Lecture #9: Combinatorial Auctions

Prof. Dr. Sven Seuken
29.3.2012

Housekeeping

- Questions? Concerns?
- Comprehension Questions
  - Grades
  - Optimal solutions
  - Late submissions
- Reading for next Monday
  - 8.2-8.4: Mechanism Design II (already online!)
  - Interlude (still to come…)

Outline

1. Recap of Mechanism Design
2. Today’s topic: Combinatorial Auctions

Recap of Mechanism Design

- When do we have a mechanism design problem?
- What does a mechanism do?
- Desirable properties?
- The Groves mechanism
- The VCG mechanism
- Positive properties of VCG
- Problems of VCG?

Today's Topic: Combinatorial Auctions

Mechanism Design vs. Combinatorial Auctions

- What is so special about combinatorial auctions?
  - Selling/buying multiple items simultaneously
  - Items may be different or the same
  - Bidders valuations can be:
    - Additive $v_i(A) + v_i(B) = v_i(A \cup B)$
    - Superadditive: $v_i(A) + v_i(B) \leq v_i(A \cup B)$
    - Subadditive: $v_i(A) + v_i(B) \geq v_i(A \cup B)$

Computational Challenges

- Bidding Language
  - m items → $2^m - 1$ possible bundles to bid on
  - Problem for bidders
  - Problem for auctioneer as well!

- Winner Determination Problem
  - Complexity: NP-hard
  - Let b denote the number of submitted bids
  - Only algorithms known that take $2^b - 1$ steps I the worst case to find the optimal allocation
  - Even if each of the n bidders only submits one bid: worst-case: $O(2^n)$

Bidding Languages

- OR
- XOR
- OR*

(T_1, w_1) OR (T_2, w_2) OR (T_3, w_2)

(T_1, w_1) XOR (T_2, w_2)

, A, B

, 50, 60, 150

$B_1(A, 50)$ XOR $B_2(B, 60)$

$B_3(AB, 150)$

OR*

- Set of items G
- Set of dummy items for each bidder i: G_i

$$b_i^{\text{or}*} = (T_1, w_1) \text{ OR } (T_2, w_2) \text{ OR } \ldots \text{ OR } (T_\ell, w_\ell)$$

where bundles $T_k \subseteq G \cup G_i$ for $k \in \{1, \ldots, \ell\}$.

- OR* is: ({A,a},50) OR ({B,a},150)
  - Fully expressive (like XOR, unlike OR)
  - Exponentially more compact than XOR

Example

- A: 60; B: 80; AB: 200;
- ~~C: 70 (if I don't get A); C: 120 (if I also get A)~~

Computation: Winner Determination

- Problem: "Find the set of bids to accept that maximizes the total value"

→This is an optimization problem

- Decision version: "Is there an allocation with value at least W"?
- Call this the "COMBINATORIAL-WINNER-DETERMINATION PROBLEM"
- Decision problem is NP-hard (reduction from independent set)

→ Optimization problem is NP-hard

Three possible solutions

1. Optimal algorithms that are fast in practice
2. Identify special cases (tractable) that can be solved in worst-case polynomial time
3. Identify fast approximation algorithms

Optimal Algorithm: Integer Programming

- We can formulate the winner determination problem as an IP
- Here, let's consider "single-minded bidders":

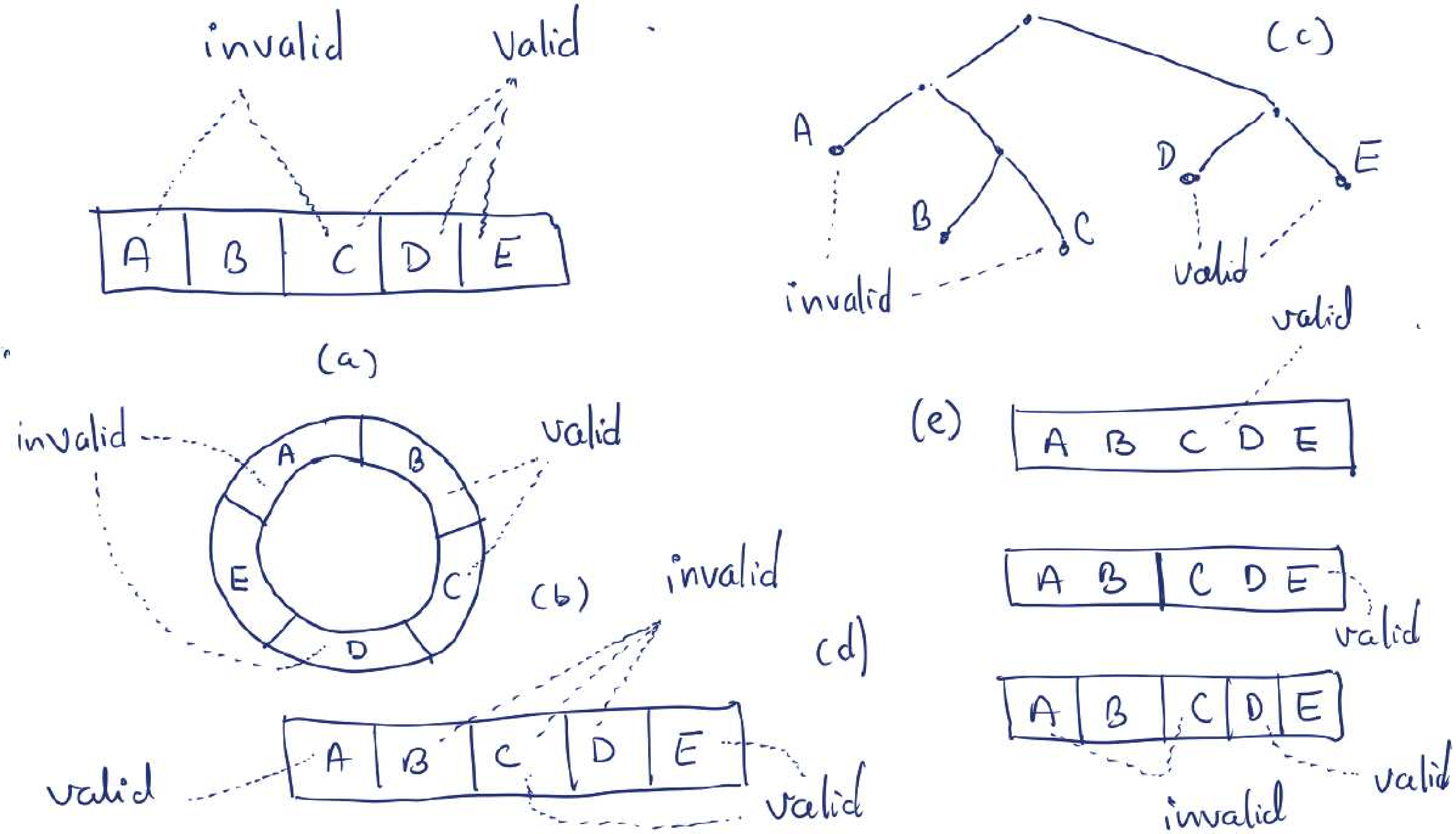
the bids are $(T_1, w_1), \ldots, (T_n, w_n)$ from agents 1 to n

$$\max_{x_1,\ldots,x_n} \sum_{i=1}^{n} w_i x_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} a_{ij} x_i \leq 1, \quad \forall j \in G$$

$$x_i \in \{0,1\}, \quad \forall i \in N$$

Special (tractable) Cases



Approximation Algorithms

Definition 10.2. *Algorithm A is a c-approximation if $\frac{W^*(b)}{W^A(b)} \le c$ for all bids $b = (b_1, \ldots, b_n)$.*

- A greedy algorithm (runs in polynomial time):

Definition 10.3 (Greedy algorithm for single-minded CAP)**.**

Input: *single-minded bids $(T_1, w_1), \ldots, (T_n, w_n)$.*

1. *Order the bids such that $w_1/\sqrt{|T_1|} \ge w_2/\sqrt{|T_2|} \ge \ldots \ge w_n/\sqrt{|T_n|}$*
2. *$S = \emptyset$.*
3. *For $i = 1$ to n:*

   If $(T_i \cap S = \emptyset)$ then $x_i = 1$ and $S := S \cup T_i$.

   Else, $x_i = 0$.

Output: *$x_1, \ldots, x_n$.*

Incentives: The VCG Mechanism

1. Every agent submits a bid b_i in a bidding language defined by the auctioneer. Let $b_i(S) \in \Re$ denote the bid value for some bundle $S \subseteq G$.
2. Compute x^*, the solution to the winner determination problem with all bids, and for every i who is allocated some items, compute x^{-i}, the solution to the winner determination problem without the bid from that agent.
3. Allocate the items according to x^*. Collect as payment $p_i = \sum_{j \neq i} b_j(x_j^{-i}) - \sum_{j \neq i} b_j(x_j^*)$ from each agent allocated some items in x^*.

Properties of VCG applied to CAs

- Efficient
- Truthful (Dominant-strategy incentive compatible)
- Individually rational
- No-deficit

→ What more could we want?

The VCG Mechanism

- Idea: each agent's payment is the externality it imposes with its presence on each other agent.

Definition 8.4 (VCG mechanism). *The Vickrey-Clarke-Groves (VCG) mechanism (x, t) on alternatives A, defines*

- *a choice rule* $x(\hat{v}) \in \arg\max_{a \in A} \sum_{i \in N} \hat{v}_i(a)$,
- *a payment rule* t *such that*

$$t_i(\hat{v}) = \sum_{j \in N, j \neq i} \hat{v}_j(a^{-i}) - \sum_{j \in N, j \neq i} \hat{v}_j(a^*), \tag{8.7}$$

with $a^* = x(\hat{v})$ *and* $a^{-i} \in \arg\max_{a \in A^{-i}} \sum_{j \in N, j \neq i} \hat{v}_j(a)$,

given reports $\hat{v} = (\hat{v}_1, \ldots, \hat{v}_n) \in V^n$, *and with* $\hat{v}_{-i} = (\hat{v}_1, \ldots, \hat{v}_{i-1}, \hat{v}_{i+1}, \ldots, \hat{v}_n)$.

Public Projects and VCG

- Where should we locate the hospital?
- Four locations: A, B*, C, D
- Agent preferences:
  - Agent 1: \$140, \$90, \$20, \$10
  - Agent 2: \$40, \$200, \$30, \$40
  - Agent 3: \$80, \$10, \$190, \$90
- Outcome? Payments?

$a^* = B$

$a^{-1} = C$

$a^{-2} = A$

$a^{-3} = B$

$t_1 = 220 - 210 = \$10$

$t_2 = 220 - 100 = 120$

$t_3 = 290 - 290 = 0$

✓

Problems when Applying VCG to CAs

- **Failure of revenue monotonicity**:
  adding a bidder can lower the revenue
- **Collusion by losers:**
  Two losing bidders can collude, submit false bids, possibly win and make no payments
- **False-name manipulations:**
  one bidder can pretend to be two bidders, submit fake bids, possibly win, and make no payments
- **Not envy free:**
  one bidder i would rather obtain the good of bidder j and pay bidder j's price

Construct examples!

- Using two items A and B
- Using at most 3 bidders

$$t_1 = x - 0 = x$$

$$t_2' = x - x = 0$$

VCG + Greedy Allocation Rule

- Idea (to get around NP-hardness result):
  - Use greedy algorithm to solve winner determination
  - Use VCG payment rule
- Problem (example in the lecture notes):
  - The auction may not be individually rational
  - The auction may not be truthful

Truthful Greedy Auction for Single-minded CA

- Idea: use *critical value payment*

Definition 10.4 (Greedy auction for the single-minded CA).

Input: *single-minded bids* $(T_1, w_1), \ldots, (T_n, w_n)$.

1. *Use the greedy algorithm with ordering based on score* $w_i/\sqrt{|T_i|}$ *to determine the allocation* $x_1, \ldots, x_n$.
2. *For bid* i*, let* L_i *denote the set of bids that are ranked below* i *according to the score* and *do not demand any items that are already allocated to agents ranked higher than* i*. These are bids that could still be allocated when considering* i*. For any agent allocated (with* $x_i = 1$*), collect as payment* p_i *the minimal bid* w'_i *such that* $\frac{w'_i}{\sqrt{|T_i|}} \geq \max_{j \in L_i, T_j \cap T_i \neq \emptyset} \frac{w_j}{\sqrt{|T_j|}}$.

Output: $x_1, \ldots, x_n$. *Payments* $p_1, \ldots, p_n$.