Towards a Universal Test Suite for Combinatorial Auctions

Kevin Leyton-Brown
Mark Pearson
Yoav Shoham

Department of Computer Science
Stanford University

** Thanks to Shobha Venkataraman*

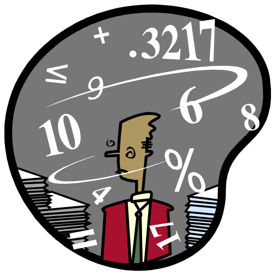
Combinatorial Auctions

- CA's: mechanisms that allow bidders to explicitly indicate complementarities and substitutabilities
  - many goods are auctioned simultaneously
    - goods may be indivisible (*single-unit*) or divisible (*multi-unit*)
  - bids name an arbitrary bundle and a price offer
  - bidders may submit multiple bids
  - bidders can indicate substitutabilities between bids

Winner Determination

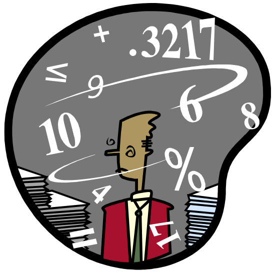
- Given a set of bids, find the revenue-maximizing subset of these bids in which no more than the maximum number of units for each good is allocated
- In the past few years, computer scientists have done a lot of work on the CA winner determination problem
  - special-purpose CA algorithms, mixed-integer formulations
  - approximation techniques, tractable cases, preprocessing, etc...
- However, it is hard to compare these approaches
  - no test sets have been universally agreed-upon
  - problems have been found with some widely-used test sets

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

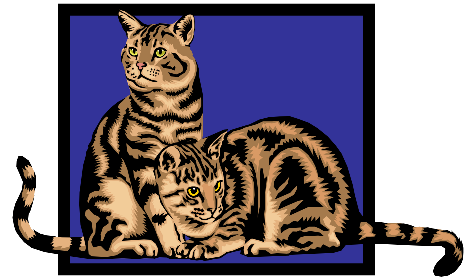


- <u>Advantage</u>: easy to generate any number of datasets parameterized by the desired number of bids, goods
- <u>Disadvantages</u>: don’t explicitly model bidders; lack a real-world economic motivation
  - all bundles requesting same number of goods are equally likely
  - price offers are unrelated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

Combinatorial Auction Test Suite (CATS)

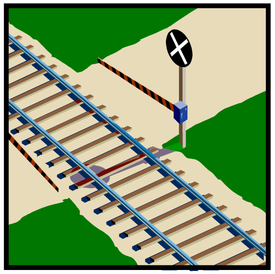
- Our goal: create a test suite for the combinatorial auction winner determination problem that will be of use to other researchers
  - today I'll present our proposal
  - we hope this will be the beginning of a collaborative effort
    - please give us your feedback if you are interested
- Start with 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



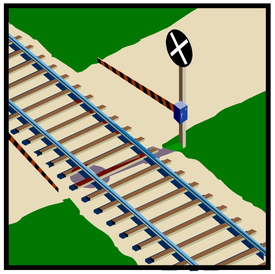
- Test distributions motivated by real-world problems, where complementarity arises from:
  1. Paths in space
  2. Proximity in space
  3. Arbitrary relationships
  4. Temporal Separation (matching)
  5. Temporal Adjacency (scheduling)
- Disclaimer:
  - simplified distributions today, for time reasons
    - though I'll give some details, I'll focus on the big picture
  - full details can be found in our paper and online

Paths in Space



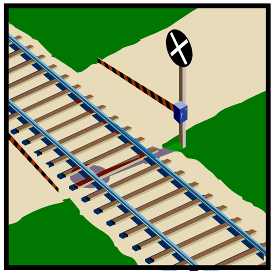
- Real-world domains:
  - railroad network
  - truck shipping, network bandwidth allocation, natural gas pipeline
    - e.g., see Brewer & Plott, 1996; Sandholm 1993; Rassenti et. al. 1994
- Problem:
  - goods are edges in a graph
  - bidder: acquire a path from a to b by buying a set of edges
- Procedure: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

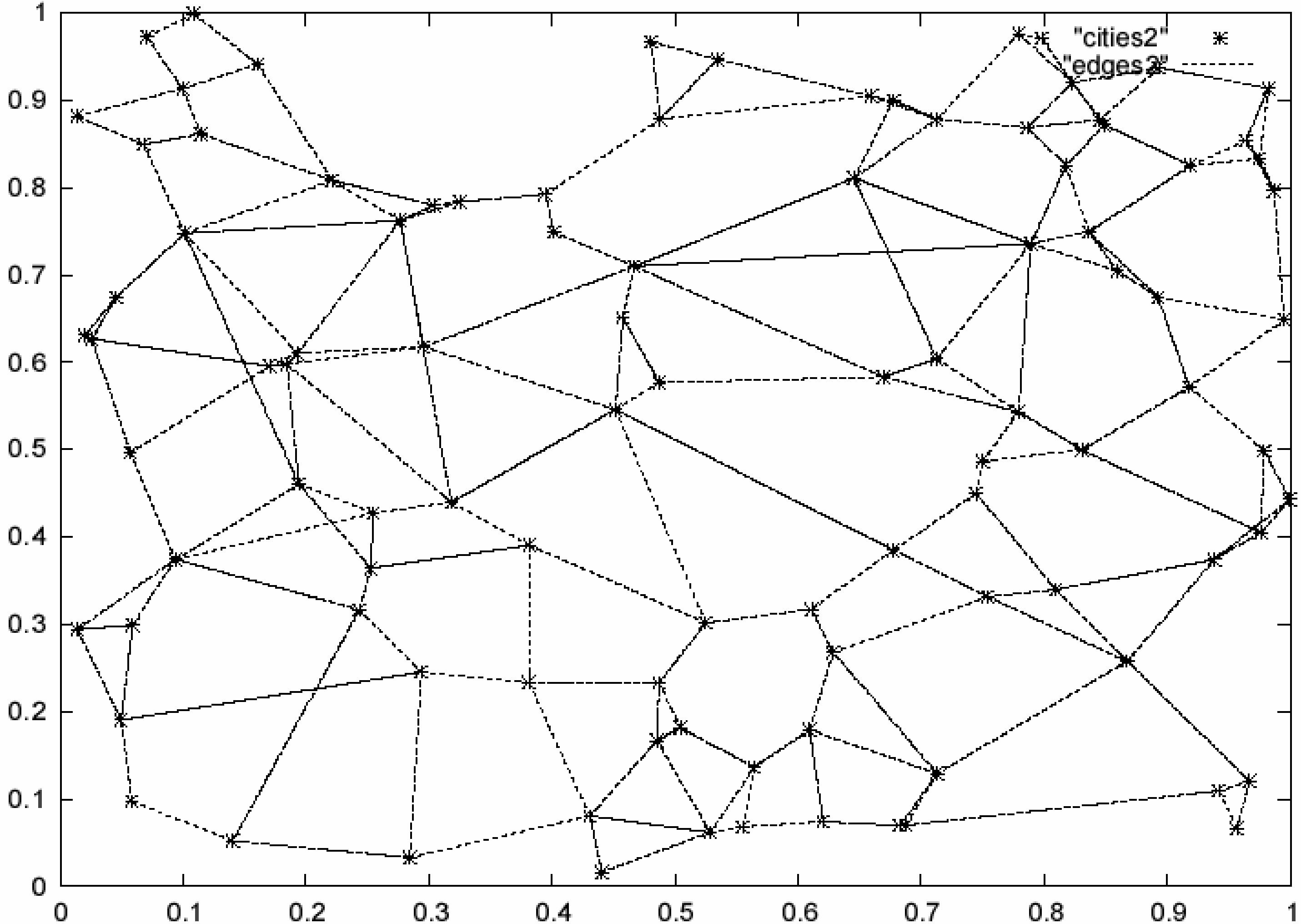
Generate Random Graph



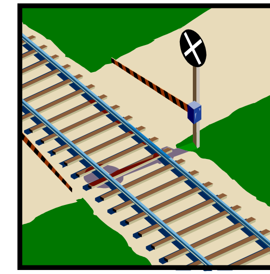
- nodes connected to 2 nearest neighbors
- repeat:
  - compare the best path between two random nodes with paths that can found after creating one or more new edges
    - taking into account a penalty for edge creation
  - if a new path is better, add the new edge(s)

Sample Graph



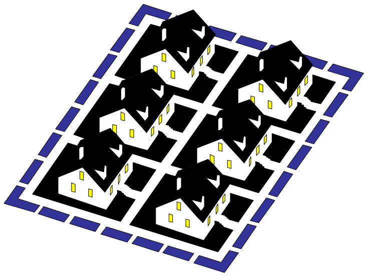


Generate Bids

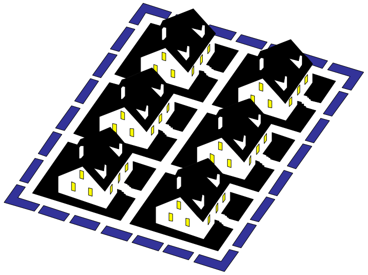


- Basic bidder preferences: desired start and end cities
- Valuation for route:
  - proportional to distance along path
  - superadditive in number of edges
  - random noise
- Substitutable bids
  - bundles for which valuation - cost of shipping > 0
    - cost of shipping is distance along path
  - price offer: valuation – cost
- ...while more bids desired, repeat for a new bidder

Proximity in Space



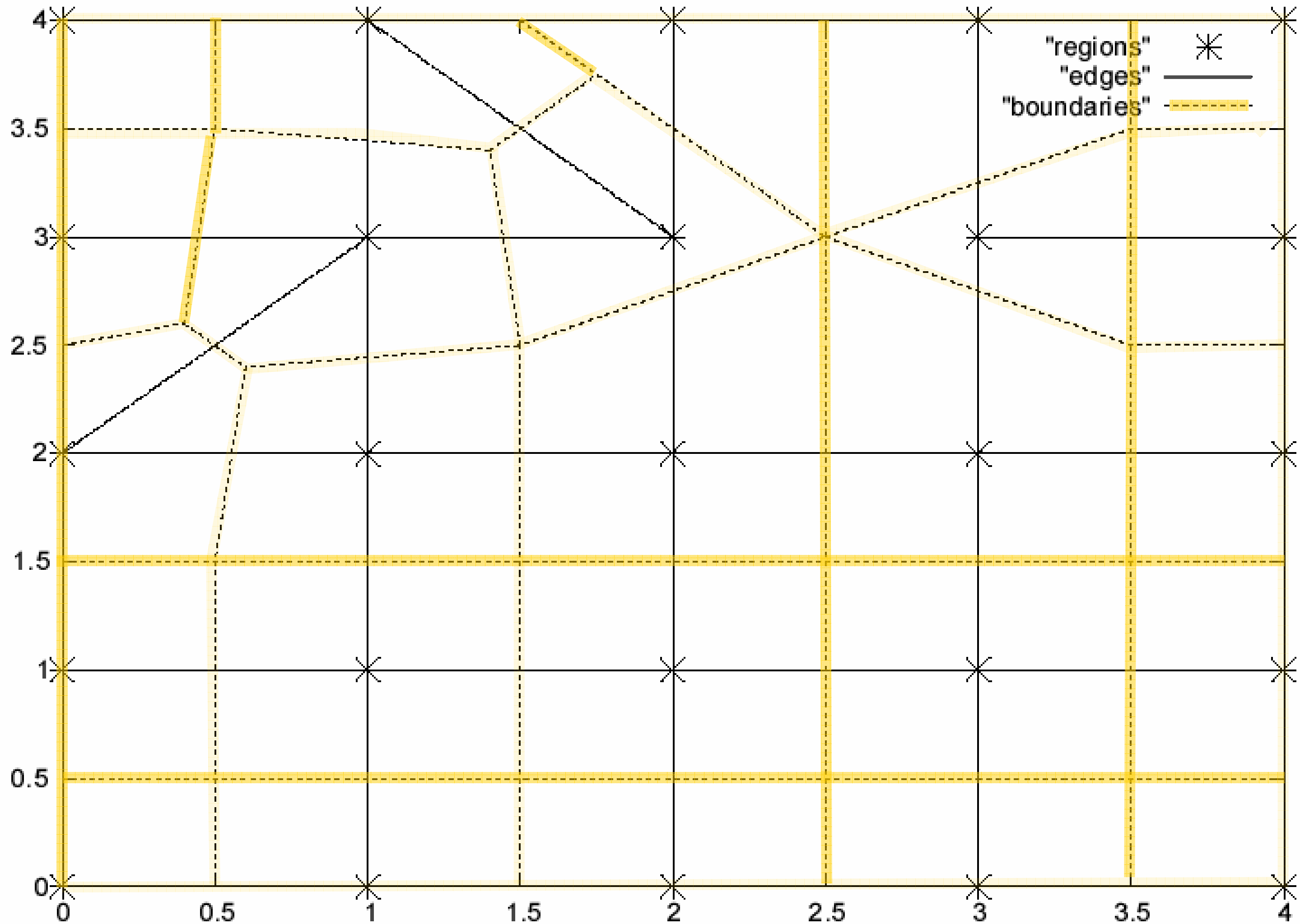
- Real-world domain: real estate
  - e.g., see Quan, 1994.
- Problem:
  - goods are nodes in a graph
  - edges indicate adjacency between goods
  - bidder: buy a set of adjacent nodes
    - according to common and private values



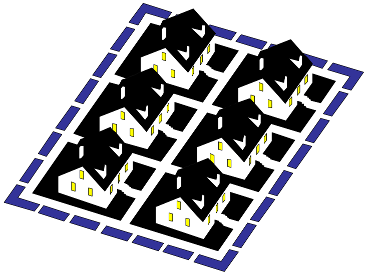
Generate a Graph

- Simple graph:
  - fixed number of neighbors per node
    - place nodes on a grid
    - edges connect horizontally- and vertically-adjacent nodes
- More complex graph:
  - allow a variable number of neighbors per node
  - follow the simple technique above, but:
    - with probability p_1, omit a horizontal or vertical edge
    - with probability p_2, add a diagonal edge
- Associate a common value with each good
  - represents appraised/market/expected resale value

Sample Graph



Generate Bids



1. Basic bidder preferences:
   - private values for each good
2. Pick one good at random, weighted by private values
3. Add another good with probability p. Which good?
   - consider only bids adjacent to already-chosen good(s)
   - weighted by # adjacent goods, bidder's preferences
4. Set price offer: depends on common value and private value; superadditive in number of goods
5. Generate additional bids substitutable with this bid
   - sharing at least one good with this bid
   - based on same private values

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

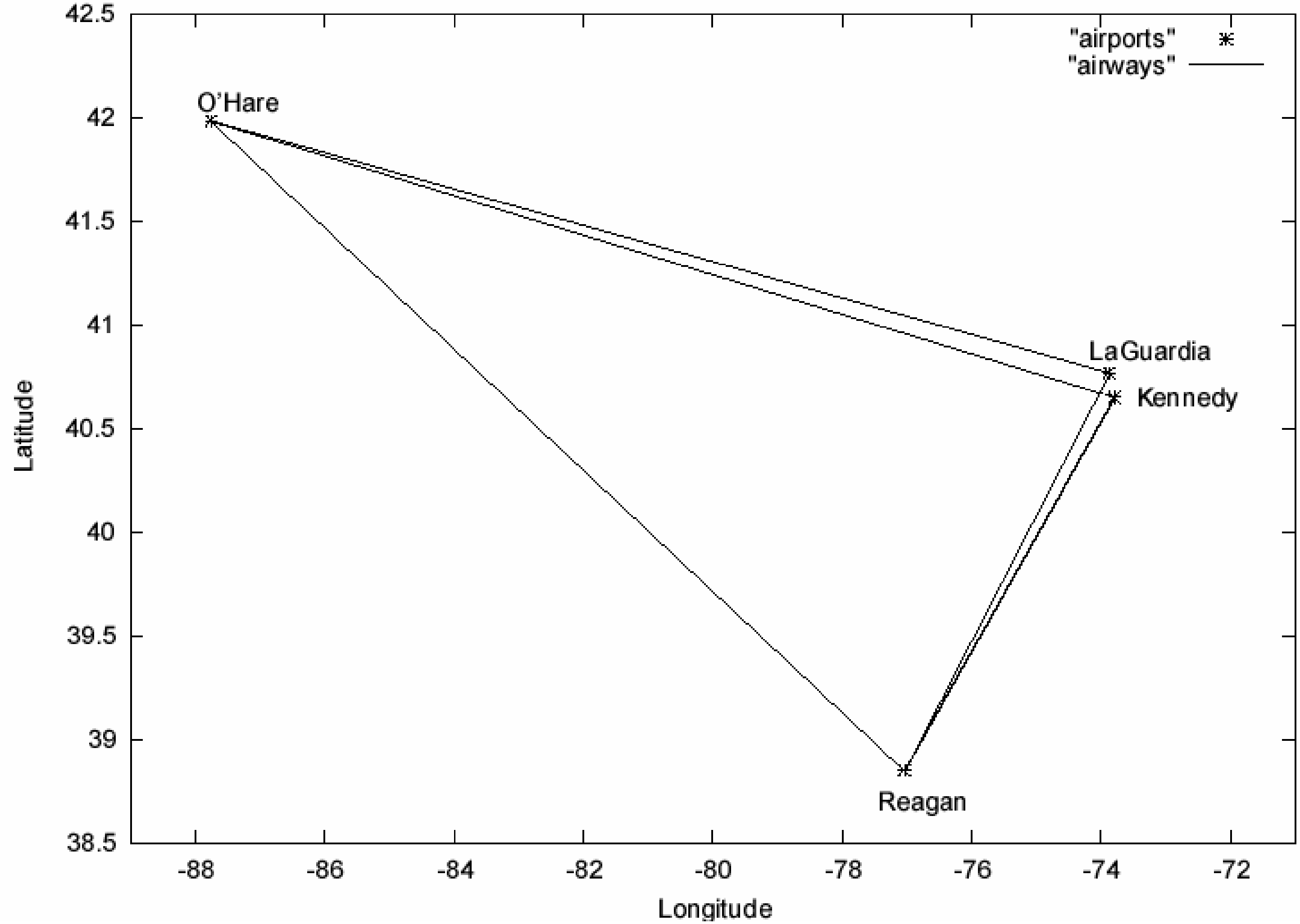
Generate Bids

- Generalization of bid-generation technique from previous section
  - basic bidder preferences are private values
  - choose a first good, biased by private values
  - repeatedly decide whether to keep adding goods
    - add one good
- Choosing *which* good to add
  - likelihood of adding good x to bundle B depends on:
    - sum of edge weights between x and other goods in B
    - private value of x

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

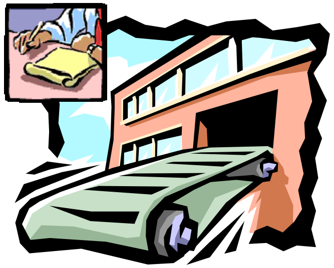


"airports"
"airways"
O'Hare
LaGuardia
Kennedy
Reagan
Latitude
Longitude
42.5
42
41.5
41
40.5
40
39.5
39
38.5
-88
-86
-84
-82
-80
-78
-76
-74
-72

Generate Bids

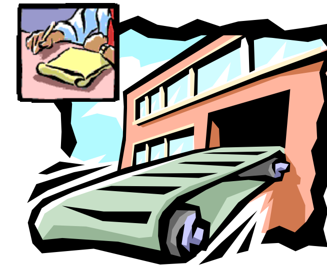
- Basic bidder preferences: preferred departure time and flight duration
  - airline gets utility u_{max} for securing these time slots
- Other slots less desirable—utility falls exponentially as:
  - the arrival time increases (plane gets later)
  - the flight duration increases (flight gets longer)
  - 0 utility for arrival time or flight duration > maximums
- Substitutable bid for every pair of time slots having positive utility
  - price offer = utility

Temporal Scheduling



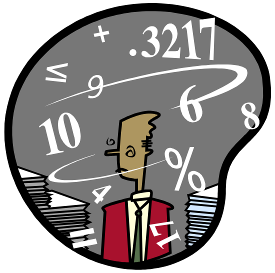
- Real-world domain: distributed job-shop scheduling with one resource
  - e.g., see Wellman et. al., 1998.
- Bidders:
  - want to use resource for a given number of time units
  - one or more deadlines having different values to them
- Assumptions:
  - all jobs are eligible to start in the first time-slot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

Bid Generation



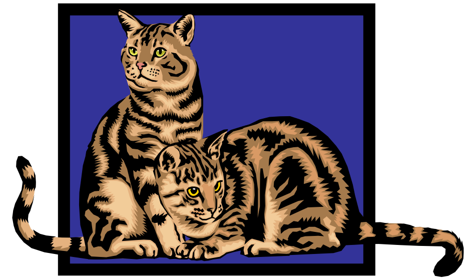
- Basic bidder preferences:
  - a set of deadlines $d_1, \ldots, d_n$
  - value of job finished by d_1 is v_1
- Set of substitutable bids:
  - Bid v_i for a job that finishes on or before d_i, after $d_{i\text{-}1}$
  - $v_i = (d_1 / d_i) \cdot v_1$
    - decrease in value is proportional to increase in lateness

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!

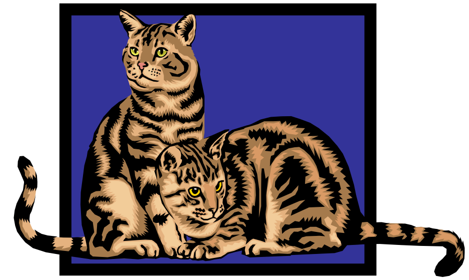
An Aside: Experimental Results



We have done some preliminary testing of our first-generation CA algorithm (CASS) vs. the new CPLEX 7.0 on distributions presented here

- we were competitive with the previous CPLEX version
- CPLEX 7.0 is much faster
  - on some problem sets CASS is as fast as CPLEX
  - on others, we have observed CPLEX to be as much as two orders of magnitude faster
  - test your CA algorithms against this new version of CPLEX!
    - a conversion utility is available on the CATS website
- results will be available on my web page (soon)

Future Work



- Update CATS according to your questions, criticisms and suggestions for improvement
- TAC distribution (arbitrary relationships)
- Add new multi-unit real-world domains:
  - bandwidth allocation, commodity flow (paths in space)
  - spectrum auctions (proximity in space)
  - multi-unit pollution rights (arbitrary relationships)
  - power generation (temporal scheduling)

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
- We hope that, with your contributions and feedback, CATS will evolve into a universal test suite for combinatorial auctions!
  - please see http://robotics.stanford.edu/CATS