.

*binary XOS will be defined soon
Results (presented)

Main takeaway for simultaneous setting:

Result 1.1: 3/4 approximation for allocation problem is easy.

Result 2.1: "3/4" approximation for decision problem is hard.
Results (presented)

Main take away
For simultaneous setting

Result 1.1: 3/4 approximation for *allocation* problem is easy
Main take away
For simultaneous setting

Result 1.1: 3/4 approximation for allocation problem is easy

Result 2.1: "3/4" approximation for decision problem is hard
Main take away
For simultaneous setting with binary XOS valuations

Result 1.1: $\frac{3}{4}$ approximation for allocation problem is easy

Result 2.1: "$\frac{3}{4}$" approximation for decision problem is hard

For general interactive setting

Result 1.2: $\frac{3}{4}$ approximation for allocation problem is tight even for randomized interactive* protocols and general XOS

*interactive means >1 round protocols
Results (not presented)

Main take away
For simultaneous setting with binary XOS valuations

Result 1.1: 3/4 approximation for allocation problem is easy

Result 2.1: "3/4" approximation for decision problem is hard

For general interactive setting

Result 1.2: 3/4 approximation for allocation problem is tight
even for randomized interactive* protocols and general XOS

Result 2.2: 3/4 approximation for decision problem is achievable by a deterministic interactive* protocols
Results (not presented)

Main take away
For simultaneous setting with binary XOS valuations

Result 1.1: 3/4 approximation for allocation problem is easy

Result 2.1: "3/4" approximation for decision problem is hard

For general interactive setting

Result 1.2: 3/4 approximation for allocation problem is tight even for randomized interactive* protocols and general XOS

Result 2.2: 3/4 approximation for decision problem is achievable by a deterministic interactive* protocols

*interactive means >1 round protocols
Results from a high level

For simultaneous setting

3/4 approximation for allocation problem is easy

"3/4" approximation for decision problem is hard

For general interactive setting

Can’t do better than 3/4 for allocation problem even with more rounds
Results from a high level

For simultaneous setting

3/4 approximation for allocation problem is easy

"3/4" approximation for decision problem is hard

For general interactive setting

Can’t do better than 3/4 for allocation problem even with more rounds

But can get to 3/4 with just 1 more round
Results from a high level

For simultaneous setting

3/4 approximation for allocation problem is easy

"3/4" approximation for decision problem is hard

For general interactive setting

Can’t do better than 3/4 for allocation problem even with more rounds

But can get to 3/4 with just 1 more round
Results (not presented) but maybe sketched if time

Natural extensions from binary to general XOS

**Result:** $\frac{23}{32}$ approximation for allocation problem

**Result:** Randomization needed for protocols of this type

No deterministic simultaneous protocol of the a “certain” type** can achieve better than $\frac{23}{32}$ approximation for either problem.

$\frac{23}{32} = 3/4 - 1/32$

**“certain type” also defined soon**
Results (not presented) but maybe sketched if time

Natural extensions from binary to general XOS

Result: 23/32* approximation for allocation problem

Exists a deterministic simultaneous protocol with poly(m) communication for a 23/32 approximation for general XOS valuations

*23/32 = 3/4 - 1/32

**"certain type" also defined soon
Results (not presented) but maybe sketched if time

Natural extensions from binary to general XOS

Result: 23/32* approximation for allocation problem

Exists a deterministic simultaneous protocol with poly(m) communication for a 23/32 approximation for general XOS valuations

Result: Randomization needed for protocols of this type

No deterministic simultaneous protocol of the a "certain" type** can achieve better than 23/32 approximation for either problem.
Results (not presented) but maybe sketched if time

Natural extensions from binary to general XOS

**Result:** 23/32* approximation for *allocation problem*

Exists a **deterministic simultaneous** protocol with poly(m) communication for a 23/32 approximation for general XOS valuations

**Result:** Randomization needed for protocols of this type

No **deterministic simultaneous** protocol of the a "certain" type** can achieve better than 23/32 approximation for either problem.

*23/32 = 3/4 - 1/32

**"certain type" also defined soon
Results (not presented) from a high level

Natural extensions from binary to general XOS

\[ \frac{23}{32} \] approximation for allocation problem

Can naturally extend technique to general XOS to get \( \frac{23}{32} \). For the extension, deterministic suffices.

Result: Randomization needed for protocols of this type to work better than \( \frac{23}{32} \)

No deterministic simultaneous protocol of this type can do better than \( \frac{23}{32} \) (even for binary case)

(Remember that the \( \frac{3}{4} \) for the binary XOS case was a “randomized” protocol)

\[ \frac{23}{32} = \frac{3}{4} - \frac{1}{32} \]

**“certain type” also defined soon**
Results (not presented) from a high level

Natural extensions from binary to general XOS

**Result: 23/32**\(^*\) approximation for allocation problem

Can naturally extend technique to general XOS to get 23/32. For the extension, deterministic suffices.

\(^*\)23/32 = 3/4 - 1/32

**“certain type” also defined soon**
Natural extensions from binary to general XOS

**Result:** $23/32^*$ approximation for **allocation problem**

Can naturally extend technique to general XOS to get $23/32$. For the extension, deterministic suffices.

**Result:** Randomization needed for protocols of this type to work better than $23/32$

No **deterministic simultaneous** protocol of this type can do better than $23/32$ (even for binary case)
Results (not presented) from a high level

Natural extensions from binary to general XOS

Result: 23/32* approximation for allocation problem

Can naturally extend technique to general XOS to get 23/32. For the extension, deterministic suffices.

Result: Randomization needed for protocols of this type to work better than 23/32

No deterministic simultaneous protocol of this type can do better than 23/32 (even for binary case)

(Remember that the 3/4 for the binary XOS case was a "randomized" protocol)

*23/32 = 3/4 - 1/32

**"certain type" also defined soon
Table of Contents

1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
   • Lower bound

4 Recap
How did we get here?
How did we get here?

Problem is interesting in its own right.
How did we get here?

Problem is interesting in its own right.

Results are interesting because it’s the first of its kind where "decision" is harder than "allocation/search"
How did we get here?

Problem is interesting in its own right.

Results are interesting because it’s the first of its kind where "decision" is harder than "allocation/search"

But the setting seems a little artificial. Where did the motivation come from?
1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
   • Lower bound

4 Recap
Combinatorial Auctions - Model

Each bidder has valuation function \( v_i : \{m\} \rightarrow \mathbb{R}^+ \). Bidders participate in a protocol.* Auctioneer divides \( m \) items amongst the \( n \) bidders and charges each bidder \( p_i \).

* Protocol and algorithm have subtle differences.

Braverman, Mao, Weinberg

On simultaneous two player combinatorial auction

December 6, 2017
Combinatorial Auctions - Model

- $n$ players, $m$ items, 1 auctioneer

Each bidder has valuation function $v_i: \{1, \ldots, m\} \to \mathbb{R}_+$.
Combinatorial Auctions - Model

- $n$ players, $m$ items, 1 auctioneer
- Each bidder has valuation function $v : 2^m \to \mathbb{R}_+$

Braverman, Mao, Weinberg
On simultaneous two player combinatorial auctions
December 6, 2017 20 / 70
Combinatorial Auctions - Model

- n players, m items, 1 auctioneer
- Each bidder has valuation function $v : 2^m \rightarrow \mathbb{R}_+$
- Bidders participate in a protocol.*
Combinatorial Auctions - Model

- n players, m items, 1 auctioneer
- Each bidder has valuation function $v : 2^m \rightarrow \mathbb{R}_+$
- Bidders participate in a protocol*.
- Auctioneer divides m items amongst the n bidders and charges each bidder $p_i$
Combinatorial Auctions - Model

- n players, m items, 1 auctioneer
- Each bidder has valuation function $v : 2^m \rightarrow \mathbb{R}_+$
- Bidders participate in a protocol*.
- Auctioneer divides m items amongst the n bidders and charges each bidder $p_i$

*protocol and algorithm have subtle difference
Combinatorial Auctions - Goal

Divide items so that "welfare" is maximized

Divide items so welfare achieved is at least $\alpha \cdot \text{OPT}$ (approximation)

Incentivize truthful behavior amongst players

Braverman, Mao, Weinberg

On simultaneous two player combinatorial auction
Combinatorial Auctions - Goal

- Divide items so that "welfare" is maximized

\[ \text{Divide items so welfare achieved is at least } \alpha \text{OPT} \text{ (approximation)} \]

- Incentivize truthful behavior amongst players

(Braverman, Mao, Weinberg)
Combinatorial Auctions - Goal

- Divide items so that "welfare" is maximized
- Divide items so welfare achieved is at least $\alpha OPT$ (approximation)
Combinatorial Auctions - Goal

- Divide items so that "welfare" is maximized
- Divide items so welfare achieved is at least $\alpha \cdot \text{OPT}$ (approximation)
- Incentivize truthful behavior amongst players
Combinatorial Auctions - Goal

- Divide items so that "welfare" is maximized
- Divide items so welfare achieved is at least $\alpha OPT$ (approximation)
- Incentivize truthful behavior amongst players

Protocol and algorithm have subtle difference
Protocols vs Algorithms

Protocols

Sets a layout for what is to be done. Example: First the auctioneer will ask players for how much they value an item. The players will all respond with a bid (may or may not be able to see others bids, and can lie). Auctioneer awards item to player with largest bid and charges them the price of 2nd highest bid. Protocols allow for a rich strategy space – one can choose to lie about a valuation in order to better ones payoff given the knowledge of how the protocol will work.

Algorithms

The players are oracles. They answer queries of the sort $v_i(X)$ – how much do you value a subset $X$. Oracles don't lie.
Protocols vs Algorithms

Protocols
- Sets a layout for what is to be done.
Protocols vs Algorithms

Protocols

- Sets a layout for what is to be done.

  Example: First the auctioneer will ask players for how much they value an item. The players will all respond with a bid (may or may not be able to see others bids, and can lie). Auctioneer awards item to player with largest bid and charges them the price of 2nd highest bid.
Protocols vs Algorithms

Protocols

- Sets a layout for what is to be done.
  Example: First the auctioneer will ask players for how much they value an item. The players will all respond with a bid (may or may not be able to see others bids, and can lie). Auctioneer awards item to player with largest bid and charges them the price of 2nd highest bid.

- Protocols allow for a rich strategy space – one can choose to lie about a valuation in order to better ones payoff given the knowledge of how the protocol will work
Protocols vs Algorithms

Protocols

- Sets a layout for what is to be done.
  Example: First the auctioneer will ask players for how much they value an item. The players will all respond with a bid (may or may not be able to see others' bids, and can lie). Auctioneer awards item to player with largest bid and charges them the price of 2nd highest bid.

- Protocols allow for a rich strategy space – one can choose to lie about a valuation in order to better one's payoff given the knowledge of how the protocol will work.

Algorithms
Protocols vs Algorithms

Protocols
- Sets a layout for what is to be done.
  Example: First the auctioneer will ask players for how much they value an item. The players will all respond with a bid (may or may not be able to see others bids, and can lie). Auctioneer awards item to player with largest bid and charges them the price of 2nd highest bid.
- Protocols allow for a rich strategy space – one can choose to lie about a valuation in order to better ones payoff given the knowledge of how the protocol will work

Algorithms
- The players are oracles. They answer queries of the sort $v_i(X)$ – how much do you value a subset $X$. 
Protocols vs Algorithms

Protocols

- Sets a layout for what is to be done.
  Example: First the auctioneer will ask players for how much they value an item. The players will all respond with a bid (may or may not be able to see others bids, and can lie). Auctioneer awards item to player with largest bid and charges them the price of 2nd highest bid.
- Protocols allow for a rich strategy space – one can choose to lie about a valuation in order to better ones payoff given the knowledge of how the protocol will work

Algorithms

- The players are oracles. They answer queries of the sort $v_i(X)$ – how much do you value a subset $X$.
- Oracles don’t lie
Table of Contents

1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
   • Lower bound

4 Recap
Goal - truthfulness

Given that every other player follows the protocol truthfully, it is better for you to also follow protocol.

We know that the algorithm version of the problem is NP-Hard.

Question: Does there exist a "truthful" protocol? How hard is it?

Yes!

Braverman, Mao, Weinberg

On simultaneous two player combinatorial auctions

December 6, 2017
"Truthful" protocol
"Truthful" protocol

- Given that every other player follows the protocol truthfully, it is better for you to also follow the protocol.
"Truthful" protocol

- Given that every other player follows the protocol truthfully, it is better for you to also follow protocol.

We know that the algorithm version of the problem is NP-Hard.

Question: Does there exist a "truthful" protocol? How hard is it?
"Truthful" protocol

- Given that every other player follows the protocol truthfully, it is better for you to also follow protocol.

We know that the algorithm version of the problem is NP-Hard.

Question: Does there exist a "truthful" protocol? How hard is it?

Yes!
Folklore

“Truthful” protocol

VCG - Mechanism that incentivizes player to report their true valuations for each outcome

Sketch: Auction charges the players the “cost of their participation. Roughly that is “max welfare without player” - “welfare of the others in current allocation”

Provably truthful

Braverman, Mao, Weinberg

On simultaneous two player combinatorial auction
"Truthful" protocol

- \([VCG61, 71, 73]\) VCG - Mechanism that incentivizes player to report their true valuations for each outcome
"Truthful" protocol

- [VCG61, 71, 73] VCG - Mechanism that incentivizes player to report their true valuations for each outcome

Sketch: Auction charges the players the "cost of their participation. Roughly that is "max welfare without player" - "welfare of the others in current allocation"
"Truthful" protocol

- [VCG61, 71, 73] VCG - Mechanism that incentivizes player to report their true valuations for each outcome

- Sketch: Auction charges the players the "cost of their participation. Roughly that is "max welfare without player" - "welfare of the others in current allocation"

- Provably truthful
What about, say, monotone functions? Poly-time algorithms?

[BKV05] \[\exp(m)\] communication needed to achieve better than a \(\sqrt{m}\) approximation

[LS05] Poly-time algorithm to match this \(\sqrt{m}\) guarantee CITE.

What about protocols? Interestingly enough, there a poly-time protocol with the same guarantee was also discovered later CITE.

Braverman, Mao, Weinberg
On simultaneous two player combinatorial auctions
December 6, 2017 26 / 70
Approximation algorithms and protocols

What about, say, monotone functions? Poly-time algorithms?

- \([BKV05]\) \(\exp(m)\) communication needed to achieve better than a \(\sqrt{m}\) approximation
Approximation algorithms and protocols

What about, say, monotone functions?
Poly-time algorithms?

- [BKV05] $exp(m)$ communication needed to achieve better than a $\sqrt{m}$ approximation
- [LS05] Poly-time algorithm to match this $\sqrt{m}$ guarantee CITE.
Approximation algorithms and protocols

What about, say, monotone functions?
Poly-time algorithms?

- [BKV05] $\exp(m)$ communication needed to achieve better than a $\sqrt{m}$ approximation
- [LS05] Poly-time algorithm to match this $\sqrt{m}$ guarantee CITE.

What about protocols?
Approximation algorithms and protocols

What about, say, monotone functions?

Poly-time algorithms?

- \([BKV05]\) \(\exp(m)\) communication needed to achieve better than a \(\sqrt{m}\) approximation
- \([LS05]\) Poly-time algorithm to match this \(\sqrt{m}\) guarantee CITE.

What about protocols?

Interestingly enough, there a poly-time protocol with the same guarantee was also discovered later CITE.
What about, say, monotone functions? 
Poly-time algorithms?

- $[BKV05]$ $\exp(m)$ communication needed to achieve better than a $\sqrt{m}$ approximation
- $[LS05]$ Poly-time algorithm to match this $\sqrt{m}$ guarantee CITE.

What about protocols?
Interestingly enough, there a poly-time protocol with the same guarantee was also discovered later CITE.
So far no difference between truthful protocols and algorithms for the general case, and for approximation guarantees.
Restrictions on valuation functions

VCG shows that truth protocols for general valuation functions are as hard as algorithms. However, the protocol is also NP-hard and takes exponential communication. Approximation in the general monotone case also fails to separate protocols from algorithms. So we restrict the class of valuation functions to see what happens.
Restrictions on valuation functions

VCG shows that truth protocols for general valuation functions are as hard as algorithms. However, the protocol is also NP-hard and takes exponential communication. Approximation in the general monotone case also fails to separate protocols from algorithms. So we restrict the class of valuation functions to see what happens.

Additive

\[ v(X) + v(x) = v(X \cup \{x\}) \]
Restrictions on valuation functions

VCG shows that truth protocols for general valuation functions are as hard as algorithms. However, the protocol is also NP-hard and takes exponential communication. Approximation in the general monotone case also fails to separate protocols from algorithms. So we restrict the class of valuation functions to see what happens.

**Additive**

\[ v(X) + v(x) = v(X \cup \{x\}) \]

**Submodular**

If \( Y \subseteq X \), then \( v(X \cup \{x\}) - v(X) \leq v(Y \cup \{x\}) - v(Y) \)
Restrictions on valuation functions

VCG shows that truth protocols for general valuation functions are as hard as algorithms. However, the protocol is also NP-hard and takes exponential communication. Approximation in the general monotone case also fails to separate protocols from algorithms. So we restrict the class of valuation functions to see what happens.

Additive

\[ v(X) + v(x) = v(X \cup \{x\}) \]

Submodular

If \( Y \subseteq X \), then

\[ v(X \cup \{x\}) - v(X) \leq v(Y \cup \{x\}) - v(Y) \]

XOS

\[ v(X) = \max_{v_i} v_i(X) \text{ where } v_i \text{ are all additive} \]
Important for today

Binary Additive

\[ v(X) + v(x) = v(X \cup \{x\}) \text{ and } v(\{x\}) \in \{0, 1\} \]

Useful to think of as

\[ v(X) = |X \cap A| \]

Binary XOS

\[ v(X) = \max_i v_i(X) \]

where \( v_i \) are all binary additive.

Useful way to think of as

\[ v(X) = \max_i |A_i \cap X| \]

Additive is easy. XOS is harder. Submodular is the hardest (XOS is special case of submodular).
Binary Additive

\[ v(X) + v(x) = v(X \cup \{x\}) \text{ and } v(\{x\}) \in \{0, 1\}. \]

Useful to think of as \[ v(X) = |X \cap A| \]
Important for today

Binary Additive

\[ v(X) + v(x) = v(X \cup \{x\}) \text{ and } v(\{x\}) \in \{0, 1\}. \]
Useful to think of as \( v(X) = |X \cap A| \)

Binary XOS

\[ v(X) = \max_i v_i(X) \text{ where } v_i \text{ are all binary additive.} \]
Useful way to think of as \( v(X) = \max_i |A_i \cap X| \)

Additive is easy. XOS is harder. Submodular is the hardest (XOS is special case of submodular)
Table of Contents

1 Problem
   - Setting
   - Results

2 Some History
   - Mechanism Design 101
   - The price of truthfulness
   - Strong lower bounds
   - Something strange
   - A reduction to simultaneous protocols

3 Approximation results
   - Warmup
   - k-summary
   - Lower bound

4 Recap
Submodular functions – Approximation Algorithms

Algorithms

- Exact solution is NP-Hard.
- Simple greedy 2 approximation algorithm for sub modular functions that runs in poly-time* [LLN01].
- Improved algorithm to 1-1/e approximation and proved the bound is tight. [DV12]

Protocols

- Exact solution is NP-Hard.
- Can’t do better than a m-approximation. [DV12]
- There exists a mechanism with a m-approximation and poly-time. [DSS15, Dob11, DV11, DV12].

So there is a gap

*Here polytime means poly(n,m) calls to the value oracles, or poly(n,m, k) where k is the description complexity of the valuation functions.
Table of Contents

1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
   • Lower bound

4 Recap
So we are done right?
So we are done right?

Not yet..
A contradiction?

A natural truthful mechanism that achieves an $O(\log m)$ approximation for XOS functions. 

[Dob07, KV12, Dob16a] Improved the mechanism to achieve an $O(\sqrt{\log m})$ approximation for XOS functions.

But submodular $\subseteq$ XOS and surely $O(\sqrt{\log m}) < O(m)$. Contradiction?

Braverman, Mao, Weinberg

On simultaneous two player combinatorial...
A contradiction?

[DNS06]

A natural truthful mechanism that achieves a $O(\log^2 m)$ approximation for XOS functions.
A contradiction?

[DNS06]
A natural truthful mechanism that achieves a $O(\log^2 m)$ approximation for XOS functions.

[Dob07, KV12, Dob16a]
Improved the mechanism to achieve a $O(\sqrt{\log m})$ approximation for XOS functions.
A contradiction?

**[DNS06]**

A natural truthful mechanism that achieves a $O(\log^2 m)$ approximation for XOS functions.

**[Dob07, KV12, Dob16a]**

Improved the mechanism to achieve a $O(\sqrt{\log m})$ approximation for XOS functions.

But submodular $\subseteq$ XOS and surely $O(\sqrt{\log m}) < O(m)$

Contradiction?
A contradiction?

[DNS06]
A natural truthful mechanism that achieves a $O(\log^2 m)$ approximation for XOS functions.

[Dob07, KV12, Dob16a]
Improved the mechanism to achieve a $O(\sqrt{\log m})$ approximation for XOS functions.

But submodular $\subseteq$ XOS and surely $O(\sqrt{\log m}) < O(m)$

Not really.
(The ** with tiny text are important!)
What’s the catch?

These "natural" truthful mechanisms are essentially, what is called in literature, "posted price" mechanisms.
What’s the catch?

These "natural" truthful mechanisms are essentially, what is called in literature, "posted price" mechanisms.

- Run the protocol
What's the catch?

These "natural" truthful mechanisms are essentially, what is called in literature, "posted price" mechanisms.

- Run the protocol
- Provide a "price" $p_i$ for each item $i$ to the bidder, and let them pick their favorite subset
What’s the catch?

These "natural" truthful mechanisms are essentially, what is called in literature, "posted price" mechanisms.

- Run the protocol
- Provide a "price" $p_i$ for each item $i$ to the bidder, and let them pick their favorite subset

The catch is that usually, finding the best subset is by itself NP hard given a succinct representation of valuations, or needs exp oracle queries.
What's the catch?

These "natural" truthful mechanisms are essentially, what is called in literature, "posted price" mechanisms.

- Run the protocol
- Provide a "price" $p_i$ for each item $i$ to the bidder, and let them pick their favorite subset

The catch is that usually, finding the best subset is by itself NP hard given a succinct representation of valuations, or needs exp oracle queries. So by asking the bidder to pick their favorite subset, the protocol is essentially "outsourcing" some of the computation.
What's the catch?

These "natural" truthful mechanisms are essentially, what is called in literature, "posted price" mechanisms.

- Run the protocol
- Provide a "price" $p_i$ for each item $i$ to the bidder, and let them pick their favorite subset

The catch is that usually, finding the best subset is by itself NP hard given a succinct representation of valuations, or needs exp oracle queries. So by asking the bidder to pick their favorite subset, the protocol is essentially "outsourcing" some of the computation.

So really, the "polytime" of these protocols is the communication complexity.
Some consolation

So, while the existing separation is great, a good question to try so solve is a separation for the communication complexity of this problem. A lower bound there would thus extend to all "natural" seeming mechanism.
Some consolation

But how do we deal with lower bounds for communication complexity while incentivizing truthfulness?
Some consolation

But how do we deal with lower bounds for communication complexity while incentivizing truthfulness?

All the lower bounds known in literature for truthful mechanisms are actually lower bounds for algorithms in general. There is no known bound that holds for mechanisms and not for algorithms.
Some consolation

A gap is conjectured
In the XOS case:

What we know is $(1-1/e)$ for algorithms (tight) vs $\sqrt{\log m}$ for protocols (best known yet)
Some consolation

A gap is conjectured
In the XOS case:
What we know is $(1 - 1/e)$ for algorithms (tight) vs $\sqrt{\log m}$ for protocols (best known yet)

In the 2 player case, we know a 3/4-approximation for algorithm (tight), vs 1/2-approximation for protocols (give to whoever values entire bundle most) and this trivial protocol is the best we know for 2 players.
Table of Contents

1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
   • Lower bound

4 Recap
Reduction to simultaneous protocols

[Dob16b]
Poly-time, truthful, interactive protocol for an $\alpha$–approximation for a class of functions $\implies$ A poly-time simultaneous protocol, (not necessarily truthful) for an $\alpha$–approximation for that same class of functions.
Reduction to simultaneous protocols

[Dob16b]
Poly-time, truthful, interactive protocol for an $\alpha-$approximation for a class of functions $\implies$ A poly-time simultaneous protocol, (not necessarily truthful) for an $\alpha-$approximation for that same class of functions.

Corollary (Lower bound implication)

Any simultaneous protocol needs exp communication for a better than $\alpha-$approximation $\implies$ No poly-time truthful mechanism can do better than $\alpha -$ approximation.
We are finally there!
One last weirdness left
Well, almost
Sketching functions

Reminder: Result

Result: 3/4 approximation for allocation problem
Sketching functions

Reminder: Result

Result: 3/4 approximation for allocation problem

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

Trivia: Authors conjectured that a better than 1/2 approximation was not possible when they started off. Ended up with a 3/4 approximation, which even matches the bound of an fully interactive protocol.
Sketching functions

Reminder: Result

**Result: 3/4 approximation for allocation problem**

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

**Sketching bounds**

Any sketching scheme for XOS valuations that allows value queries to be evaluated with a $o(m)$-factor requires super-poly(m) size.
Sketching functions

Reminder: Result

**Result: 3/4 approximation for allocation problem**

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

**Sketching bounds**

Any sketching scheme for XOS valuations that allows value queries to be evaluated with a $o(m)$-factor requires super-poly(m) size.

So the simultaneous protocol is somehow now transmitting enough "information" about the valuation.
Sketching functions

Reminder: Result

Result: 3/4 approximation for allocation problem

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

Sketching bounds

Any sketching scheme for XOS valuations that allows value queries to be evaluated with a \( o(m) \)-factor requires super-poly(m) size.

So the simultaneous protocol is somehow now transmitting enough "information" about the valuation.

Trivia: Authors conjectured that a better than 1/2 approximation was not possible when they started off.
Sketching functions

Reminder: Result

Result: 3/4 approximation for allocation problem

Exists a randomized simultaneous protocol with poly(m) communication that guarantees a 3/4 approximation for binary XOS valuations*

Sketching bounds

Any sketching scheme for XOS valuations that allows value queries to be evaluated with a $o(m)$-factor requires super-poly(m) size.

So the simultaneous protocol is somehow now transmitting enough "information" about the valuation.

Trivia: Authors conjectured that a better than 1/2 approximation was not possible when they started off. Ended up with a 3/4 approximation, which even matches the bound of an fully interactive protocol.
Table of Contents

1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
   • Lower bound

4 Recap
We first start off by a quick warmup that will motivate the definitions and ideas of the full proof.
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.
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.

We will glance over the generalized definition for the general XOS case (proof sketch is similar) if time.
Table of Contents

1 Problem
   - Setting
   - Results

2 Some History
   - Mechanism Design 101
   - The price of truthfulness
   - Strong lower bounds
   - Something strange
   - A reduction to simultaneous protocols

3 Approximation results
   - Warmup
   - k-summary
   - Lower bound

4 Recap
Quick reminders

Recall the problem and the setting

- 2 Players, Alice and Bob, and an auctioneer

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 to maximize

\[
SW(S) = v_A(S) + v_B(\overline{S})
\]

Approach

Alice and Bob simultaneously announce some information. Auctioneer uses information to allocate \( S, \overline{S} \) to Alice and Bob respectively.
Quick reminders

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 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 respectively.
Quick reminders

Recall the problem and the setting

- 2 Players, Alice and Bob, and an auctioneer
- Alice and Bob have valuation functions $v_A, v_B : 2^m \rightarrow \mathbb{R}_+$
- Valuation functions are Binary XOS (BXOS)
Quick reminders

Recall the problem and the setting

- 2 Players, Alice and Bob, and an auctioneer
- Alice and Bob have valuation functions $v_A, v_B : 2^{|m|} \rightarrow \mathbb{R}_+$
- 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 $\max_i |A_i \cap S|$
Quick reminders

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 to maximize $\mathcal{SW}(S) = v_A(S) + v_B(\bar{S})$
Quick reminders

Recall the problem and the setting

- 2 Players, Alice and Bob, and an auctioneer
- Alice and Bob have valuation functions \( v_A, v_B : 2^m \rightarrow \mathbb{R}_+ \)
- 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 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.
Quick reminders

Recall the problem and the setting

- 2 Players, Alice and Bob, and an auctioneer
- Alice and Bob have valuation functions \( v_A, v_B : 2^m \rightarrow \mathbb{R}_+ \)
- 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 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.

\[
\begin{align*}
1/2\text{-approximation for allocation problem} \\
\text{Let's start in a restricted setting where Alice and Bob announce only 1 set.}
\end{align*}
\]
1/2-approximation for allocation problem

Let’s start in a restricted setting where Alice and Bob announce only 1 set. Question: If both could announce just 1 set each, then which would they pick?

Claim: Each pick the largest set they have. Auctioneer flips coin to pick one of these sets, and gives all items to the player, and rest to other.

Proof of Claim:

\[|A_{\text{max}}| = \max_i |A_i \cap [m]| = v_A([m]) \geq v_A(S)\]

\[|B_{\text{max}}| = \max_i |B_i \cap [m]| = v_B([m]) \geq v_B(\bar{S})\]

Output = \[1/2 \left( SW(A) + SW(\bar{B}) \right) \geq 1/2 \left( v_A([m]) + v_B([m]) \right) \geq 1/2 \left( v_A(S) + v_B(\bar{S}) \right) \]
1/2-approximation for allocation problem

Let’s start in a restricted setting where Alice and Bob announce only 1 set. **Question**: If both could announce just 1 set each, then which would they pick?

**Claim**: Each pick the largest set they have. Auctioneer flips coin to pick one of these sets, and gives all items to the player, and rest to other.
Let’s start in a restricted setting where Alice and Bob announce only 1 set. **Question**: If both could announce just 1 set each, then which would they pick?

**Claim**: Each pick the largest set they have. Auctioneer flips coin to pick one of these sets, and gives all items to the player, and rest to other.

**Proof of Claim**: 
\[ |A_{\text{max}}| = \max_i |A_i \cap [m]| = v_A([m]) \geq v_A(S) \]
1/2-approximation for allocation problem

Let’s start in a restricted setting where Alice and Bob announce only 1 set. 
**Question:** If both could announce just 1 set each, then which would they pick?

**Claim:** Each pick the largest set they have. Auctioneer flips coin to pick one of these sets, and gives all items to the player, and rest to other. 

**Proof of Claim:**

$$|A_{\text{max}}| = \max_i |A_i \cap [m]| = v_A([m]) \geq v_A(S)$$

$$|B_{\text{max}}| = \max_i |B_i \cap [m]| = v_B([m]) \geq v_B(\bar{S})$$
1/2-approximation for allocation problem

Let's start in a restricted setting where Alice and Bob announce only 1 set. **Question:** If both could announce just 1 set each, then which would they pick?

**Claim:** Each pick the largest set they have. Auctioneer flips coin to pick one of these sets, and gives all items to the player, and rest to other.

**Proof of Claim:**

\[
|A_{\text{max}}| = \max_i |A_i \cap [m]| = \nu_A([m]) \geq \nu_A(S) \\
|B_{\text{max}}| = \max_i |B_i \cap [m]| = \nu_B([m]) \geq \nu_B(\bar{S})
\]

Output =

\[
1/2(SW(A) + SW(\bar{B})) \geq 1/2(\nu_A([m]) + \nu_B([m])) \geq 1/2(\nu_A(S) + \nu_B(\bar{S})) = 1/2(SW(\nu_A, \nu_B))
\]
1/2-approximation for allocation problem

Let's start in a restricted setting where Alice and Bob announce only 1 set. **Question**: If both could announce just 1 set each, then which would they pick?

**Claim**: Each pick the largest set they have. Auctioneer flips coin to pick one of these sets, and gives all items to the player, and rest to other.

**Proof of Claim**:

\[
|A_{\text{max}}| = \max_i |A_i \cap [m]| = v_A([m]) \geq v_A(S) \\
|B_{\text{max}}| = \max_i |B_i \cap [m]| = v_B([m]) \geq v_B(\bar{S})
\]

Output = 
\[
1/2(SW(A) + SW(\bar{B})) \geq 1/2(v_A([m]) + v_B([m])) \geq 1/2(v_A(S) + v_B(\bar{S})) = 1/2(SW(v_A, v_B))
\]
2/3-approximation for allocation problem

Let’s start in a restricted setting where Alice and Bob can now announce 2 sets.
2/3-approximation for allocation problem

Let's start in a restricted setting where Alice and Bob can now announce 2 sets.

**Question:** If both could announce just 2 sets each, what would they pick?
2/3-approximation for allocation problem

Let’s start in a restricted setting where Alice and Bob can now announce 2 sets.

**Question:** If both could announce just 2 sets each, what would they pick?

**Possible Strategy:** 2 clauses with the largest union so they cover as much of $[m]$ as possible!
2/3-approximation for allocation problem

Let’s start in a restricted setting where Alice and Bob can now announce 2 sets.

**Question**: If both could announce just 2 sets each, what would they pick?

**Possible Strategy**: 2 clauses with the largest union so they cover as much of $[m]$ as possible!

What about 3 clauses – now we can send both the largest clause, but also the 2 with the largest union.
2/3-approximation for allocation problem

Let’s start in a restricted setting where Alice and Bob can now announce 2 sets.

**Question:** If both could announce just 2 sets each, what would they pick?

**Possible Strategy:** 2 clauses with the largest union so they cover as much of \([m]\) as possible!

What about 3 clauses – now we can send both the largest clause, but also the 2 with the largest union.

**Claim**

Bob picks \(B_i\) which maximizes \(|B_i|\) and \(B_{j1}, B_{j2}\) which maximize 
\[ SW(B_{j1}, B_{j2}) = |B_{j1} \cup B_{j2}| \]
Alice does the same with \(A_k, A_{k1}\) and \(A_{k2}\)
Auctioneers strategy coming soon ...
2/3-Approximation for allocation

Protocol I

- Alice picks $A_i$ which maximizes $|A_i|$ and $A_{j1}, A_{j2}$ which maximize $\mathcal{SW}(A_{j1}, A_{j2}) = |A_{j1} \cup A_{j2}|$. 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.
2/3-Approximation for allocation

Protocol I

- Alice picks $A_i$ which maximizes $|A_i|$ and $A_{j_1}, A_{j_2}$ which maximize $\mathcal{SW}(A_{j_1}, A_{j_2}) = |A_{j_1} \cup A_{j_2}|$. 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

Protocol I is a 2/3 approximation for the allocation problem.
2/3-Approximation for allocation

Protocol I

- Alice picks $A_i$ which maximizes $|A_i|$ and $A_{j_1}, A_{j_2}$ which maximize $SW(A_{j_1}, A_{j_2}) = |A_{j_1} \cup A_{j_2}|$. 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

Protocol I is a 2/3 approximation for the allocation problem.
2/3-Approximation for allocation

**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**: Basically, Alice only wants items in \( X \), so giving items from \( X \) to Bob cannot increase the welfare.
2/3-Approximation for allocation

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: 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
Recall

- $A_1 = \max |A_i|$. $A_2, A_3 = \arg \max |A_i \cup A_j|$

- Auctioneer allocates all items in randomly sent clause $A_i$ to Alice and the rest to Bob
2/3-Approximation for allocation

Recall

- $A_1 = \max |A_i|$. $A_2, A_3 = \arg \max |A_i \cup A_j|$

- 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 $\nu_A, \nu_B$. $A_{OPT}, B_{OPT}$ are the participating clauses.

- $OPT(\nu_A, \nu_B) = |A_{OPT} \cup B_{OPT}| = SW(A_{OPT}) = SW(\bar{B}_{OPT}) = SW(S)$

- $OPT(\nu_A, \nu_A) = |A_2 \cup A_3| = |A_2| + |\bar{A}_2 \cap A_3| \geq |S \cap A_{OPT}| + |\bar{S} \cap A_1|$
2/3-Approximation for allocation

Notation

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

Proof sketch: Algorithm gets $1/3$rd of

$$\sum_{i=1}^{3} SW(A_i, v_B)$$

$$= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3))$$
2/3-Approximation for allocation

Notation

- \( S, \bar{S} \) is any real OPT partition for \( v_A, v_B \)
- \( A_{OPT}, B_{OPT} \) are the participating clauses.

\[
OPT(v_A, v_B) = |A_{OPT} \cup B_{OPT}| = SW(A_{OPT}) = SW(\bar{B}_{OPT}) = SW(S)
\]

\[
OPT(v_A, v_A) = |A_2 \cup A_3| = |A_2| + |\bar{A}_2 \cap A_3| \geq |S \cap A_{OPT}| + |\bar{S} \cap A_1|
\]

Proof sketch: Algorithm gets 1/3rd of

\[
\sum_{i=1}^{3} SW(A_i, v_B)
\]

\[
= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3))
\]

\[
\geq (|A_1 \cap S| + v_B(\bar{S}) + |A_2| + v_B(\bar{A}_2) + |\bar{A}_2 \cap A_3| + v_B(A_2))
\]
2/3-Approximation for allocation

Notation

- \( S, \bar{S} \) is any real OPT partition for \( v_A, v_B \) \( A_{OPT}, B_{OPT} \) are the participating clauses.

- \( OPT(v_A, v_B) = |A_{OPT} \cup B_{OPT}| = SW(A_{OPT}) = SW(B_{OPT})) = SW(S) \)

- \( OPT(v_A, v_A) = |A_2 \cup A_3| = |A_2| + |\bar{A}_2 \cap A_3| \geq |S \cap A_{OPT}| + |\bar{S} \cap A_1| \)

Proof sketch: Algorithm gets 1/3rd of \( \sum_{i=1}^{3} SW(A_i, v_B) \)

\[
= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3)) \\
\geq (|A_1 \cap S| + v_B(\bar{S}) + |A_2| + v_B(\bar{A}_2) + |\bar{A}_2 \cap A_3| + v_B(A_2))
\]
2/3-Approximation for allocation

Notation
- $S, \bar{S}$ is any real OPT partition for $v_A, v_B$ $A_{OPT}, B_{OPT}$ are the participating clauses.

- $OPT(v_A, v_B) = |A_{OPT} \cup B_{OPT}| = SW(A_{OPT}) = SW(\bar{B}_{OPT})) = SW(S)$
- $OPT(v_A, v_A) = |A_2 \cup A_3| = |A_2| + |\bar{A}_2 \cap A_3| \geq |S \cap A_{OPT}| + |\bar{S} \cap A_1|$

Proof sketch: Algorithm gets 1/3rd of $\sum_{i=1}^{3} SW(A_i, v_B)$

$$= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3))$$

$$\geq (|A_1 \cap S| + v_B(\bar{S}) + |A_2| + v_B(\bar{A}_2) + |\bar{A}_2 \cap A_3| + v_B(A_2))$$

$$\geq (|A_1 \cap S| + v_B(\bar{S}) + |S \cap A_{OPT}| + v_B(\bar{A}_2) + |\bar{S} \cap A_1| + v_B(A_2))$$
2/3-Approximation for allocation

Notation

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

Proof sketch: Algorithm gets 1/3rd of 
$\sum_{i=1}^{3} SW(A_i, v_B)$

$$= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3))$$

$$\geq (|A_1 \cap S| + v_B(\bar{S}) + |A_2| + v_B(\bar{A}_2) + |\bar{A}_2 \cap A_3| + v_B(A_2))$$

$$\geq (|A_1 \cap S| + v_B(\bar{S}) + |S \cap A_{OPT}| + v_B(\bar{A}_2) + |\bar{S} \cap A_1| + v_B(A_2))$$
2/3-Approximation for allocation

Notation

- $S, \bar{S}$ is any real OPT partition for $v_A, v_B$ $A_{OPT}, B_{OPT}$ are the participating clauses.
- $OPT(v_A, v_B) = |A_{OPT} \cup B_{OPT}| = SW(A_{OPT}) = SW(\bar{B}_{OPT}) = SW(S)$
- $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|

Proof sketch: Algorithm gets 1/3rd of

$\sum_{i=1}^{3} SW(A_i, v_B)$

$= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3))$

$\geq (|A_1 \cap S| + v_B(\bar{S}) + |A_2| + v_B(\bar{A}_2) + |\bar{A}_2 \cap A_3| + v_B(A_2))$

$\geq (|A_1 \cap S| + v_B(\bar{S}) + |S \cap A_{OPT}| + v_B(\bar{A}_2) + |\bar{S} \cap A_1| + v_B(A_2))$

$\geq (|A_1| + |B_{OPT}| + v_B(\bar{S}) + |S \cap A_{OPT}|)$
2/3-Approximation for allocation

Notation

- $S, \bar{S}$ is any real OPT partition for $v_A, v_B$. 
- $A_{OPT}, B_{OPT}$ are the participating clauses.

$OPT(v_A, v_B) = |A_{OPT} \cup B_{OPT}| = SW(A_{OPT}) = SW(\bar{B}_{OPT}) = SW(S)$

$OPT(v_A, v_A) = |A_2 \cup A_3| = |A_2| + |\bar{A}_2 \cap A_3| \geq |S \cap A_{OPT}| + |\bar{S} \cap A_1|$

**Proof sketch**: Algorithm gets $1/3$rd of

$$\sum_{i=1}^{3} SW(A_i, v_B)$$

$$= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3))$$

$$\geq (|A_1 \cap S| + v_B(\bar{S}) + |A_2| + v_B(\bar{A}_2) + |\bar{A}_2 \cap A_3| + v_B(A_2))$$

$$\geq (|A_1 \cap S| + v_B(\bar{S}) + |S \cap A_{OPT}| + v_B(\bar{A}_2) + |\bar{S} \cap A_1| + v_B(A_2))$$

$$\geq (|A_1| + |B_{OPT}| + v_B(\bar{S}) + |S \cap A_{OPT}|)$$

$$= (|A_1| + |B_{OPT}| + |B_{OPT} \cap \bar{S}| + |S \cap A_{OPT}|)$$
2/3-Approximation for allocation

Notation

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

Proof sketch: Algorithm gets $1/3$rd of

$$\sum_{i=1}^{3} SW(A_i, v_B)$$

$$= (|A_1| + v_B(\tilde{A}_1) + |A_2| + v_B(\tilde{A}_2) + |A_3| + v_B(\tilde{A}_3))$$

$$\geq (|A_1 \cap S| + v_B(\tilde{S}) + |A_2| + v_B(\tilde{A}_2) + |\tilde{A}_2 \cap A_3| + v_B(A_2))$$

$$\geq (|A_1 \cap S| + v_B(\tilde{S}) + |S \cap A_{OPT}| + v_B(\tilde{A}_2) + |\tilde{S} \cap A_1| + v_B(A_2))$$

$$\geq (|A_1| + |B_{OPT}| + v_B(\tilde{S}) + |S \cap A_{OPT}|)$$

$$= (|A_1| + |B_{OPT}| + |B_{OPT} \cap \tilde{S}| + |S \cap A_{OPT}|)$$

$$\geq (|A_{OPT} \cap \tilde{B}_{OPT}| + |B_{OPT}| + |B_{OPT} \cap \tilde{S}| + |S \cap A_{OPT}|)$$
2/3-Approximation for allocation

Notation

- $S, \bar{S}$ is any real OPT partition for $v_A, v_B$. $A_{OPT}, B_{OPT}$ are the participating clauses.

- $OPT(v_A, v_B) = |A_{OPT} \cup B_{OPT}| = SW(A_{OPT}) = SW(\bar{B}_{OPT}) = SW(S)$

- $OPT(v_A, v_A) = |A_2 \cup A_3| = |A_2| + |\bar{A}_2 \cap A_3| \geq |S \cap A_{OPT}| + |\bar{S} \cap A_1|$

Proof sketch: Algorithm gets 1/3rd of $\sum_{i=1}^{3} SW(A_i, v_B)$

$$= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3))$$

$$\geq (|A_1 \cap S| + v_B(\bar{S}) + |A_2| + v_B(\bar{A}_2) + |\bar{A}_2 \cap A_3| + v_B(A_2))$$

$$\geq (|A_1 \cap S| + v_B(\bar{S}) + |S \cap A_{OPT}| + v_B(\bar{A}_2) + |\bar{S} \cap A_1| + v_B(A_2))$$

$$\geq (|A_1| + |B_{OPT}| + v_B(\bar{S}) + |S \cap A_{OPT}|)$$

$$= (|A_1| + |B_{OPT}| + |B_{OPT} \cap \bar{S}| + |S \cap A_{OPT}|)$$

$$\geq (|A_{OPT} \cap \bar{B}_{OPT}| + |B_{OPT}| + |B_{OPT} \cap \bar{S}| + |S \cap A_{OPT}|)$$

$$= SW(\bar{B}_{OPT}) + SW(S)$$
2/3-Approximation for allocation

Notation

- $S, \bar{S}$ is any real OPT partition for $v_A, v_B$ $A_{OPT}, B_{OPT}$ are the participating clauses.

- $OPT(v_A, v_B) = |A_{OPT} \cup B_{OPT}| = SW(A_{OPT}) = SW(\bar{B}_{OPT}) = SW(S)$

- $OPT(v_A, v_A) = |A_2 \cup A_3| = |A_2| + |\bar{A}_2 \cap A_3| \geq |S \cap A_{OPT}| + |\bar{S} \cap A_1|$ 

Proof sketch: Algorithm gets 1/3rd of 

$\sum_{i=1}^{3} SW(A_i, v_B)$

\[
= (|A_1| + v_B(\bar{A}_1) + |A_2| + v_B(\bar{A}_2) + |A_3| + v_B(\bar{A}_3)) \\
\geq (|A_1 \cap S| + v_B(\bar{S}) + |A_2| + v_B(\bar{A}_2) + |\bar{A}_2 \cap A_3| + v_B(A_2)) \\
\geq (|A_1 \cap S| + v_B(\bar{S}) + |S \cap A_{OPT}| + v_B(\bar{A}_2) + |\bar{S} \cap A_1| + v_B(A_2)) \\
\geq (|A_1| + |B_{OPT}| + v_B(\bar{S}) + |S \cap A_{OPT}|) \\
= (|A_1| + |B_{OPT}| + |B_{OPT} \cap \bar{S}| + |S \cap A_{OPT}|) \\
\geq (|A_{OPT} \cap \bar{B}_{OPT}| + |B_{OPT}| + |B_{OPT} \cap \bar{S}| + |S \cap A_{OPT}|) \\
= SW(\bar{B}_{OPT}) + SW(S)) \\
= 2OPT(v_A, v_B)
\]
2/3 approximation for allocation

Key takeaways

- 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.

But what happens if we take Bobs input too? Can we get a similar allocation guarantee while knowing the social welfare that it leads to?
3/5 approximation for decision (and allocation)

### 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 $A_1, A_2, A_3$.
- 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.

### Claim
Protocol II is a 3/5 approximation for the decision and allocation) problem.
3/5 approximation for decision (and allocation)

**Protocol II**

- Alice picks $A_i$ which maximizes $|A_i|$ and $A_{j_1}, A_{j_2}$ which maximize $SW(A_{j_1}, A_{j_2}) (= |A_{j_1} \cup A_{j_2}|)$. 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.

**Claim**

Protocol II 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.

**Turns out,** we can remove the "binary" assumption and just use $\max SW(X, Y)$ and still get the same guarantee.
Table of Contents

1 Problem
   - Setting
   - Results

2 Some History
   - Mechanism Design 101
   - The price of truthfulness
   - Strong lower bounds
   - Something strange
   - A reduction to simultaneous protocols

3 Approximation results
   - Warmup
   - k-summary
   - Lower bound

4 Recap
Generalizing

How does one generalize this notion of "largest" clause as well as "clauses that cover the largest section"

Observations:

- 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..
Generalizing - Binary case

How does one generalize this notion of "largest" clause as well as "clauses that cover the largest section"

Suppose Alice were to send $k$ clauses $\{A_i\}$. Let $x_i = \frac{1}{k} \sum_{j=1}^{k} 1_{i \in C_j}$

Intuitively, $x_i$ captures how much importance the $k$ clauses give the element $i$

Observations:

Maximize $\sum x_i$ – what is the answer when $k = 1$?

Minimize $\sum x_i^2$ – not immediately obvious that this is the right way to do things.

Somehow need a balance.. maximize $\sum x_i - \alpha x_i^2$

Definition $A(k, \alpha)$ – summary is the set of $k$ clauses defined as $(A_1, A_2, \ldots, A_k) = \arg\max (C_1, C_2, \ldots, C_k \in v_A) \sum x_i - \alpha x_i^2$ and $x_i = \frac{1}{k} \sum_{j=1}^{k} 1_{i \in C_j}$ is as defined above.
Generalizing - Binary case

How does one generalize this notion of "largest" clause as well as "clauses that cover the largest section"

Suppose Alice were to send $k$ clauses $\{A_i\}$. Let $x_i = 1/k \sum_{j=1}^{k} 1_{i \in A_j}$
Generalizing - Binary case

How does one generalize this notion of "largest" clause as well as "clauses that cover the largest section"

Suppose Alice were to send k clauses \( \{A_i\} \). Let \( x_i = \frac{1}{k} \sum_{j=1}^{k} \mathbb{1}_{i \in A_j} \)

Intuitively, \( x_i \) captures how much importance the k clauses give the element \( i \).
Generalizing - Binary case

How does one generalize this notion of "largest" clause as well as "clauses that cover the largest section"

Suppose Alice were to send $k$ clauses $\{A_i\}$. Let $x_i = 1/k \sum_{j=1}^{k} \mathbb{1}_{i \in A_j}$

Intuitively, $x_i$ captures how much importance the $k$ clauses give the element $i$

Observations:
Generalizing - Binary case

How does one generalize this notion of "largest" clause as well as "clauses that cover the largest section"

Suppose Alice were to send $k$ clauses $\{A_i\}$. Let $x_i = 1/k \sum_{j=1}^{k} 1_{i \in A_j}$

Intuitively, $x_i$ captures how much importance the $k$ clauses give the element $i$

Observations:

- Maximize $\sum x_i$ – what is the answer when $k = 1$?
Generalizing - Binary case

How does one generalize this notion of "largest" clause as well as "clauses that cover the largest section"

Suppose Alice were to send k clauses \( \{A_i\} \). Let \( x_i = 1/k \sum_{j=1}^{k} 1_{i \in A_j} \)

Intuitively, \( x_i \) captures how much importance the k clauses give the element \( i \)

Observations:

- Maximize \( \sum x_i \) – what is the answer when \( k = 1 \)?
- Minimize \( \sum x_i^2 \) – not immediately obvious that this is the right way to do things.
Generalizing - Binary case

How does one generalize this notion of "largest" clause as well as "clauses that cover the largest section"

Suppose Alice were to send $k$ clauses $\{A_i\}$. Let $x_i = 1/k \sum_{j=1}^{k} \mathbb{1}_{i \in A_j}$

Intuitively, $x_i$ captures how much importance the $k$ clauses give the element $i$

Observations:

- Maximize $\sum x_i$ – what is the answer when $k = 1$?
- Minimize $\sum x_i^2$ – not immediately obvious that this is the right way to do things.
- Somehow need a balance.. maximize $\sum x_i - \alpha x_i^2$
Generalizing - Binary case

How does one generalize this notion of "largest" clause as well as "clauses that cover the largest section"

Suppose Alice were to send k clauses \( \{A_i\} \). 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 \( i \)

Observations:

- Maximize \( \sum x_i \) – what is the answer when \( k = 1 \)?
- Minimize \( \sum x_i^2 \) – not immediately obvious that this is the right way to do things.
- Somehow need a balance.. maximize \( \sum x_i - \alpha x_i^2 \)

Definition

A \((k, \alpha)\)–summary is the set of k clauses defined as

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

and \( x_i = \frac{1}{k} \sum_{j=1}^{k} 1_{i \in C_j} \) is as defined above.
Sanity check for $(k, \alpha)$ - summaries

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!!

Simple proofs – skipped but fun to try out!
Protocols

Recall:

- 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 the 2/3-BXOS, auctioneer did not know the value.
Protocols

Recall:

- 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.
Protocols

Recall:

- 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.
Recall:

- 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.
Protocols

Recall:

- 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

Results

**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>
Results

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>

- 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.
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>

- 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 $(k, \alpha)$-summary)
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 \((k, \alpha)\)-summary)

Still open whether it is possible to beat 23/32 with a deterministic protocol for the allocation problem!
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>(\perp)</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>

- 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 \((k,\alpha)\)-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 \((k,\alpha)\)-summary (skipped proof)
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 \((k, \alpha)\)-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 \((k, \alpha)\)-summary (skipped proof)

No simultaneous, randomized protocol can do better than \((3/4-1/108)\) approximation with less than \(\exp(m)\) communication (WHP)
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 \((k, \alpha)\)-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 \((k, \alpha)\)-summary (skipped proof)

No simultaneous, randomized protocol can do better than \((3/4 - 1/108)\) approximation with less than \(\exp(m)\) communication (WHP)

Finally, we sketch a proof for the lower bound on the communication.
Table of Contents

1 Problem
   • Setting
   • Results

2 Some History
   • Mechanism Design 101
   • The price of truthfulness
   • Strong lower bounds
   • Something strange
   • A reduction to simultaneous protocols

3 Approximation results
   • Warmup
   • k-summary
     • Lower bound

4 Recap
Proof sketch: The idea is to create exp 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 ....
3/4-108 needs exponential communication simultaneously

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 $|S| = |T| = m/6$
3/4-108 needs exponential communication simultaneously

A bit more formally..

1. First sample a \( S \) and a \( T \) uniformly at random from all sets such that 
   \[ |S \cap T| = m/6 \text{ and } |S| = |T| = m/6 \]
2. Sample \( \exp \) many sets \( A_i \) such that 
   \[ |A_i \cap S| = m/3 \text{ and } |A_i \cap \bar{S}| = m/6 \]
   \( (|A_i| = m/2) \)
3/4-108 needs exponential communication simultaneously

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 $|B_i \cap \bar{T}| = m/6$ ($|B_i| = m/2$)
3/4-108 needs exponential communication simultaneously

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.
3/4-108 needs exponential communication simultaneously

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
3/4-108 needs exponential communication simultaneously

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
3/4-108 needs exponential communication simultaneously

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 \( m \).
3/4-108 needs exponential communication simultaneously

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)
3/4-108 needs exponential communication simultaneously

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)
Main idea: To be able to do better than $3/4$ at decision algorithm needs to tell Case 1 and Case 2 apart.
**Main idea:** 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).
Main idea: 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. Can prove this with a bound using mutual information! between coin toss and information sent to auctioneer by Bob and Alice.
Main idea: 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. Can prove this with a bound using mutual information! between coin toss and information sent to auctioneer by Bob and Alice.
Table of Contents

1 Problem
   - Setting
   - Results

2 Some History
   - Mechanism Design 101
   - The price of truthfulness
   - Strong lower bounds
   - Something strange
   - A reduction to simultaneous protocols

3 Approximation results
   - Warmup
   - k-summary
   - Lower bound

4 Recap
Recap

DOB reduction says L.B. for simultaneous communication complexity of welfare maximization $\implies$ lower bound on communication for truthful welfare maximization protocol.

Goal: Find lower bound for sim. CC of XOS/BXOS valuations.

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!)

Additional not mentioned:

- 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

In case you didn't notice, no lower bound of the type that was actually needed was proved :-(

But came pretty close and this is an open problem!
Recap

DOB reduction says L.B. for simultaneous communication complexity of welfare maximization $\implies$ lower bound on communication for truthful welfare maximization protocol.
Goal: Find lower bound for sim. CC of XOS/BXOS valuations.
Recap

DOB reduction says L.B. for simultaneous communication complexity of welfare maximization \( \Rightarrow \) lower bound on communication for truthful welfare maximization protocol.
Goal: Find lower bound for sim. CC of XOS/BXOS valuations.

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!)

Additional not mentioned:
- 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

In case you didn’t notice, no lower bound of the type that was actually needed was proved :( But came pretty close and this is an open problem!
Recap

DOB reduction says L.B. for simultaneous communication complexity of welfare maximization \( \implies \) lower bound on communication for truthful welfare maximization protocol.

Goal: Find lower bound for sim. CC of XOS/BXOS valuations.

**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!)

Additional not mentioned:

- 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

In case you didn’t notice, no lower bound of the type that was actually needed was proved :(

But came pretty close and this is an open problem!
Recap

DOB reduction says L.B. for simultaneous communication complexity of welfare maximization $\implies$ lower bound on communication for truthful welfare maximization protocol.
Goal: Find lower bound for sim. CC of XOS/BXOS valuations.

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)
Recap

DOB reduction says L.B. for simultaneous communication complexity of welfare maximization $\implies$ lower bound on communication for truthful welfare maximization protocol.

Goal: Find lower bound for sim. CC of XOS/BXOS valuations.

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!)
Recap

DOB reduction says L.B. for simultaneous communication complexity of welfare maximization $\implies$ lower bound on communication for truthful welfare maximization protocol.
Goal: Find lower bound for sim. CC of XOS/BXOS valuations.

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 upped bound for approximation ratio of algorithms that does 23/32 for XOS problem (counter example is Binary!)

Additional not mentioned:

- Interesting new type of truthful protocol for BXOS that gets 3/4 ratio
Recap

DOB reduction says L.B. for simultaneous communication complexity of welfare maximization $\implies$ lower bound on communication for truthful welfare maximization protocol.
Goal: Find lower bound for sim. CC of XOS/BXOS valuations.

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 upped bound for approximation ratio of algorithms that does 23/32 for XOS problem (counter example is Binary!)

Additional not mentioned:

- 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
Recap

DOB reduction says L.B. for simultaneous communication complexity of welfare maximization $\implies$ lower bound on communication for truthful welfare maximization protocol.
Goal: Find lower bound for sim. CC of XOS/BXOS valuations.

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!)

Additional not mentioned:

- 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

In case you didn’t notice, no lower bound of the type that was actually needed was proved :( But came pretty close and this is an open problem!
Open problems!

- Deterministic 3/4 approximation for BXOS (interesting because would them hope to imply det. protocol)

- Deterministic 3/4 lower bound for general XOS (3/4 is best known). (Again, can use Dobzinski's reductions)
Open problems!

- 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)
Open problems!

- 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!)
Open problems!

- 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)
Open problems!

- 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)
Acknowledgements

Thanks to Prof. Weinberg and Jieming for clarifying questions about paper!!
Acknowledgements

Thanks to Prof. Weinberg and Jieming for clarifying questions about paper!!

Thank you all for listening and your patience!