Auctions, Auction Theory, and Hard Computational Problems in Auctions

Kevin Leyton-Brown

This talk is adapted from slides by Yoav Shoham, Moshe Tenenholtz and Michael Wellman

Overview

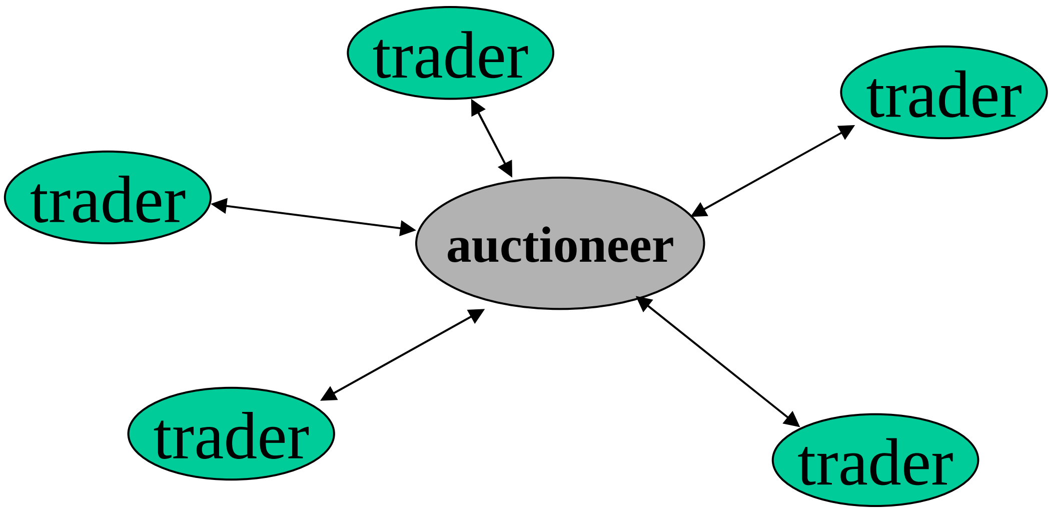
- **Auctions**
- Single dimensional auctions: taxonomy
- Game Theoretic Foundations
- Auction Theory
- Combinatorial Auctions
- Hard Computational Problems
- A Test Suite for Combinatorial Auctions

Auctions: Definition

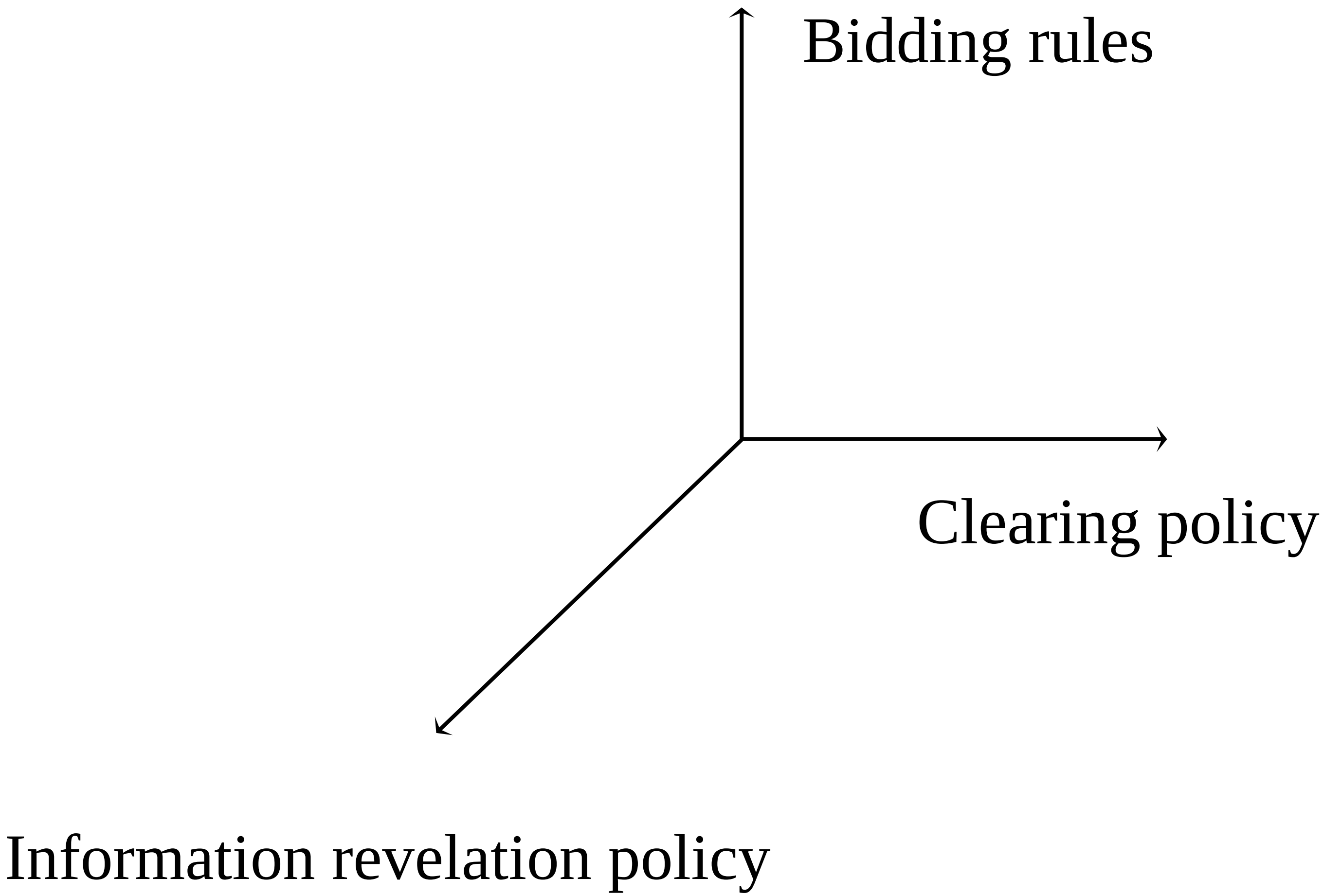
- There’s a lot more to auctions than the classic “going… going… gone!” mechanism that first jumps to mind
- An auction is any negotiation mechanism that is:
  - Mediated
    - impartial auctioneer
  - Well-specified
    - runs according to explicit rules
  - Market-based
    - determines an exchange in terms of standard currency

Auctioneer

- Receives Bids
- Disseminates Information
- Arranges trades (clear market)



Auction Dimensions



Bidding Rules

- Who can bid, when
- What is form of bid
- Restrictions on offers, as a function of
  - Trader’s own previous bid
  - Auction state (everyone’s bids)
  - Eligibility (e.g., financial)
  - …
- Expiration, withdrawal, replacement

Information Revelation

- When to reveal information
- What information
- To whom

Sealed bid — Open outcry

Clearing Policy

- *Clear*: Translates offers into agreed trades, according to specified rules.

- Policy choices:
  - When to clear:
    - at specified intervals
    - on each bid
    - on inactivity
  - Who gets what (*allocation*)
  - At what prices (*payment*)

Overview

- Auctions
- **Single dimensional auctions: taxonomy**
- Game Theoretic Foundations
- Auction Theory
- Combinatorial Auctions
- Hard Computational Problems
- A Test Suite for Combinatorial Auctions

Single-dimensional auctions

1. one sided
   1.1 English
   1.2 Dutch
   1.3 Japanese
   1.4 Sealed bid
2. two sided
   2.1 Continuous double auction (CDA)
   2.2 Call market (periodic clear)

Single-unit English auction

- Bidders call ascending prices
- Auction ends:
  - at a fixed time
  - when no more bids
  - a combination of these
- Highest bidder pays his bid

Multi-unit English auctions

- Different pricing schemes
  - lowest accepted (uniform pricing, sometimes called “Dutch”)
  - highest rejected (uniform pricing, GVA)
  - pay-your-bid (discriminatory pricing)
- Different tie-breaking rules
  - quantity
  - time bid was placed
- Different restrictions on partial quantities
  - allocate smaller quantities at same price-per-unit
  - all-or-nothing
    - finding the winners is NP-Hard: weighted knapsack problem

Dutch (“descending clock”) auction

- Auctioneer calls out descending prices
- First bidder to jump in gets the good at that price

- With multiple units: bidders shout out a quantity rather than “mine”. The clock can continue to drop, or reset to any value.

Japanese auction

- Auctioneer calls out ascending prices
- Bidders are initially “in”, and drop out (irrevocably) at certain prices
- Last guy standing gets it at that price

- Multi-unit version: bidders call out quantities rather than simple “in” or “out”, and the quantities decrease between rounds. Auction ends when supply meets or exceeds demand. (Note: what happens if exceeds?)

Sealed bid auctions

- Each bidder submits a sealed bid
- (Usually) highest bid wins
- Price is
  - first price
  - second price
  - k’th price
- Note: Can still reveal interesting information during auction
- In multiple units: similar pricing options as in English

Reverse (procurement) auctions

- English descending
- Dutch ascending
- Japanese descending

Two-sided (double) auctions

- Continuous double auction (CDA)
  - every new order is matched immediately if possible
  - otherwise, or remainder, is put on the order book
  - NASDAQ-like

- Call (“periodic clear”) market
  - orders are matched periodically
  - Arizona stock exchange (AZX) -like

Intuitive comparison of the basic four auctions

	English	Dutch	Japanese	Sealed Bid
Regret	no	yes	no	1st: yes 2nd: no
Duration	#bidders, increment	starting price, clock speed	#bidders, increment	fixed
Information Revealed	2nd-highest val; bounds on others	winner's bid	all val's but winner's	none
Jump bids	yes	n/a	no	n/a
Price Discovery	yes	no	yes	no

What about agents' strategies in each auction type?

Overview

- Auctions
- Single dimensional auctions: taxonomy
- **Game Theoretic Foundations**
- Auction Theory
- Combinatorial Auctions
- Hard Computational Problems
- A Test Suite for Combinatorial Auctions

Static Games in Strategic Form

- A (two-player) game in strategic form is a tuple <S_1, S_2, U_1, U_2> where S_1 is a set of strategies available to player *i*, and U_i: $S_1 \times S_2 \rightarrow R$ is a utility/payoff function for player *i*.

- Usually depicted through a *payoff matrix*

Examples of game in strategic form

- Prisoners' Dilemma (PD)

1,1	3,0
0,3	2,2

- The coordination game

1,1	0,0
0,0	1,1

- Matching pennies

1,-1	-1,1
-1,1	1,-1

A solution concept: the Nash equilibrium

- A pair of strategies (s,t) is a Nash equilibrium if

  $$\forall(s' \in S_1, t' \in S_2),\ U_1(s', t) \le U_1(s, t),\ U_2(s, t') \le U_2(s, t)$$

1,1	3,0
0,3	2,2

1,1	0,0
0,0	1,1

1,-1	-1,1
-1,1	1,-1

Strategy Types

- Dominant Strategy
  - Best to do no matter what others do
  - e.g., prisoner's dilemma

- Mixed Strategy
  - Mixed strategies of player i: probability distributions on S_i.
  - Nash equilibrium is easily generalized to mixed strategies
    - rather than look at payoff, look at expected payoff.
  - **Thm.** There always exists a Nash equilibrium in mixed strategies. The result holds also for the case of n players.

Auctions as games, unsuccessful attempt

- Consider a 1st-price auction
  - N bidders, valuations $v_i > v_2 > \ldots > v_n$
- Unsuccessful game-theoretic model:
  - Strategies: the bids b_i
  - Payoffs: $v_i - b_i$ for winner, zero otherwise
  - In all equilibria the agent with v_1 wins; there are many such equilibria
  - BUT: this implicitly assumes that the valuations are common knowledge (that is, the game is known).
    - then what's the point of having an auction?

Uncertainty: Bayesian Games

- Represent games in which agents have partial information about one another
- Bayesian games add this ingredient in one of two equivalent ways:
  - Posit a set of games, with each player having a belief (probability) about which is being played
  - Posit a single game with an added player, Nature, with each player receiving some signal about Nature’s move.
- Bayes-Nash equilibrium is a generalization of Nash equilibrium to this setting.

Auction as a Bayesian game

- Players: bidders + Nature
- Nature chooses valuations for each agent
- Each agent’s signal is his own valuation.
- Agent’s strategy: mapping from valuation to bidding strategy

Agents care about utility, not valuation

- Actions are really lotteries, so you must compare expected utility rather than utility.
- Risk attitude speak about the shape of the utility function
  - linear/concave/convex utility function refers to risk-neutrality/risk-aversion/risk-seeking, respectively.
- The types of utility functions, and the associated risk attitudes of agents, are among the most important concepts in Bayesian games, and in particular in auctions. Most theoretical results about auction are sensitive to the risk attitude of the bidders.

Overview

- Auctions
- Single dimensional auctions: taxonomy
- Game Theoretic Foundations
- **Auction Theory**
- Combinatorial Auctions
- Hard Computational Problems
- A Test Suite for Combinatorial Auctions

Two yardsticks for good auction design

- Revenue: The seller should extract the highest possible price
- Efficiency: The buyer with the highest valuation should get the good
  - usually achieved by ensuring “incentive compatibility”: bidders are induced to bid their true valuation
  - maximizing over those bids ensures efficiency.
- The two are sometimes but not always aligned

Direct mechanisms and incentive compatibility

- In a direct mechanism you simply announce your valuation
- The auction is incentive compatible if it’s in your best interest not to lie about your true valuation
- Example: 2nd price (“Vickrey”) auction
- Another example: the generalized Vickrey auction (GVA)

The revelation principle

- You can transform any auction into an “equivalent” one which is direct and incentive compatible
- “Rather than lie, the mechanism will lie for you”
- Example: Assume two bidders, with valuations drawn uniformly from a fixed interval (plus other assumptions). The optimal strategy is to bid 1/2 your true value. But if the rule is changed so that the winner only pays half his bid, it is optimal to bid your true value.

Independent Private Value (IPV) versus Common Value (CV)

- In a CV model agents' valuations are correlated.
  - the revelation of information during the auction plays a significant role
- In the IPV model they are independent.

Connections

- Dutch = 1^{st}-price sealed bid
- English ~ Japanese
- English = 2^{nd}-price sealed bid under IPV

The Revenue Equivalence Theorem

- In all auctions for k units with the following properties
  - Buyers are risk neutral
  - IPV, with values independently and identically distributed over [a,b] (technicality – distribution must be atomless)
  - Each bidder demands at most 1 unit
  - Auction allocates the units to the k highest bids
  - The bidder with the lowest valuation has a surplus of 0
- a buyer with a given valuation will will make the same expected payment, and therefore
- all such auctions have the same expected revenue

Overview

- Auctions
- Single dimensional auctions: taxonomy
- Game Theoretic Foundations
- Auction Theory
- **Combinatorial Auctions**
- Hard Computational Problems
- A Test Suite for Combinatorial Auctions

What are combinatorial auctions (CAs)

- Multiple goods are auctioned simultaneously
- Each bid may claim any combination of goods
- A typical combination: a bundle (“I bid $100 for the TV, VCR and couch”)
- More complex combinations are possible

Motivation: complementarity and substitutability

- Complementary goods have a superadditive utility function:
  - V({a,b}) > V({a}) + V({b})
  - In the extreme, V({a,b}) >>0 but V({a}) = V({b}) = 0
  - Example: different segments of a flight

- Substitutable goods have a subadditive utility function:
  - V({a,b}) < V({a}) + V({b})
  - In the extreme, V({a,b}) = MAX[V({a}) , V({b})]
  - Examples: a United ticket and a Delta ticket

Unstructured bidding is impractical

- Bidder sends his entire valuation function (over all possible allocations) to auctioneer.
  - Problem: Exponential size

- Bidder sends his valuation as a computer program (applet)
  - Problem: requires exponential access by any auctioneer algorithm

Often, valuations have structure

- "Classic":
  - (take-off right) AND (landing right)
  - (frequency A) XOR (frequency B)
- Online Computational resources:
  - Links: ((a--b) AND (b--c)) XOR ((a--d) AND (d--c))
  - (disk size > 10G) AND (speed >1M/sec)
- E-commerce:
  - chair AND sofa -- of matching colors
  - (machine A for 2 hours) AND (machine B for 1 hour)

Bidding Language Requirements

- Expressiveness
  - Must be expressive enough to represent every possible valuation.
  - Representation should not be too long
- Simplicity
  - Easy for humans to understand
  - Easy for auctioneer algorithms to handle

AND, OR, and XOR bids

- {left-sock, right-sock}:10
- {blue-shirt}:8 XOR {red-shirt}:7
- {stamp-A}:6 OR {stamp-B}:8

General OR bids and XOR bids

- {a,b}:7 OR {d,e}:8 OR {a,c}:4
  - {a}=0, {a, b}=7, {a, c}=4, {a, b, c}=7, {a, b, d, e}=15
  - Can only express valuations with no substitutabilities.

- {a,b}:7 XOR {d,e}:8 XOR {a,c}:4
  - {a}=0, {a, b}=7, {a, c}=4, {a, b, c}=7, {a, b, d, e}=8
  - Can express any valuation
  - Requires exponential size to represent

    {a}:1 OR {b}:1 OR … OR {z}:1

OR of XORs example

{couch}:7 XOR {chair}:5

OR

{TV, VCR}:8 XOR {Book}:3

Relative expressive power of different formats

- OR bids can represent valuations without substitutabilities
- XOR bids can represent all valuations
- Additive valuations can be represented linearly with OR bids, but only exponentially with XOR bids

The expressive power of ‘dummy’ goods

- Transform “$10 for a XOR (b and c)” into two bids: “$10 for a and x” and “$10 for b, c and x”; x is the dummy good.
  - The idea: any decent CA will never grant the two bids
- With dummy goods, OR can represent any function
- How many dummy goods are needed?
  - In the worst case, exponentially many
    - Example: the Majority valuation
  - OR-of-XORs: s, where s is the number of atomic bids in the input
  - XOR-of-ORs: s^2

Auction theory applied to CA's

- We've examined the technical issues behind *how* agents will bid
- However, *what* will they bid?
- How can we change the CA mechanism to influence agents' strategic behavior?

- Most naïve CA mechanism:
  - agents submit bids for bundles
  - auctioneer computes revenue-maximizing allocation
  - bidders pay the amounts of their bids

The Naïve CA is not incentive compatible

- Naïve CA: Given a set of bids on bundles, auctioneer finds a subset containing non-conflicting bids that maximizes revenue, and charges each winning bidder his bid
- This is not incentive compatible, and thus not (economically) efficient
- Example:
  - v1(x,y)=200, v1(x,¬y)=v1(¬x,y)=0
  - v1(x,y)=100, v2(x, ¬y)=v2(¬x,y)=75
  - Bidder 1 has incentive to “lie” and bid less
    - in this example he would win with a bid of $101
  - If bidder 2 lies then bidder 1 has an incentive to lie even more

Lessons from the single dimensional case

- 1st-price sealed bid auction is not incentive compatible (in equilibrium, it pays to "shave" a bit off your true value)

- 2nd-price sealed bid ("Vickrey") auction is incentive compatible

- Can we pull off the same trick here?

The Generalized Vickrey Auction (GVA)* is incentive compatible

- The Generalized Vickrey Auction charges each bidder their social cost
- Example:
  - Red bids 10 for {a}, Green bids 19 for {a,b}, Blue bids 8 for {b}
  - Naïve: Green gets {a,b} and pays 19
  - GVA: Green gets {a,b} and pays 18 (10 due to Red, 8 due to Blue)

* aka the Vickrey-Clarke-Groves (VCG) mechanism

Formal definition of GVA

- Each i reports a utility function $r_i(\cdot)$ possibly different from $u_i(\cdot)$
- The center calculates (x^*) which maximizes sum of r_is
- The center calculates $(\hat{x}_{-i})$ which maximizes sum of r_is without i
- Agent i receives (x_i^*) and also a payment of

$$\sum_{j \neq i} r_j(x^*) - \sum_{j \neq i} r_j(\hat{x}_{\sim i})$$

- Thus agent i's utility is

$$u_i(x^*) + \sum_{j \neq i} r_j(x^*) - \sum_{j \neq i} r_j(\hat{x}_{\sim i})$$

Other remarks about GVA

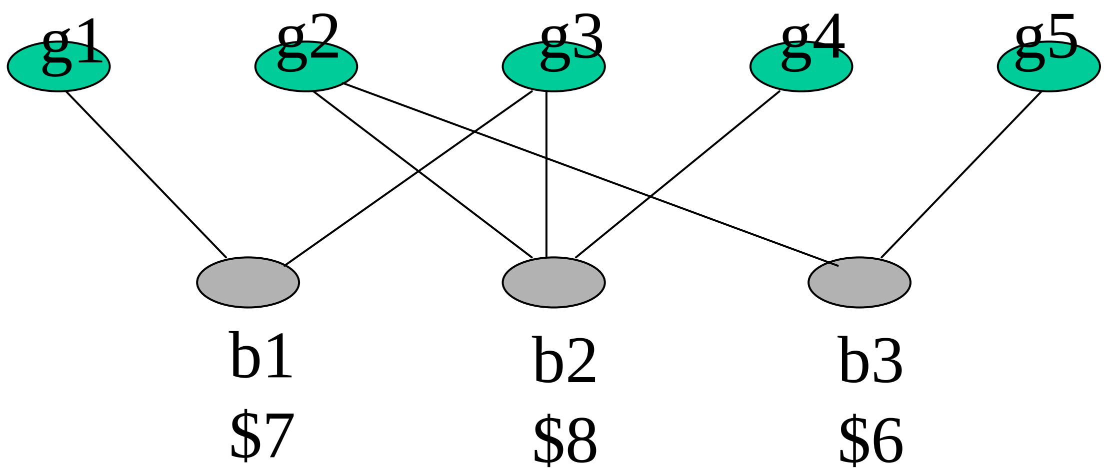
- Applies not only to auctions as we know them, but to general resources allocation problems
  - When “externalities” exist
  - E.g, with public goods
- Cannot simultaneously guarantee
  - Participation
  - Incentive compatibility
  - Budget balance
- Not collusion-proof

Overview

- Auctions
- Single dimensional auctions: taxonomy
- Game Theoretic Foundations
- Auction Theory
- Combinatorial Auctions
- **Hard Computational Problems**
- A Test Suite for Combinatorial Auctions

The optimization problem of CAs

- "Given a set of bids on bundles, find a subset containing non-conflicting bids that maximizes revenue"
- Performed once by the naïve method, n+1 times by GVA
- Requires exponential time in the number of goods and bids (assuming they are polynomially related)



What's known about the problem?

- Weighted set packing: NP-Hard
- Uniform approximation is equally hard
- Best known polynomial approx. bound is $1/\sqrt{k}$, k is # goods
- Approaches
  - Incomplete heuristic methods
    - however, GVA not incentive compatible if we use these
  - Complete algorithms
    - tractable special cases
    - complete heuristic methods
- How to test these algorithms? The need for a test suite

Weighted Set Packing as an Integer Program

- n items -- indexed by i (some may be dummies)
- m atomic bids: (S_j, p_j) (maybe multiple ones from same bidder)
- Goal: optimize social efficiency

$$\begin{aligned}
&\textit{Maximize} && \sum_{j=1}^{m} x_j p_j \\
&\textit{Subject} && \textit{to}: \\
&\sum_{i \in S_j} x_j \le 1 && \forall i \\
&x_j \in \{0,1\} && \forall j
\end{aligned}$$

- Problem: IP is hard

Linear Programming Relaxation of the IP

$$\textit{Maximize} \quad \sum_{j=1}^{m} x_j p_j$$

$$\textit{Subject to}:$$

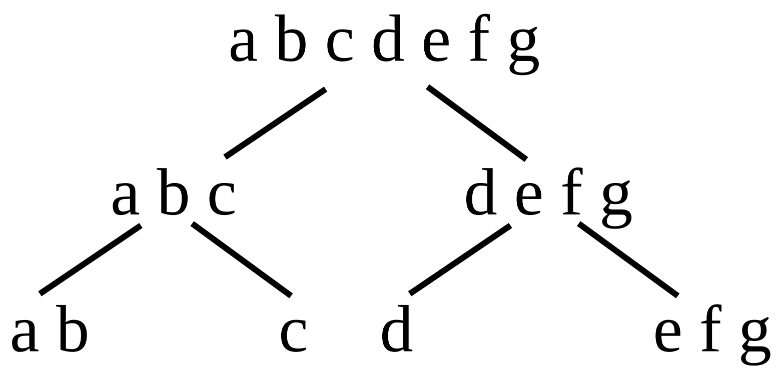
$$\sum_{i \in S_j} x_j \le 1 \quad \forall i$$

$$x_j \ge 0 \quad \forall j$$

- Good news: LP is easy
- Bad news: “fractional” allocations
  - x_j specifies what fraction of bid *j* is obtained.
- If we’re lucky, the solution will be integer anyway

When do we get lucky?

- Tree structured bundles:



- Continguous single-dimensional goods ("consecutive ones"); e.g., time intervals
- Bundles of size at most 2
- A general condition: Total Unimodular (TU) matrices

State of the art

- Recent years have seen an explosion of specialized search algorithms for CAs
- Complete methods guarantee optimal results, but not quick convergence. On test cases the algorithms scale to about 100 goods and 10000’s of bids.
- Incomplete, greedy-search methods sometimes perform an order of magnitude faster
- Very recent results on the multi-unit case
- CPLEX 7.0 holding its own…
- A major challenge: testing the algorithms (CATS)

Hard problems: Summary

- Multi-unit English Auction
  - weighted knapsack problem
- Single-unit combinatorial auctions
  - weighted set packing problem
- Combinatorial auctions for procurement
  - weighted set cover problem instead of set packing
- Multi-unit combinatorial auctions
- GVA: solve one of the above problems n+1 times
  - where n is the number of winners
  - note: the n+1 problems are closely related

Overview

- Auctions
- Single dimensional auctions: taxonomy
- Game Theoretic Foundations
- Auction Theory
- Combinatorial Auctions
- Hard Computational Problems
- **A Test Suite for Combinatorial Auctions**

Testing CA’s: Past Work

1. Experiments with human subjects
   – good for understanding how real people bid;
     less good for examining computational characteristics
   – valuation functions hand-crafted
   – untrained human subjects may be overwhelmed by large problems
2. Analysis of particular problems to which CA’s are well-suited
   – generally propose alternate (restricted) mechanisms
   – useful for learning about problem domains

3. Artificial Distributions

- Advantage: easy to generate any number of datasets parameterized by the desired number of bids, goods
- Disadvantages: don't explicitly model bidders; lack a real-world economic motivation
  - all bundles requesting same number of goods are equally likely
  - price offers are unrelated to which goods requested
  - price offers usually not superadditive in number of goods
  - no meaningful way to construct sets of substitutable bids

Combinatorial **A**uction **T**est **S**uite (**CATS**)

- Our goal: create a test suite for the combinatorial auction winner determination problem that will be of use to other researchers
  - a collaborative effort with CA community
- Start with a domain, basic bidder preferences
- Derive an economic motivation for:
  - goods in bundle
  - valuation* of a bundle
    - * we assume incentive compatibility
  - what bundles form sets of substitutable bids

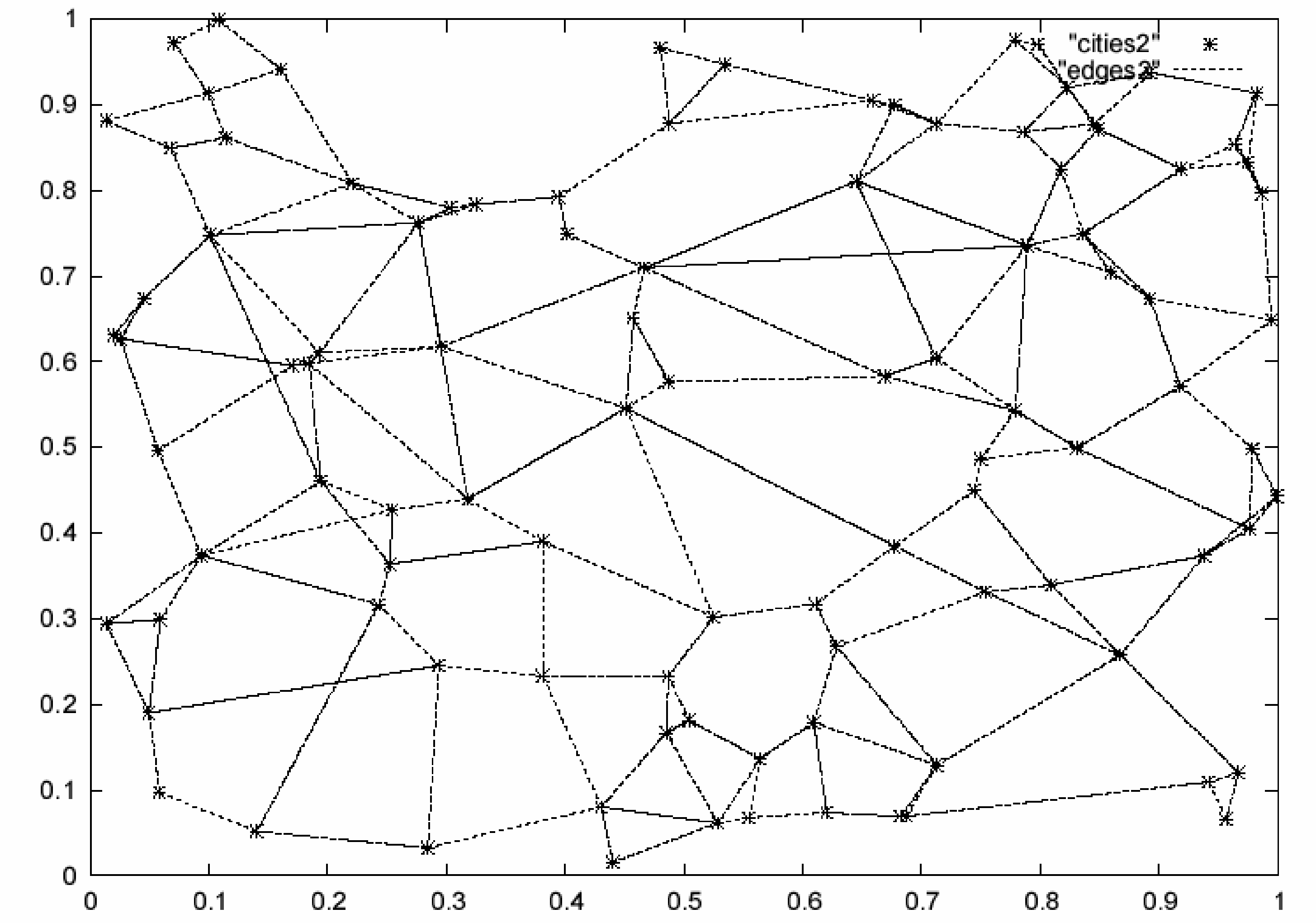
CATS Distributions

- Test distributions motivated by real-world problems, where complementarity arises from:
  1. Paths in space
  2. Proximity in space
  3. Arbitrary relationships
  4. Temporal Separation (matching)
  5. Temporal Adjacency (scheduling)

Paths in Space

- Real-world domains:
  - railroad network
  - truck shipping, network bandwidth allocation, natural gas pipeline
    - e.g., see Brewer & Plott, 1996; Sandholm 1993; Rassenti et. al. 1994
- Problem:
  - goods are edges in a graph
  - bidder: acquire a path from a to b by buying a set of edges
- Procedure:
  - generate a random graph
    - why not use a real railroad (etc.) map? Scaling the number of goods.
  - generate bids for each bidder

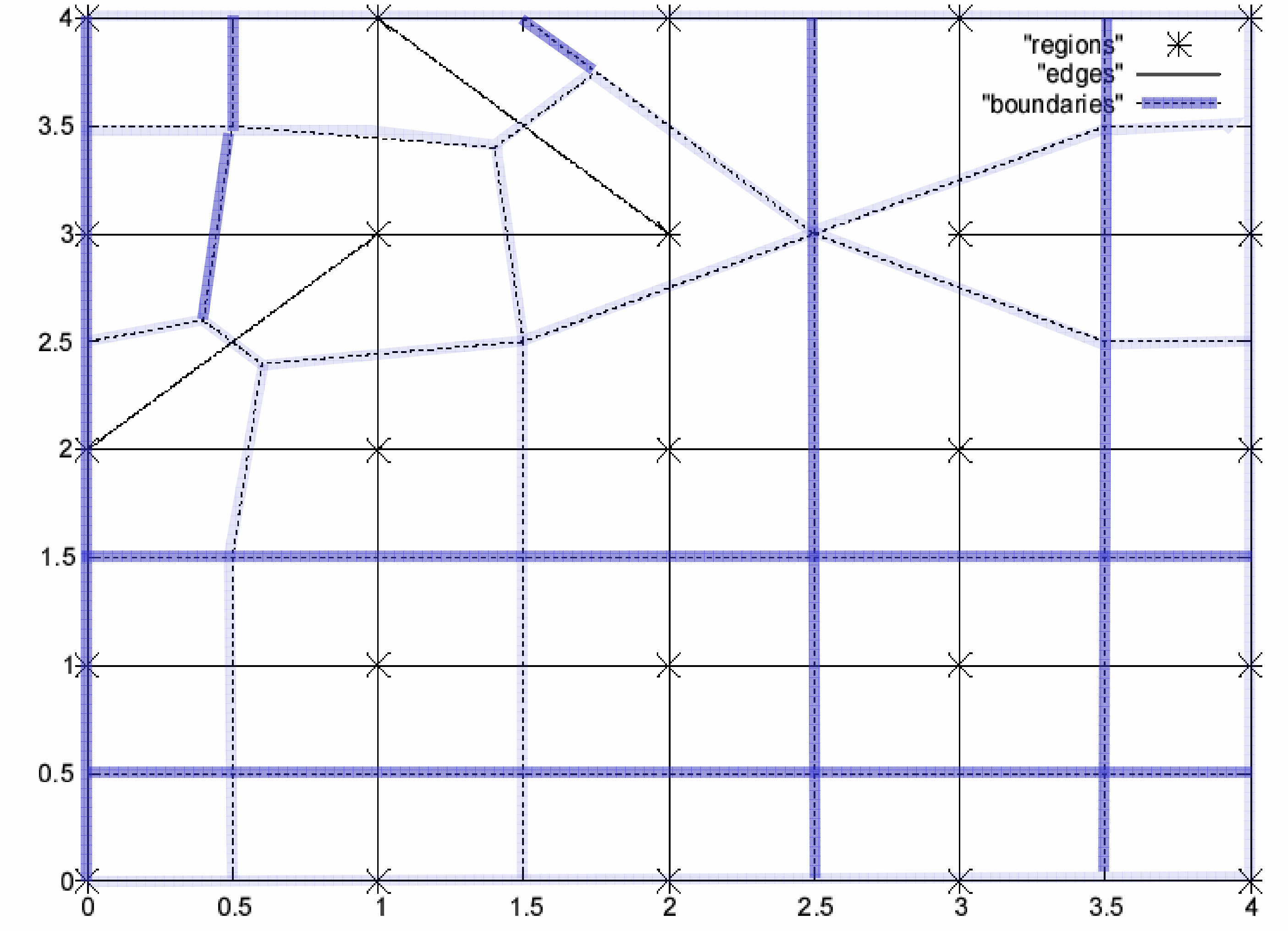
Sample Graph



Proximity in Space

- Real-world domain: real estate
  - e.g., see Quan, 1994.
- Problem:
  - goods are nodes in a graph
  - edges indicate adjacency between goods
  - bidder: buy a set of adjacent nodes
    - according to common and private values

Sample Graph



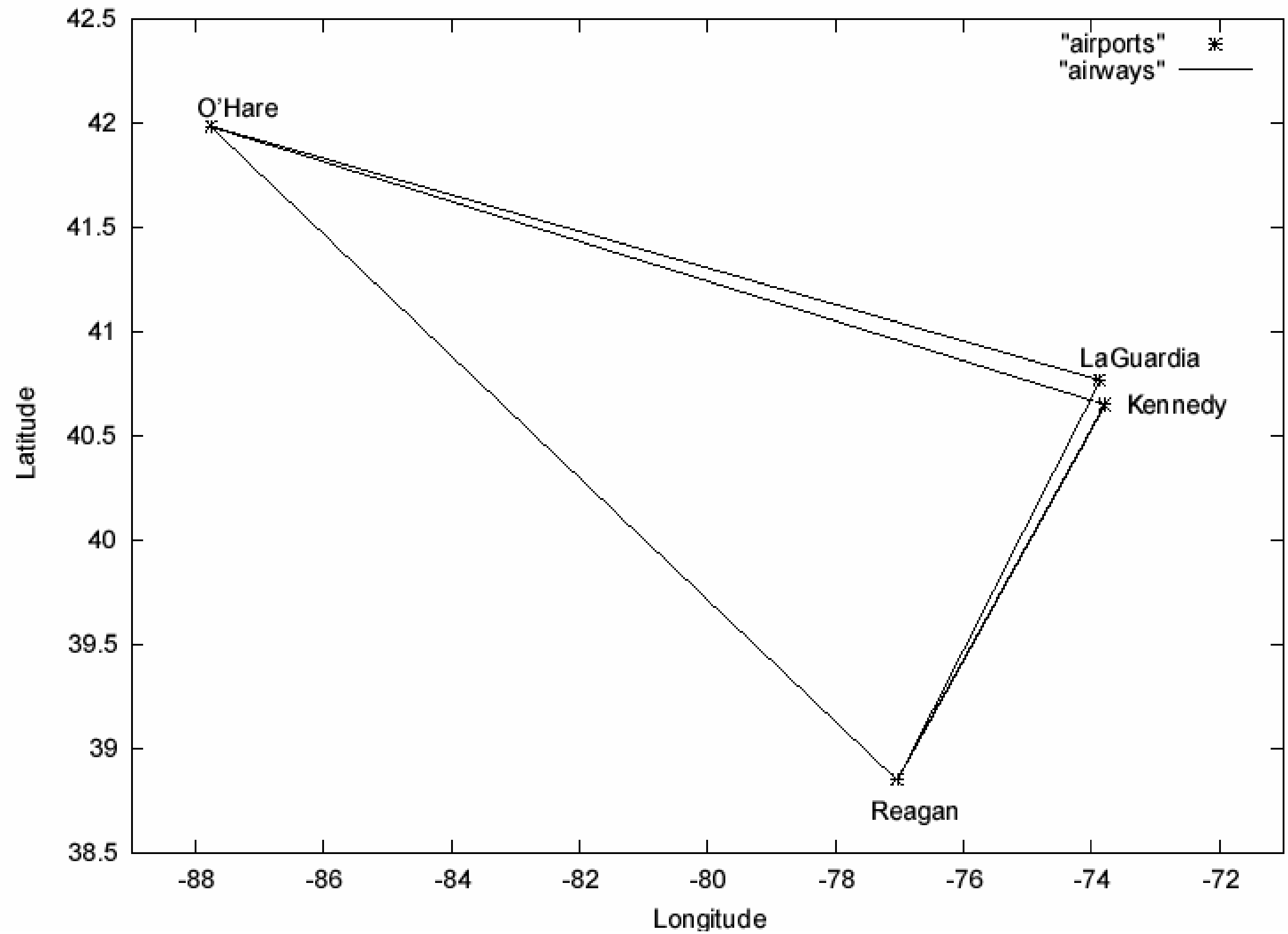
Arbitrary Relationships

- Some goods do not give rise to a notion of *adjacency*, but regularity in complementarity relationships can still exist
  - e.g., physical objects: collectables, semiconductors, …
- Problem:
  - goods are nodes in a fully-connected graph
  - edges weighted with probability that the pair of goods will appear together in a bid
- Procedure:
  - generate a fully connected graph with random weights, CV's
  - generate sets of bids for each bidder
    - bias the likelihood that a good will be added to a bid according to the weights of the edges it shares with goods already in the bid

Temporal Matching

- Real-world domain:
  - corresponding time slices must be secured on multiple resources
  - e.g., aircraft take-off and landing rights
    - e.g., see Rassenti et. al., 1982; Grether et. al. 1989.
- Airport map
  - goods are time slots, not nodes or edges
    - thus, a random graph is not needed for scalability
  - we use the map of airports for which take-off and landing rights are actually sold
    - the four busiest airports in the USA

Airport Map



"airports"
"airways"
O'Hare
LaGuardia
Kennedy
Reagan
Latitude
Longitude
42.5
42
41.5
41
40.5
40
39.5
39
38.5
-88
-86
-84
-82
-80
-78
-76
-74
-72

Temporal Scheduling

- Real-world domain: distributed job-shop scheduling with one resource
  - e.g., see Wellman et. al., 1998.
- Bidders:
  - want to use resource for a given number of time units
  - one or more deadlines having different values to them
- Assumptions:
  - all jobs are eligible to start in the first time-slot
  - each job is allocated continuous time on resource

Legacy Distributions

- CA algorithm researchers have compared performance using each other's distributions
  - e.g., Andersson et. al., Boutilier et. al., de Vries & Vohra, Fujishima et. al., Parkes, Sandholm, others…
  - despite the drawbacks discussed earlier, these distributions will remain important for comparing new work to previously published work
- CATS has a legacy distributions section to facilitate future testing
  - if we left something out, we'll add it!

Conclusion

- CATS is a test suite for combinatorial auction winner determination algorithms
- It represents a step beyond current CA testing techniques because distributions:
  - model real-world problems
  - model bidders explicitly
  - are economically motivated
- Please see http://robotics.stanford.edu/CATS