Combinatorial Auctions as Graphs

David Loker[1]

[1]School of Computer Science
University of Waterloo

Ontario Combinatorics Workshop, 2006
(joint work with C. Boucher)

Table of Contents

Motivation

- Auctions are used to allocate goods, resources, and services
  - Ebay, Amazon, ...
- FCC holds auctions for parts of electromagnetic spectrum
- Need an efficient way to determine who wins what

Combinatorial Auctions

- Need an efficient way to auction multiple items
  - Multiple single-item auctions and iterative auctions have economic inefficiencies
- What happens if I value a TV + DVD-Player more than I value a DVD-Player or TV individually?
- Combinatorial Auctions allow us to bid on **bundles** of items!

Combinatorial Auctions

- Need an efficient way to auction multiple items
  - Multiple single-item auctions and iterative auctions have economic inefficiencies
- What happens if I value a TV + DVD-Player more than I value a DVD-Player or TV individually?
- Combinatorial Auctions allow us to bid on **bundles** of items!

Combinatorial Auctions

Notation

- One seller
- Agents/Bidders: N, $|N| = n$
- Items: M, $|M| = m$
- Action Sets: $A = \{A_1, A_2, \ldots, A_n\}$
  - How an agent will bid on all $2^m - 1$ combinations of items
  - Otherwise known as **bidder valuations**
  - Denote bidder's value for $S \subseteq M$ as $b(S) \in \mathbb{Z}$
- Outcome: O
  - How items will be allocated and how much each agent will pay

Combinatorial Auctions

Notation

- One seller
- Agents/Bidders: N, $|N| = n$
- Items: M, $|M| = m$
- Action Sets: $A = \{A_1, A_2, \ldots, A_n\}$
  - How an agent will bid on all $2^m - 1$ combinations of items
  - Otherwise known as **bidder valuations**
  - Denote bidder's value for $S \subseteq M$ as $b(S) \in \mathbb{Z}$
- Outcome: O
  - How items will be allocated and how much each agent will pay

Combinatorial Auctions

Notation

- One seller
- Agents/Bidders: N, $|N| = n$
- Items: M, $|M| = m$
- Action Sets: $A = \{A_1, A_2, \ldots, A_n\}$
  - How an agent will bid on all $2^m - 1$ combinations of items
  - Otherwise known as **bidder valuations**
  - Denote bidder's value for $S \subseteq M$ as $b(S) \in \mathbb{Z}$
- Outcome: O
  - How items will be allocated and how much each agent will pay

Combinatorial Auctions

Notation

- One seller
- Agents/Bidders: N, $|N| = n$
- Items: M, $|M| = m$
- Action Sets: $A = \{A_1, A_2, \ldots, A_n\}$
  - How an agent will bid on all $2^m - 1$ combinations of items
  - Otherwise known as **bidder valuations**
  - Denote bidder's value for $S \subseteq M$ as $b(S) \in \mathbb{Z}$
- Outcome: O
  - How items will be allocated and how much each agent will pay

Bidder Valuations

Definition

An **atomic bid** for agent i is a pair (S, p), where $S \subseteq M$ and p is the maximum value that agent i is willing to pay for S.

Bidder Valuations

Definition

An **atomic bid** for agent i is a pair (S, p), where $S \subseteq M$ and p is the maximum value that agent i is willing to pay for S.

- Two atomic bids $(S_i, p_i), (S_j, p_j)$ are disjoint if $S_i \cap S_j = \emptyset$
- Recall: $b(S) = p$
- We can extrapolate information from this bid to say:
  - For all T such that $S \subseteq T$ we have $b(T) = p$
  - That's it!

Bidder Valuations

Problem

How do we communicate $2^m - 1$ atomic bids to the seller?!

Bidder Valuations

Problem

How do we communicate $2^m - 1$ atomic bids to the seller?!

- Non-zero bids ($b(S) \neq 0$) are **very** sparse over entire bid space
- Bidding language to condense bidder valuations
  - Concentrate on ways of representing important bids

OR Bidding Language

Definition

A **valid outcome** $\mathcal{X}$ is a set of bundles of items $\{S_1, S_2, \ldots, S_\ell\}$, where

1. $S_i \subseteq M$, $S_i \geq 1$ for all $1 \leq i \leq \ell$
2. $S_i \cap S_j = \emptyset$ for all $1 \leq i < j \leq \ell$

OR Bidding Language

Definition [Nisan 2005]

An OR bid is a set of atomic bids $\{(S_1, p_1), (S_2, p_2), \ldots, (S_q, p_q)\}$. It is implicit that the agent wishes to obtain any number of disjoint atomic bids such that the sum of the prices is maximized.

OR Bidding Language

Definition [Nisan 2005]

An OR bid is a set of atomic bids $\{(S_1, p_1), (S_2, p_2), \ldots, (S_q, p_q)\}$. It is implicit that the agent wishes to obtain any number of disjoint atomic bids such that the sum of the prices is maximized.

Definition [Nisan 2005]

$b(S)$ for $S \subseteq M$ is defined to be the maximum over all possible valid collections W of the value

$$\sum_{i \in W} p_i,$$

where W is valid if $\mathcal{X} = \{S_i \mid i \in W, S_i \subseteq S\}$ is a valid outcome

OR* Bidding Language

- Need to express exclusivity - I want bundle S or T, but not both

OR* Bidding Language

- Need to express exclusivity - I want bundle S or T, but not both

Definition [Nisan 2005]

A dummy item d is a "fake" item that has no intrinsic value. Denote the set of dummy items available for agent i as D_i.

- **Only** agent i may bid on items in D_i

OR* Bidding Language

Definition [Nisan 2005]

An OR* bid is a set of atomic bids $\{(S_1, p_1), (S_2, p_2), \ldots, (S_q, p_q)\}$ where $S_i \subseteq M \cup D_i$.

OR* Bidding Language

Definition [Nisan 2005]

An OR* bid is a set of atomic bids $\{(S_1, p_1), (S_2, p_2), \ldots, (S_q, p_q)\}$ where $S_i \subseteq M \cup D_i$.

Example

- Bids: $\{\text{Coffee}, \$2\}, \{\text{Tea}, \$1\}$
- I obviously don't want coffee and tea at the same time..
- Introduce dummy item d_1
- New bids: $\{\{\text{Coffee}, d_1\}, \$2\}, \{\{\text{Tea}, d_1\}, \$1\}$

OR* Bidding Language

Definition [Nisan 2005]

An OR* bid is a set of atomic bids $\{(S_1, p_1), (S_2, p_2), \ldots, (S_q, p_q)\}$ where $S_i \subseteq M \cup D_i$.

Example

- Bids: $\{\text{Coffee}, \$2\}, \{\text{Tea}, \$1\}$
- I obviously don't want coffee and tea at the same time..
- Introduce dummy item d_1
- New bids: $\{\{\text{Coffee}, d_1\}, \$2\}, \{\{\text{Tea}, d_1\}, \$1\}$

OR* Bidding Language

Definition [Nisan 2005]

An OR* bid is a set of atomic bids $\{(S_1, p_1), (S_2, p_2), \ldots, (S_q, p_q)\}$ where $S_i \subseteq M \cup D_i$.

Example

- Bids: $\{\text{Coffee}, \$2\}, \{\text{Tea}, \$1\}$
- I obviously don't want coffee and tea at the same time..
- Introduce dummy item d_1
- New bids: $\{\{\text{Coffee}, d_1\}, \$2\}, \{\{\text{Tea}, d_1\}, \$1\}$

Winner Determination

- We need to allocate items to agents
- Maximize seller profit
- Allocate an item to at most one agent
- Efficiency and optimality are important

Winner Determination

Definition

A **valid outcome** $\mathcal{X}$ is a set of bundles of items $\{S_1, S_2, \ldots, S_\ell\}$, where

1. $S_i \subseteq M$, $S_i \geq 1$ for all $1 \leq i \leq \ell$
2. $S_i \cap S_j = \emptyset$ for all $1 \leq i < j \leq \ell$

Winner Determination

Definition

A **valid outcome** $\mathcal{X}$ is a set of bundles of items $\{S_1, S_2, \ldots, S_\ell\}$, where

1. $S_i \subseteq M$, $S_i \geq 1$ for all $1 \leq i \leq \ell$
2. $S_i \cap S_j = \emptyset$ for all $1 \leq i < j \leq \ell$

Definition

An **exhaustive valid outcome** is a valid outcome where every item is included in exactly one subset.

Winner Determination

- Denote $b^*(S) = \max_{i \in N} b_i(S)$
  - Taking maximum bid for a given bundle - we can get rid of the rest

A solution to the winner determination problem is the following:

$$\max_{\mathcal{X}} \sum_{S \in \mathcal{X}} b^*(S)$$

Winner Determination

Definition

$$x_S = \begin{cases} 1 & \text{if the highest bid for } S \text{ is chosen to be winning,} \\ 0 & \text{otherwise} \end{cases}$$

A IP representation to the winner determination problem is the following:

$$\max_{\vec{x}} \sum_{S \subseteq M} b^*(S) \cdot x_S$$

$$\forall S \subseteq M : x_s \in \{0, 1\},$$

$$\forall i \in M : \sum_{S | i \in S} x_s \leq 1$$

Setup

Combinatorial Auction

- Agents N, Items M
- OR* bidding language: each agent i has OR* bid V_i
  - V_i consists of atomic bids $(S_j, p_j), S_j \subseteq M \cup D_i$ and $p_j > 0$
  - r_i is the number of atomic bids in V_i
  - M_i is the total number of items over all atomic bids (counting each item exactly once)
  - R is the number of unique atomic bids across all V_i

Setup

Combinatorial Auction

- Agents N, Items M
- OR* bidding language: each agent i has OR* bid V_i
  - V_i consists of atomic bids (S_j, p_j), $S_j \subseteq M \cup D_i$ and $p_j > 0$
  - r_i is the number of atomic bids in V_i
  - M_i is the total number of items over all atomic bids (counting each item exactly once)
  - R is the number of unique atomic bids across all V_i

Setup

Combinatorial Auction

- Agents N, Items M
- OR* bidding language: each agent i has OR* bid V_i
  - V_i consists of atomic bids $(S_j, p_j), S_j \subseteq M \cup D_i$ and $p_j > 0$
  - r_i is the number of atomic bids in V_i
  - M_i is the total number of items over all atomic bids (counting each item exactly once)
  - R is the number of unique atomic bids across all V_i

Setup

Combinatorial Auction

- Agents N, Items M
- OR* bidding language: each agent i has OR* bid V_i
  - V_i consists of atomic bids (S_j, p_j), $S_j \subseteq M \cup D_i$ and $p_j > 0$
  - r_i is the number of atomic bids in V_i
  - M_i is the total number of items over all atomic bids (counting each item exactly once)
  - R is the number of unique atomic bids across all V_i

Setup

Combinatorial Auction

- Agents N, Items M
- OR* bidding language: each agent i has OR* bid V_i
  - V_i consists of atomic bids $(S_j, p_j), S_j \subseteq M \cup D_i$ and $p_j > 0$
  - r_i is the number of atomic bids in V_i
  - M_i is the total number of items over all atomic bids (counting each item exactly once)
  - R is the number of unique atomic bids across all V_i

Construction Algorithm

- Construct a graph to represent all atomic bids
- Label vertices $\{i, S_j, p_j\}$, where:
  - (S_j, p_j) was an atomic bid for agent i
  - Also means agent i had highest bid for bundle S_j

Construction Algorithm

For agents $i = 1, 2, \dots, n$, do:

- For each atomic bid (S_j, p_j) in OR* bid V_i
  1. Iterate through all existing vertices $v_\ell = \{\ell, S_\ell, p_\ell\}$ and find v_ℓ such that $S_j = S_\ell$
     - If $p_\ell \geq p_j$, do nothing
     - Else set $v_\ell = \{i, S_j, p_j\}$
  2. No such v_ℓ was found so we create new vertex $v_j = \{i, S_j, p_j\}$
     - For each pre-existing vertex $u = \{u, S_u, p_u\}$ if $S_j \cap S_u \neq \emptyset$ then add undirected edge $\{v_j, u\}$

Example

- Two Agents
- Bids:

Agent 1	Agent 2
$\{TV, \$100\}$	$\{TV, \$125\}$
$\{\{TV,\ DVD\text{-}Player\}, \$130\}$	$\{\{DVD\text{-}Player, d_2\}, \$5\}$
$\{DVD\text{-}Player, \$10\}$	$\{\{VCR, d_2\}, \$2\}$
$\{\{Couch, d_1\}, \$50\}$	$\{\{Couch,\ Chair\}, \$60\}$
$\{\{Chair, d_1\}, \$10\}$	$\{Chair, \$15\}$

Example

- Two Agents
- Bids:

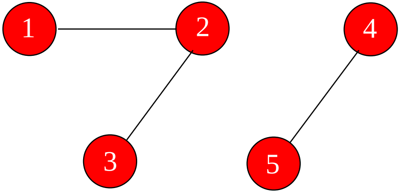
Agent 1	Agent 2
1 $\{TV, \$100\}$	6 $\{TV, \$125\}$
2 $\{\{TV,\ DVD\text{-}Player\}, \$130\}$	7 $\{\{DVD\text{-}Player, d_2\}, \$5\}$
3 $\{DVD\text{-}Player, \$10\}$	8 $\{\{VCR, d_2\}, \$2\}$
4 $\{\{Couch, d_1\}, \$50\}$	9 $\{\{Couch,\ Chair\}, \$60\}$
5 $\{\{Chair, d_1\}, \$10\}$	10 $\{Chair, \$15\}$

Example

Agent 1	Agent 2
1 $\{\text{TV}, \$100\}$	6 $\{\text{TV}, \$125\}$
2 $\{\{\text{TV, DVD-Player}\}, \$130\}$	7 $\{\{\text{DVD-Player}, d_2\}, \$5\}$
3 $\{\text{DVD-Player}, \$10\}$	8 $\{\{\text{VCR}, d_2\}, \$2\}$
4 $\{\{\text{Couch}, d_1\}, \$50\}$	9 $\{\{\text{Couch, Chair}\}, \$60\}$
5 $\{\{\text{Chair}, d_1\}, \$10\}$	10 $\{\text{Chair}, \$15\}$

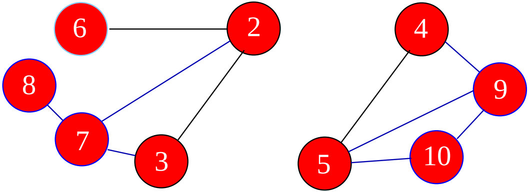
Example

Agent 1	Agent 2
1. $\{\text{TV}, \$100\}$	6. $\{\text{TV}, \$125\}$
2. $\{\{\text{TV, DVD-Player}\}, \$130\}$	7. $\{\{\text{DVD-Player}, d_2\}, \$5\}$
3. $\{\text{DVD-Player}, \$10\}$	8. $\{\{\text{VCR}, d_2\}, \$2\}$
4. $\{\{\text{Couch}, d_1\}, \$50\}$	9. $\{\{\text{Couch, Chair}\}, \$60\}$
5. $\{\{\text{Chair}, d_1\}, \$10\}$	10. $\{\text{Chair}, \$15\}$



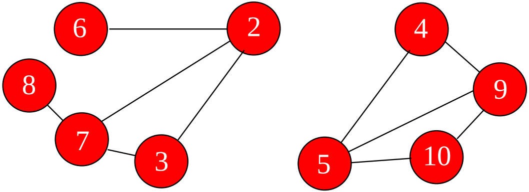
Example

Agent 1	Agent 2
1. $\{TV, \$100\}$	6. $\{TV, \$125\}$
2. $\{\{TV,\ DVD\text{-}Player\}, \$130\}$	7. $\{\{DVD\text{-}Player, d_2\}, \$5\}$
3. $\{DVD\text{-}Player, \$10\}$	8. $\{\{VCR, d_2\}, \$2\}$
4. $\{\{Couch, d_1\}, \$50\}$	9. $\{\{Couch,\ Chair\}, \$60\}$
5. $\{\{Chair, d_1\}, \$10\}$	10. $\{Chair, \$15\}$



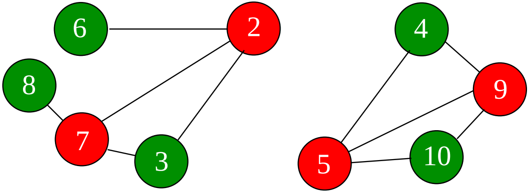
Example

Agent 1	Agent 2
1. $\{TV, \$100\}$	6. $\{TV, \$125\}$
2. $\{\{TV,\ DVD\text{-}Player\}, \$130\}$	7. $\{\{DVD\text{-}Player, d_2\}, \$5\}$
3. $\{DVD\text{-}Player, \$10\}$	8. $\{\{VCR, d_2\}, \$2\}$
4. $\{\{Couch, d_1\}, \$50\}$	9. $\{\{Couch,\ Chair\}, \$60\}$
5. $\{\{Chair, d_1\}, \$10\}$	10. $\{Chair, \$15\}$



Example

Agent 1	Agent 2
1 $\{TV, \$100\}$	6 $\{TV, \$125\}$
2 $\{\{TV,\ DVD\text{-}Player\}, \$130\}$	7 $\{\{DVD\text{-}Player, d_2\}, \$5\}$
3 $\{DVD\text{-}Player, \$10\}$	8 $\{\{VCR, d_2\}, \$2\}$
4 $\{\{Couch, d_1\}, \$50\}$	9 $\{\{Couch,\ Chair\}, \$60\}$
5 $\{\{Chair, d_1\}, \$10\}$	10 $\{Chair, \$15\}$



What do we have?

- Two vertices are adjacent if and only if their bundles are not disjoint
- The winner determination problem is equivalent to finding a maximum weighted independent set
  - NP-complete [Karp 1972]

Construction Algorithm

Time required for construction is:

$$\begin{aligned}\sum_{j=1}^{i}(r_j \cdot M_i \cdot M_j) &\leq \sum_{j=1}^{n}(r_j \cdot m \cdot M_j)\\ &\leq \sum_{j=1}^{n}(r_j \cdot m^2)\\ &\leq R \cdot m^2\end{aligned}$$

Approximation

- INDEPENDENT SET is extremely difficult to approximate [Hastad 1999]
- Slightly improved approximation for weighted case, but still difficult [Halldórsson 2000]
- Reduction from INDEPENDENT SET to WEIGHTED INDEPENDET SET shows approximation remains difficult

Approximation is hard!

Graph Size

- Number of vertices exactly the number of unique atomic bids, R
  - Naive algorithm requires $O(R^2 \cdot 2^R)$ time for max weighted independent set
- Restricting number of atomic bids per agent
  - Bound number of atomic bids per agent to B
  - R is at most $n \cdot B$
  - Naive algorithm now requires $O((n \cdot B)^2 \cdot 2^{(n \cdot B)})$ time
- Number of edges exactly the number of pairwise non-disjoint atomic bids

Conclusions

- Mapping from structure of combinatorial auctions to graphs
  - Polynomial-time construction
  - Problem is computationally difficult to solve
- Approximation is difficult so might as well stick to exact algorithms
- Combinatorial auction structure impacts the graph
  - The opposite is also true: restricting the graph class affects the combinatorial auction

Future Work

- Graph classes for which WEIGHTED INDEPENDENT SET is polynomial
    1. Perfect graphs and all subclasses of perfect graphs
    2. Circular arc graphs
    3. Trees
    4. Grid graphs

Future Work

- Graph classes for which WEIGHTED INDEPENDENT SET is polynomial
    1. Perfect graphs and all subclasses of perfect graphs
    2. Circular arc graphs
    3. Trees
    4. Grid graphs

  What happens to the structure of the combinatorial auction for these graph classes?

Magnús M. Halldórsson,
"Approximation of weighted independent set and hereditary subset problems,"
In Computing and Combinatorics, J. Graph Algorithms Appl. 4 (1) (2000) 1-16.

J. Hastad,
"Clique is hard to approximate within $n^{1-\epsilon}$,"
Acta Math., 182(1):105–142,
1999.

R. M. Karp,
"Reducibility among combinatorial problems,"
in: R. E. Miller, J. W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press,
New York, 1972, pp. 85-103.

Noam Nisan,
"Bidding Languages for Combinatorial Auctions,"
In Combinatorial Auctions by Cramton, Shoham and Steinberg (eds.),
MIT Press, 2006.

T. Sandholm,
"Algorithm for Optimal Winner Determination in Combinatorial Auctions,"
Artificial Intelligence, 135, 1-54,
2002.