A multi-phase iterative combinatorial auction for airport landing slots

Robert Day & S. Raghavan
University of Maryland

Acknowledgment: Thanks to Mike Ball for many fruitful discussions in the course of this research.

FAA Landing Slot Auctions

- Slots are controlled at 3 US airports
- Currently slots are allocated administratively/politically – Discussion on using market based mechanisms (i.e., auctions) to allocate slots.
- An auction should be *designed to the constraints of the system*:

  800 or so slots/day at LGA, for example

  Goods for sale are mostly substitutes

  Complements arise in the form of banking (flights arrive together) and shuttling (e.g., hourly flights)

FAA Landing Slot Auctions

- Focus on *single airport*, as there are only 3 slot controlled airports. (i.e., ignore network effects as there is reason to believe they would be negligible in this case)
- Focus on *landing slots* (for example these may be the right to land within a ½ hr or ¼ hr interval). Easy to incorporate take-off slots if necessary. With gate capacity constraints, orderly arrival should translate into orderly departure.
- Focus on long-term market. Auction is to be held to auction rights for landing slots for next 5 or 10 yrs.

Combinatorial Auctions

- Allow bidders to more accurately express their preferences by allowing them to place bids on bundles.
- Else bidder faces *exposure* problem.
  - Substitutes exposure – stuck with 2 goods when we want 1.
  - Complements exposure – paid too much for a single good.

Bidding honestly exposes a bidder to risk

- Computational Problem: With 800 landing slots the problem will become computationally explosive. (No hope for directly requesting bids on all possible bundles bidders are interested in, and solving the winner determination problem).

Other Issues

- Desire *price discovery*. Bidders may not be sure how much a landing slot is worth and would benefit in an auction where the price of the items increase gradually.
- *Threshold Problem*
- *Free-Rider Problem*

$b_1(\{A,B\}) = 10$

$b_2(\{A\}) = 7$

$b_3(\{B\}) = 6$

Landing Slot Auction

- Stage I : Bid Tables (multi-round)
  - Price Discovery, Linear Prices, Deals with Substitute Preferences, Addresses Substitutes Exposure Problem, Rapid (instantaneous) solution.
- Stage II : Package Bidding (multi-round)
  - Demand revelation for bundles. Deals with complement preferences, Addresses Complements Exposure Problem, Players "probe" bundles to determine a winning price, Rapid (instantaneous) Solution
- Stage III : Sealed Bid Auction (single round)
  - Addresses threshold and free-rider problem. Finds most efficient solution to the slot allocation problem. Computationally more challenging, but takes place only after price discovery/demand revelation phases.

Consistent with the latest hybrid methods

Improves upon the PAUSE auction of Kelly & Steinberg [2000]

Well-Suited to the Landing Slot Application

Example: A Bid Table

	LAX	MIA	ATL1	ATL2	ATL3
9:00	10				
10:00	15				
11:00	10				
12:00		15			
1:00		17			
2:00			18	15	10
3:00			18	15	10
4:00			18	15	10
5:00		20			

Stage 1: Ascending Bid Table Auction

- Bidder submits tabular bids for slots. (These remain private until the end of Stage 1.)
- Auctioneer calculates winners solving assignment problem, and the minimum Walrasian equilibrium price (these are linear supporting prices for the slots).
- Auctioneer reports current winners for the slots and the current prices for the slots.
- Bidders need to resubmit tabular bids with following eligibility rules:
  - Entries in bid tables can only increase
  - Entries in winning rows (i.e., if you have provisionally won the slot) need not increase
  - Entries in non-winning rows must be increased to the current price + 1 increment, or you get one last chance to increase it and then it stays fixed.
- Process repeats until no new bid tables are submitted. Stage 1 is over. Final bid tables are published.

Stage I : Benefits of Bid Tables

- Alleviates Exposure Amongst Substitutes
- New entrants may simultaneously compete for several potential goods
- Tactical manipulation is more difficult here than in SMR
- Individual good prices are more meaningful than in the SMR context

Stage II : Package Bidding

- Alleviates Exposure Amongst Complements
- Deliberately delays addressing threshold and free-rider problem. (providing information to address threshold problem exacerbates free-rider problem)
- Bidders *probe* to find a bundle worth paying for
- Probing
  - interested in slots a, f, and g. How much do I need to pay to win these 3 slots as a bundle.
  - Can answer this by removing a, f, and g, and finding how much revenue goes down.
  - Bidder must pay amount equal to revenue decrease + 1 unit to win goods a, f, and g.
  - Can extend this calculation in the presence of previously accepted probed bids. (assignment problem needs to be solved)
- Bidder can *rapidly* (instantaneously) probe bundles to find a bundle they wish to bid on

Stage II : Package Bidding

- Probing takes place continuously
- In each round bidders go in random order submitting bids (since they are based on probing, they are immediately provisionally winning).
- Continues until no further probed bids submitted.
- Could limit size of bundles as we progress through these rounds (limit probed bundle size to 5, 10, 15, 20, 25, etc).
- <u>BENEFITS</u>:
  - Demand revelation/price discovery phase for bundles
  - Bidders share the computational burden of searching for improving bundles

Stage III : Sealed Bid Auction

- Alleviates Threshold Problem
- Makes Free-Riding Risky
- Several bundles have been eliminated from consideration in the earlier rounds

$b_1(ABCD) = 70$

A **10**	**B** **10**
C **10**	**D** **10**

$v_2(AB) = 40$

$v_3(CD) = 40$

Stage III : Sealed Bid Auction

- Bidders submit bids on all packages they are interested in.
- Auctioneer finds *efficient allocation* (solves a large Integer Programming problem)
  - Alleviates Threshold Problem
  - Makes Free-Riding Risky
  - Large number of bundles have been eliminated from consideration because of the earlier rounds
- Auctioneer finds “appropriate set of payments for winners”; specifically set of bidder pareto-optimal core payments

Stage II : Package Bidding

Why not wait until Stage III?

- Probed bundle price can be used as a bound in the sealed-bid round
- Eliminates jump bidding problems
- Each Stage III bid is screened so that it could not be winning in Stage II
- Bundles won in Stage II have no constraints in Stage III (for that bidder)

Concluding Remarks on 3-Phase Slot Auction

- Developed a 3-phase auction to deal with large number of slots.
- Stage I deals with substitutes.
  - Promotes price discovery
  - Individual good prices are used as long as they are guaranteed to be meaningful
- Stage II deals with complements.
  - Promotes demand revelation for bundles
  - Forces bidders to participate by capping their Stage III bid based on Stage II participation
- During the human interaction stages (i.e., Stage I and II) all computations can be performed quickly
- Difficult computations are postponed until human interaction is complete

BACKUP

Theoretical Result

- Assignment Preferences are contained in the set of "gross substitute" preferences.
- Implication: In a bid table auction with Vickrey payments shill bidding is not profitable. Further, Vickrey payments are in the core.

Stage III Winner Determination

Maximize $\sum_{i,j,k} b_{ijk} x_{ijk} + \sum_{l} B_l \, y_l$

subject to:

Supply Constraints

$$\sum_{j,k} x_{ijk} + \sum_{l \,|\, i \in S(l)} y_l \le 1 \text{ , for each good } i$$

$$\sum_{i} x_{ijk} + \sum_{l \,|\, j,k \in F(l)} y_l \le 1 \text{ , for each bid table column } j,k$$

Demand Constraints

Where $x_{ijk} = 1$ if bid table entry i,j,k is accepted

$y_l = 1$ if package bid l is accepted

$= 0$ otherwise

Stage III : Finding a blocking coalition

Maximize $\sum_{i,j,k} b_{ijk} x_{ijk} + \sum_{l} B_l y_l - \sum (V_j - \pi_j) \gamma_j$

subject to:

Supply Constraints

$\sum_{j,k} x_{ijk} + \sum_{l \mid i \in S(l)} y_l \leq 1$, for each good i

$\sum_{i} x_{ijk} + \sum_{l \mid j,k \in F(l)} y_l \leq \gamma_j$, for each bid table column j,k

Demand Constraints

Where $x_{ijk} = 1$ if bid table entry i,j,k is accepted

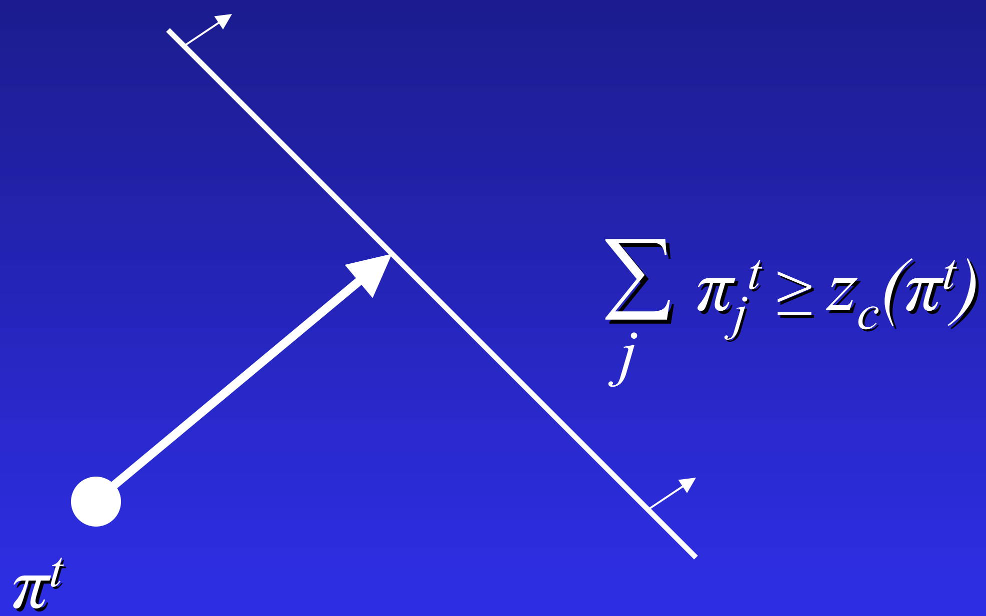
$y_l = 1$ if package bid l is accepted

$= 0$ otherwise

Bidder-Pareto-Optimal Core Prices

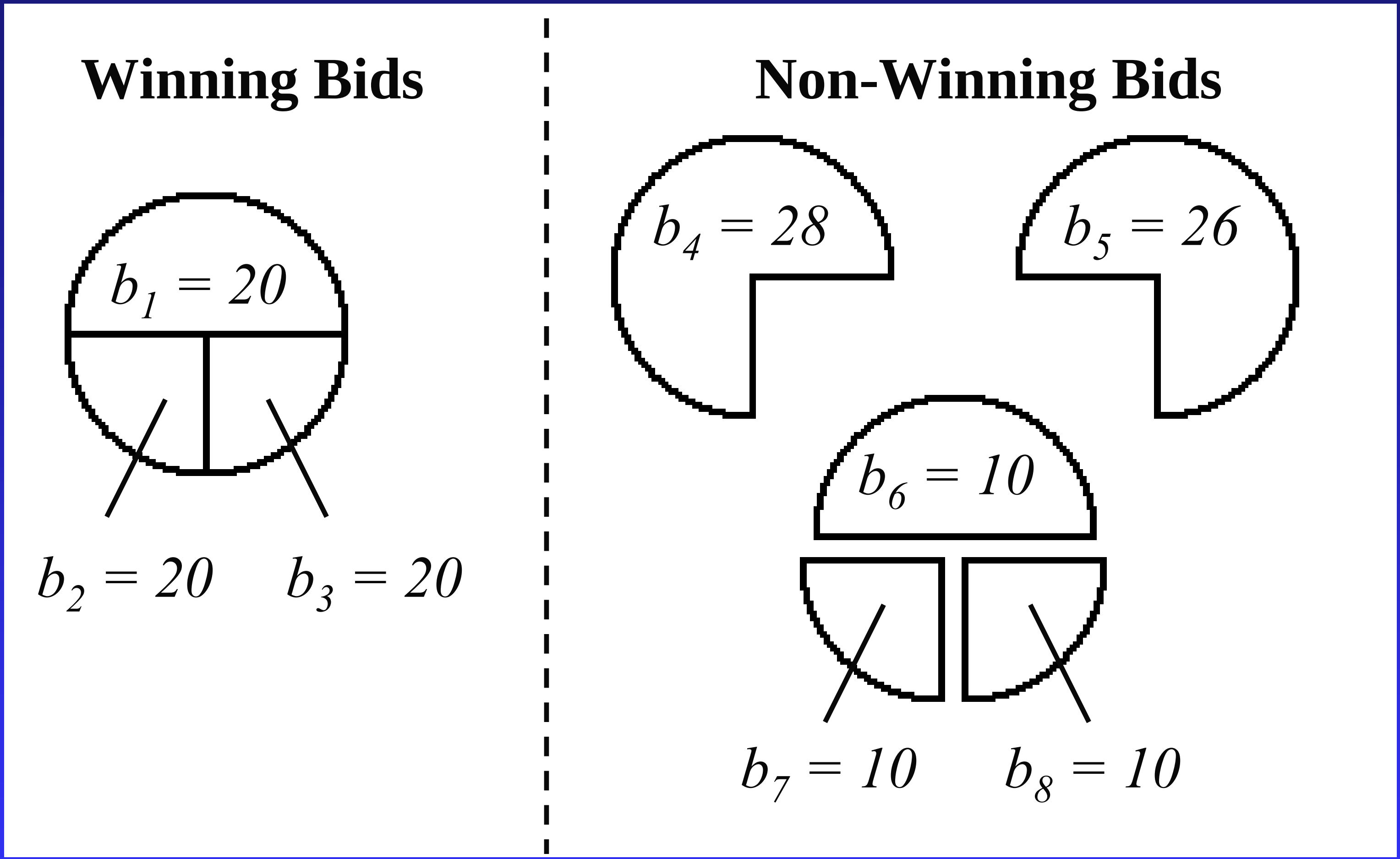
- VCG prices may not be in the core when complements are present
- Ausubel-Milgrom Proxy Auction. Can take a long time to converge.
- Hoffman et al accelerate: Start at VCG, then run Ausubel-Milgrom
- Our Approach: Start at VCG, then run Core Constraint Generation
  - Payment problem. Minimize sum of payments by winners subject to core constraints (1 for each coalition)
  - Separation problem. Given a set of payments by winners find a coalition that is willing to offer the auctioneer a higher sum total payment to change the allocation.

Generation of Core Constraints



Coalitions appear at most once

Converges to Bidder-Pareto-Optimal Core Prices



Threshold outcome

$p_1 = 14, p_2 = 14, p_3 = 12$

Min Total Pay outcome

$p_1 = 16, p_2 = 12, p_3 = 10$