Beyond Equilibria: Mechanisms for Repeated Combinatorial Auctions

Brendan Lucier
University of Toronto
Combinatorial Auctions

Bidders: 🟥_circle, 🟢_circle, 🟦_circle
Objects: 🌟_star, 💡_square, 🏼_triangle
Combinatorial Auctions

Bidders

Objects

Valuations
Combinatorial Auctions

Bidders

Objects

Valuations

$2,000
Combinatorial Auctions

Bidders

Objects

Valuations

$10

$2,000
Combinatorial Auctions

Bidders

Objects

Valuations

$1,000

$10

$2,000
Combinatorial Auctions

Bidders: Red, Green, Blue

Objects: Star, Diamond, Triangle

Valuations: $1,000, $10, $2,000, $1,001
Combinatorial Auctions

Bidders

Objects

Valuations

- $10
- $1,000
- $2,000
- $1,001
- $3,000
Combinatorial Auctions

- $1,000
- $100
- $600
- $900

- $2,000
- $100
- $500
- $200

- $1,200
- $100
- $100
- $100

- $100
- $600
Combinatorial Auctions

- Pay $900
- Pay $1,000
- Pay $0
Combinatorial Auctions

- Pay $900
- Pay $1,000
- Pay $0

$1,000, $100, $600, $900, $1,300, $2,000, $0
Combinatorial Auctions

- Variants of the combinatorial auction problem, depending on the definition of a feasible allocation
Combinatorial Auctions

- Variants of the combinatorial auction problem, depending on the definition of a feasible allocation

<table>
<thead>
<tr>
<th>Problem</th>
<th>Feasibility condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA Problem</td>
<td>Disjoint sets</td>
</tr>
<tr>
<td>s-CA Problem</td>
<td>Disjoint, size at most s</td>
</tr>
<tr>
<td>Convex Bundles</td>
<td>Disjoint convex areas in plane, max aspect ratio R.</td>
</tr>
<tr>
<td>Unsplittable Flow</td>
<td>Possible to route flow between graph vertices, max capacity B</td>
</tr>
</tbody>
</table>
Combinatorial Auctions

- Variants of the combinatorial auction problem, depending on the definition of a feasible allocation

<table>
<thead>
<tr>
<th>Problem</th>
<th>Feasibility condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA Problem</td>
<td>Disjoint sets</td>
</tr>
<tr>
<td>s-CA Problem</td>
<td>Disjoint, size at most s</td>
</tr>
<tr>
<td>Convex Bundles</td>
<td>Disjoint convex areas in plane, max aspect ratio R.</td>
</tr>
<tr>
<td>Unsplittable Flow</td>
<td>Possible to route flow between graph vertices, max capacity B</td>
</tr>
</tbody>
</table>

- Many variants admit greedy approximation algorithms if we ignore game-theoretic issues.
Combinatorial Auctions

- Variants of the combinatorial auction problem, depending on the definition of a feasible allocation

<table>
<thead>
<tr>
<th>Problem</th>
<th>Feasibility condition</th>
<th>Approximation Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA Problem</td>
<td>Disjoint sets</td>
<td>$O(\sqrt{m})$</td>
</tr>
<tr>
<td>s-CA Problem</td>
<td>Disjoint, size at most $s$</td>
<td>$s+1$</td>
</tr>
<tr>
<td>Convex Bundles</td>
<td>Disjoint convex areas in plane, max aspect ratio $R.$</td>
<td>$O(R^{4/3})$</td>
</tr>
<tr>
<td>Unsplittable Flow</td>
<td>Possible to route flow between graph vertices, max capacity $B$</td>
<td>$O(B \frac{m^1}{B-1})$</td>
</tr>
</tbody>
</table>

- Many variants admit greedy approximation algorithms if we ignore game-theoretic issues.
Combinatorial Auctions

- Variants of the combinatorial auction problem, depending on the definition of a feasible allocation

<table>
<thead>
<tr>
<th>Problem</th>
<th>Feasibility condition</th>
<th>Approximation Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA Problem</td>
<td>Disjoint sets</td>
<td>$O(\sqrt{m})$</td>
</tr>
<tr>
<td>s-CA Problem</td>
<td>Disjoint, size at most $s$</td>
<td>$s+1$</td>
</tr>
<tr>
<td>Convex Bundles</td>
<td>Disjoint convex areas in plane, max aspect ratio $R$.</td>
<td>$O(R^{4/3})$</td>
</tr>
<tr>
<td>Unsplittable Flow</td>
<td>Possible to route flow between graph vertices, max capacity $B$</td>
<td>$O(B m^{1/(B-1)})$</td>
</tr>
</tbody>
</table>

- Many variants admit greedy approximation algorithms if we ignore game-theoretic issues.
- Can we implement a given approximation algorithm A as a mechanism for rational agents with (close to) the same approximation ratio?
Repeated Combinatorial Auctions

- Pay $0
- Pay $1,000

$1,000 $100
$100 $500
$2,000

Blue circle
Red circle
Repeated Combinatorial Auctions

- $1,000
- $100
- Pay $0
- Pay $1,000

- $2,000
- Pay $0
- Pay $1,000

- $100
- $500
- Pay $100
- Pay $100

- $2,500
- $100
- Pay $100

- $500
- $100
- Pay $500
- Pay $0

- $1,000
- Pay $500
Repeated Combinatorial Auctions

- \( t_i(S) \) - agent i's true value for object set S
- History of agent declarations: \( D = (d^1, \ldots, d^T) \)
- Each round, mechanism M maps \( d^t \) to allocation \( (S_1, S_2, \ldots, S_n) \) and payments \( p_1, \ldots, p_n \)
- Designer's objective: maximize average social welfare:

\[
SW_{avg}(D) = \frac{1}{T} \sum_t \sum_i t_i(S^t_i)
\]

- Agent's objective: maximize average utility:

\[
u_i(D) = \frac{1}{T} \sum_t (t_i(S^t_i) - p^t_i)
\]
Solution Concepts

- Standard behaviour models for repeated games imply solution concepts for designing our auction:
  1. Agents apply *regret-minimizing strategies*. [BHLR08]
  2. Agents apply *best-response strategies*. [GMV05]

- Given a $c$-approximation algorithm $A$, can we implement $A$ as a mechanism that obtains an approximation ratio close to $c$, on average over many rounds, given a model of agent behaviour?
Regret Minimization

- Given history $D = (d^0, d^1, ..., d^T)$, the *external regret* for agent $i$ is
  \[ \text{Max}_{d^*} \left( \frac{1}{T} \left( \sum_t u_i(d^*, d^t_{-i}) - \sum_t u_i(d^t) \right) \right) \]
- Agent $i$ *minimizes external regret* if his regret tends to 0 as $T \to \infty$. 
Regret Minimization

- Given history $D = (d^0, d^1, ..., d^T)$, the external regret for agent $i$ is
  \[ \text{Max}_{d^*} \left( \frac{1}{T} \sum_t u_i(d^*, d^t_{-i}) - \sum_t u_i(d^t) \right) \]
- Agent $i$ minimizes external regret if his regret tends to $0$ as $T \to \infty$. 
Regret Minimization

- Given history \( D = (d^0, d^1, ..., d^T) \), the external regret for agent \( i \) is
  \[
  \text{Max}_{d^*} \left( \frac{1}{T} \left( \sum_t u_i(d^*,d^t_{\{-i\}}) - \sum_t u_i(d^t) \right) \right)
  \]
- Agent \( i \) minimizes external regret if his regret tends to 0 as \( T \to \infty \).
Regret Minimization

- Given history $D = (d^0, d^1, ..., d^T)$, the external regret for agent $i$ is
  \[ \text{Max}_{d^*} \left( \frac{1}{T} \sum_t u_i(d^*, d^{t}_{-i}) - \sum_t u_i(d^t) \right) \]
- Agent $i$ minimizes external regret if his regret tends to 0 as $T \to \infty$. 
Regret Minimization

- Given history $D = (d^0, d^1, ..., d^T)$, the *external regret* for agent $i$ is
  \[ \text{Max}_{d^*} \left( \frac{1}{T} \left( \sum_t u_i(d^*, d^{t}_{-i}) - \sum_t u_i(d^t) \right) \right) \]
- Agent $i$ *minimizes external regret* if his regret tends to 0 as $T \to \infty$.
- Simple* algorithms (e.g. follow-the-leader) can be used to minimize external regret [cite].
- The *price of total anarchy* of mechanism $M$ is
  \[ \max \frac{SW_{\text{opt}}}{SW_{\text{avg}}}(D) \]
Regret Minimization

- Theorem: Any monotone, loser-independent c-approximation algorithm A can be implemented* as a mechanism M(A) with price of total anarchy (c+1).
Regret Minimization

- Theorem: Any **monotone, loser-independent** c-approximation algorithm A can be implemented* as a mechanism M(A) with price of total anarchy (c+1).

- Many greedy algorithms for variants of the CA problem are monotone and loser-independent.
Regret Minimization

- Theorem: Any **monotone**, **loser-independent** \( c \)-approximation algorithm \( A \) can be implemented* as a mechanism \( M(A) \) with price of total anarchy (\( c+1 \)).

- Many greedy algorithms for variants of the CA problem are monotone and loser-independent.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Feasibility condition</th>
<th>Approximation Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA Problem</td>
<td>Disjoint sets</td>
<td>( O(\sqrt{m}) )</td>
</tr>
<tr>
<td>s-CA Problem</td>
<td>Disjoint, size at most ( s )</td>
<td>( s+1 )</td>
</tr>
<tr>
<td>Convex Bundles</td>
<td>Disjoint convex areas in plane, max aspect ratio ( R ).</td>
<td>( O(R^{4/3}) )</td>
</tr>
<tr>
<td>Unsplittable Flow</td>
<td>Possible to route flow between graph vertices, max capacity ( B )</td>
<td>( O(B^{m^1/(B-1)}) )</td>
</tr>
</tbody>
</table>
Regret Minimization

- Theorem: Any **monotone, loser-independent** c-approximation algorithm A can be implemented* as a mechanism M(A) with price of total anarchy *(c+1)*.

- Many greedy algorithms for variants of the CA problem are monotone and loser-independent.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Feasibility condition</th>
<th>Approximation Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA Problem</td>
<td>Disjoint sets</td>
<td>$O(\sqrt{m})$</td>
</tr>
<tr>
<td>s-CA Problem</td>
<td>Disjoint, size at most $s$</td>
<td>$s+1$</td>
</tr>
<tr>
<td>Convex Bundles</td>
<td>Disjoint convex areas in plane, max aspect ratio $R$.</td>
<td>$O(R^{4/3})$</td>
</tr>
<tr>
<td>Unsplittable Flow</td>
<td>Possible to route flow between graph vertices, max capacity $B$.</td>
<td>$O(B^{m^1/(B-1)})$</td>
</tr>
</tbody>
</table>
Mechanism M(A)

Input: declared valuation profile d

1. SIMPLIFY(d)

2. S_1, ..., S_n <- A(d)

3. Return S_1, ..., S_n and charge critical prices
Mechanism M(A)

Input: declared valuation profile d
1. SIMPLIFY(d)
2. S_1, ..., S_n ← A(d)
3. Return S_1, ..., S_n and charge critical prices

SIMPLIFY:

$2,000 \rightarrow $100 \rightarrow $300 \rightarrow $500 \rightarrow $2,000
Mechanism M(A)

Input: declared valuation profile $d$

1. $SIMPLIFY(d)$

2. $S_1, ..., S_n \leftarrow A(d)$

3. Return $S_1, ..., S_n$ and charge critical prices

- Without loss of generality, each agent bids only on a single desired set each round. Furthermore, it is an undominated strategy to declare one's true value for the desired set.

- Since an undominated strategy is determined by the choice of a desired set, the size of an agent's strategy space is equal to the number of sets he desires.

- Agents can therefore apply standard algorithms for minimizing regret.
Byzantine Players

- Our argument is resilient to the presence of byzantine agents who do not necessarily minimize regret.

- Theorem: if agents do not over-bid, and some subset of the agents minimize regret, then mechanism $M(A)$ obtains a $(c+1)$ approximation to the optimal social welfare obtainable by the regret-minimizing agents.

- We think of byzantine players as not being savvy enough to understand how to bid “intelligently” in the auction.
Best Response

- Pay $400
- Pay $1,000
- Pay $0
Best Response

Pay $400

Pay $1,000

Pay $0
Best Response

- Pay $400
- Pay $1,000
- Pay $1,400
Best Response

Pay $400

Pay $1,000

Pay $1,400
Best Response

- Pay $1,200
- Pay $0
- Pay $1,400
Best Response

- On each round, an agent is chosen uniformly at random.
- That agent is free to change his declared valuation to the optimal, given the current declarations of the other bidders.
- The *price of sinking* of mechanism M is

  \[
  \max \frac{SW_{\text{opt}}}{E_D [SW_{\text{avg}} (D)]}
  \]

- where the max is over all types, and expectation is over histories corresponding to best-response dynamics.
Best Response

- For mechanism $M(A)$, best response dynamics might not converge to an equilibrium.
  - in fact, may include infinitely many states with approximation ratio $\Omega(m)$.
- A probabilistic argument demonstrates that these “bad states” cannot occur very often, for a modified mechanism $M'(A)$.
- Theorem: There is a mechanism for the general CA problem with $O(\sqrt{m})$ price of sinking.
Conclusions

- Applied repeated-game solution concepts to the problem of designing mechanisms for repeated combinatorial auctions.

- There is a general reduction from a broad class of approximation algorithms to approximation mechanisms, under the assumption that agents minimize external regret.

- For the general CA problem, it is possible to obtain an $O(\sqrt{m})$ approximation when agents apply best-response dynamics.
Future Work

- Does the general reduction used for regret-minimizing bidders also yield an $O(c)$ approximation for best-response bidders?
  - Does this approximation hold for an arbitrary mix of regret-minimizing and best-response bidders?
- Generalize to broader classes of algorithms.
- Generalize to broader classes of problems.
  - Problems that apply restrictions on the agents' valuation functions, e.g. submodular CAs.
Thank You