

Using Mathematical Programming to Make VCG Auctions Tractable

Amy Cohn
October 24, 2006

with Damian Beil, Amitabh Sinha,
Richard Chen, and Shervin AhmadBeygi



Michigan**Engineering**
Industrial and Operations Engineering



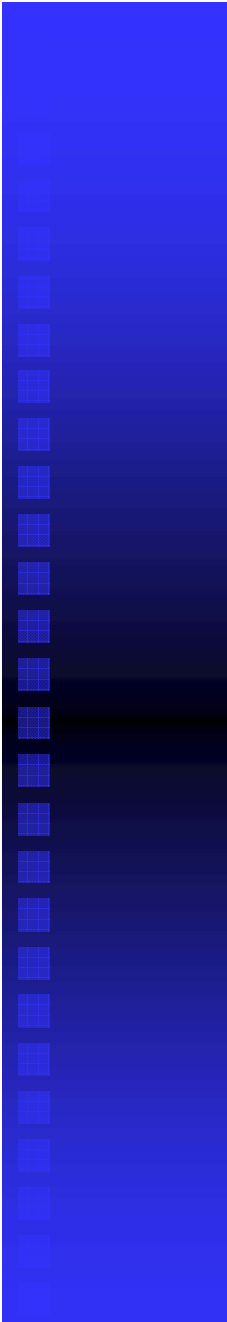
MICHIGAN
ROSS SCHOOL OF BUSINESS

Overview

- *VCG combinatorial auctions* are single-round, second-price auctions in which bidders bid on bundles of items simultaneously
- In theory, this allows bidders to capture the economies of scale and scope between items
- In practice, the exponential number of bids makes both bidding and winner determination intractable
- We propose a method that leverages mathematical programming to *implicitly* capture all bundles without explicit enumeration

Outline

- A Brief History of Auctions
- An Introduction to Mathematical Programming
- Truckload Procurement: A Motivating Example
- Using MP to Make VCG Auctions Tractable
- Examples and Computational Results
- Conclusions and Areas for Future Research



A Brief History of Auctions

A Brief History of Auctions

- Date as far back as 500 BC in Babylon
- Records from Roman auctions in 193 AD
- Sotheby's and Christie's art auctions from 1700's
- eBAY – baseball cards to diamonds to Pez dispensers to rare stamps to Michigan/Iowa football tickets

A Brief History of Auctions

- Many types of auctions
 - ◆ Selling vs. buying
 - ◆ Bidding structure
 - ◆ Open vs. closed bid
 - ◆ Single vs. multi-round
 - ◆ Determining winner
 - ◆ Determining payment

A Brief History of Auctions

- Vickrey (second-price) procurement auction
 - ◆ Single-round, closed bid
 - ◆ Each bidder places a bid
 - ◆ Lowest bidder wins
 - ◆ Payment is *second lowest* bid

A Brief History of Auctions

- Optimal Vickrey bidding strategy: Bid your true cost
 - ◆ If you bid higher...
 - ◆ ...you decrease your chances of winning
 - ◆ ...you don't improve your payment if you do win
 - ◆ If you bid lower...
 - ◆ ...you might win, but lose money!

A Brief History of Auctions

- Benefits of a Vickrey auction:
 - ◆ Economically efficient allocations
 - ◆ Well-defined optimal bidding strategy

A Brief History of Auctions

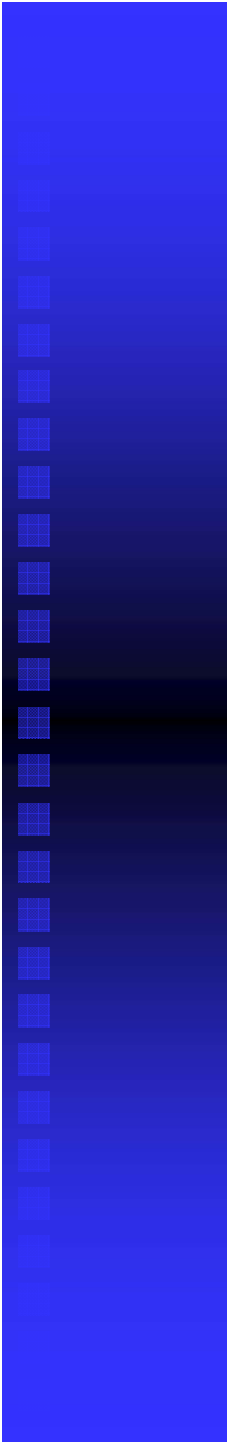
- VCG combinatorial auction
 - ◆ Many items being auctioned simultaneously
 - ◆ Costs are not necessarily additive
 - ◆ Economies of scale
 - ◆ Economies of scope

A Brief History of Auctions

- VCG combinatorial mechanism
 - ◆ Bidders bid on *bundles* of items, capturing costs accurately
 - ◆ Winner determination problem
 - ◆ Payment determination problem
- Challenges
 - ◆ Combinatorial bidding
 - ◆ Combinatorial winner, payment determination problems

A Brief History of Auctions

- Current approaches to VCG combinatorial auctions in the literature:
 - ◆ Limit the combinations on which bids can be place (Pekeč and Rothkopf 1998, 2003, 2006)
 - ◆ Iterative (multi-round) auctions to replicate VCG results (Kwon, Anandalingam, Ungar 2005; Ausubel, Cramton, Milgrom 2006; Sandholm and Boutilier 2006)
 - ◆ Approximations and special cases (Sandholm 2002; Tennenholtz 2002)
 - ◆ Algorithms to solve winner determination across all bids (Sandholm, Suri, Gilpin, and Levine 2005; Sandholm 2002)



An Introduction to Mathematical Programming

An Introduction to Mathematical Programming

- Optimization problem
 - ◆ Quantifiable decisions (how much? how many? yes/no?)
 - ◆ Rules that control the interaction between these decisions
 - ◆ Goal/metric for evaluating decisions
- We'll assume deterministic and static

An Introduction to Mathematical Programming

- Mathematical program
 - ◆ Decisions map to variables
 - ◆ Rules map to constraints
 - ◆ Goal maps to objective function
- Linearity is key
- Continuous is nice

An Introduction to Mathematical Programming

An example:

Minimize the cost of my diet, subject to satisfying lower bounds on protein, and upper bounds on calories and fat

An Introduction to Mathematical Programming

Goal: Minimize cost

Rules:

Lower bound on protein

Upper bound on calories

Upper bound on fat

Decisions:

How much:

Chocolate Special K Liver Orange juice Pizza

An Introduction to Mathematical Programming

Goal: Minimize cost

Rules:

Minimum protein requirement

Maximum calories limit

Maximum fat limit

Variables:

$$\mathbf{x}_c \geq 0 \quad \mathbf{x}_k \geq 0 \quad \mathbf{x}_l \geq 0 \quad \mathbf{x}_o \geq 0 \quad \mathbf{x}_p \geq 0$$

An Introduction to Mathematical Programming

Objective:

Minimize

$$c_c x_c + c_k x_k + c_l x_l + c_o x_o + c_p x_p$$

Rules:

Minimum protein requirement

Maximum calories limit

Maximum fat limit

Variables:

$$x_c \geq 0 \quad x_k \geq 0 \quad x_l \geq 0 \quad x_o \geq 0 \quad x_p \geq 0$$

An Introduction to Mathematical Programming

Objective:

$$\text{Minimize } c_c x_c + c_k x_k + c_l x_l + c_o x_o + c_p x_p$$

Subject to:

$$p_c x_c + p_k x_k + p_l x_l + p_o x_o + p_p x_p \geq P$$

Maximum calories limit

Maximum fat limit

Variables:

$$x_c \geq 0 \quad x_k \geq 0 \quad x_l \geq 0 \quad x_o \geq 0 \quad x_p \geq 0$$

An Introduction to Mathematical Programming

Objective:

$$\text{Minimize } c_c x_c + c_k x_k + c_l x_l + c_o x_o + c_p x_p$$

Subject to:

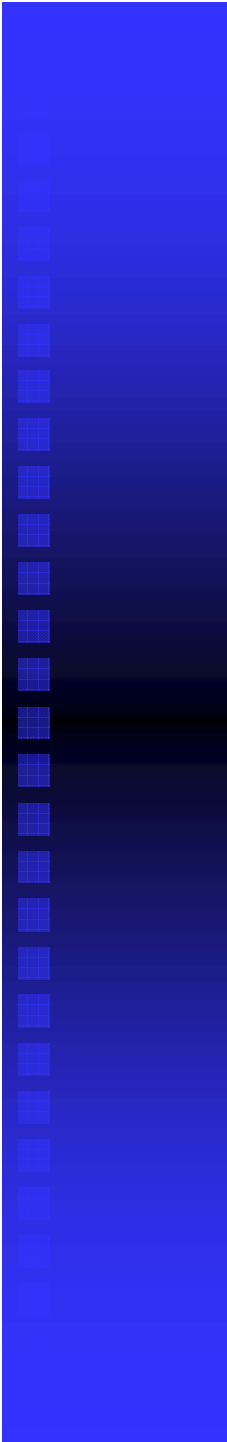
$$p_c x_c + p_k x_k + p_l x_l + p_o x_o + p_p x_p \geq P$$

$$w_c x_c + w_k x_k + w_l x_l + w_o x_o + w_p x_p \leq W$$

$$f_c x_c + f_k x_k + f_l x_l + f_o x_o + f_p x_p \leq F$$

Variables:

$$x_c \geq 0 \quad x_k \geq 0 \quad x_l \geq 0 \quad x_o \geq 0 \quad x_p \geq 0$$



Truckload Procurement: A Motivating Example

Truckload Procurement: A Motivating Example

- Shippers auction off *loads* – origin/destination pairs – to trucking companies to transport for them
- A shipper might have many different loads to offer at the same time
- The costs trucking companies incur depends on the distance, the company's cost-per-mile...
- ... *but also on the other loads the trucking company is moving*

Truckload Procurement: A Motivating Example

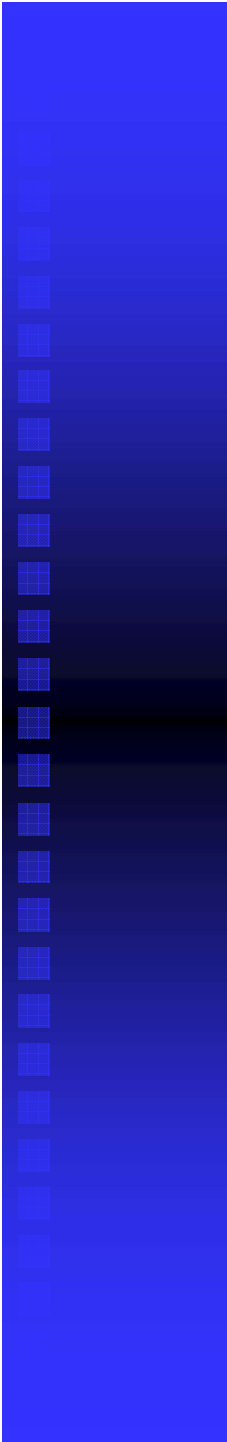
- In bidding for a load, your cost depends on what other loads you win as well
 - ◆ Your cost to carry a load from A to B is:
$$\text{cost/mile} * (d(A/B) + d(B/A))$$
 - ◆ Your cost to carry a load from B to A is:
$$\text{cost/mile} * (d(B/A) + d(A/B))$$
 - ◆ Your cost to carry both loads is:
$$\text{cost/mile} * (d(A/B) + d(B/A))$$

Truckload Procurement: A Motivating Example

- If you only bid on individual loads, you lose this economy of scope
 - ◆ Bid your true cost for just that load, and you may over-bid and lose to a competitor
 - ◆ Bid your estimated cost, assuming other loads to create a *continuous move*, and you may win the load but lose money (because you don't win the other loads)

Truckload Procurement: A Motivating Example

- To capture the true cost, you should bid on continuous moves – bundles of loads pulled in sequence to minimize empty mileage
 - ◆ Exponential number of bids to compute
 - ◆ Exponential number of bids to communicate
 - ◆ Exponential number of bids to consider in winner determination
 - ◆ Exponential number of bids to consider in payment determination



Using MP to Make VCG Auctions Tractable

Using MP to Make VCG Auctions Tractable

- Given a set of bidders and a set of items, enumerate all possible subsets
- Each bidder provides their true cost for each subset
- Optimization problem: Find the cheapest collection of bids such that every item is in exactly one bid (winner determination)
- Optimization problem: For every bidder who is awarded at least one item, strip out this bidder's bids and re-solve the optimization problem (payment determination)

Using MP to Make VCG Auctions Tractable

- Exhaustive bidding approach:

- ◆ N – set of bidders
- ◆ M – set of services

$$\text{Min} \quad \sum_{n \in N} \sum_{s \subseteq M} c_s^n x_s^n$$

$$\text{St} \quad \sum_{n \in N} \sum_{s \subseteq M: i \in s} x_s^n = 1 \quad \forall i \in M$$

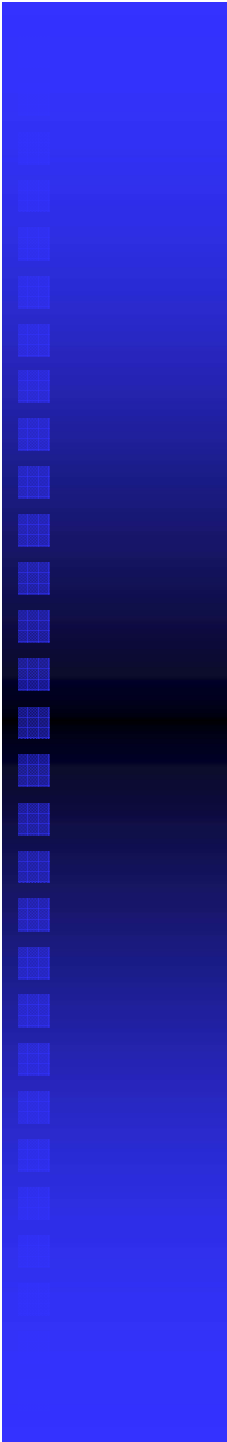
$$x_s^n \in \{0,1\} \quad \forall s \subseteq M, n \in N$$

Using MP to Make VCG Auctions Tractable

- Bidders cannot explicitly compute all 2^n bids, nor can the auctioneer process them, because of the exponential size
- But bidders often *do* have a mechanism for computing each bid – can we embed this mechanism in a mathematical program?
 - ◆ Mathematical programming is about finding the best solution without considering each solution explicitly...

Using MP to Make VCG Auctions Tractable

- If this true cost type can be represented compactly...
- ... and given that the bidder's optimal strategy is to bid truthfully...
- ...can we use the true cost type, rather than the explicit bids, to solve the winner and payment determination problems?



Examples and Computational Results

Examples and Computational Results

- Truckload procurement
 - ◆ Each bidder has a set of existing loads
 - ◆ Bidding on new loads
 - ◆ Cost of a continuous move is mileage-based
 - ◆ Bid loads have one cost per mile
 - ◆ Empty miles have another cost per mile
 - ◆ Existing loads are “free”

Examples and Computational Results

■ Notation

- ◆ K – set of carriers
- ◆ L^0 – set of bid loads
- ◆ L^k – set of existing loads for carrier k
- ◆ m_l^k – cost to carrier k of moving bid load l
- ◆ e_{ij}^k – cost to carrier k of moving empty from load i to load j

Examples and Computational Results

■ Variables

- ◆ $x_l^k = 1$ if carrier k pulls load l , else 0
- ◆ $y_{ij}^k = 1$ if carrier k pulls loads i and j , moving empty in between, else 0

Examples and Computational Results

$$\begin{aligned} \text{Min} \quad & \sum_{k \in K} \sum_{l \in L} m_l^k x_l^k \\ & + \\ & \sum_{k \in K} \sum_{i \in L} \sum_{j \in L} e_{ij}^k y_{ij}^k \end{aligned}$$

(minimize the cost of bid loads plus the cost of empty mileage between loads)

Examples and Computational Results

St

$$\sum_{k \in K} x_l^k = 1 \quad \forall l \in L^0$$

(Each bid load must be awarded to exactly one carrier)

$$x_l^k = \sum_{j \in L} y_{lj}^k \quad \forall l \in L, k \in K$$

$$x_l^k = \sum_{j \in L} y_{jl}^k \quad \forall l \in L, k \in K$$

(If a carrier carries a load, that load must be preceded and followed by two other loads)

Examples and Computational Results

# Carriers	# O/D's	# BL	# EL	# Runs	Avg. # binary var's	Avg. # cont. var's	Avg. # constraints	Avg. # B-B nodes	Avg. run-time (sec.)
10	20	20	100	10	300	9120	630	0.00	0.30
20	40	60	400	10	1600	128371	3280	1.10	16
30	60	100	600	10	3600	432595	7330	10.00	471
30	60	100	600	10	3600	458986	7330	0.00	39
30	60	100	600	5	3600	432559	7330	0.00	81
30	60	100	600	5	3600	432536	7392	15.25	672
30	60	200	1000	5	7000	1634114	14230	17.00	6542

3 dominant carriers with 100 ELs each.

Hub and spoke. 3 Regional hubs. All BLs originate for these hubs.

Max number of winners = 20. Min number of winners = 5.

Biggest runs thus far. 1000 ELs and 200 BLs.

Examples and Computational Results

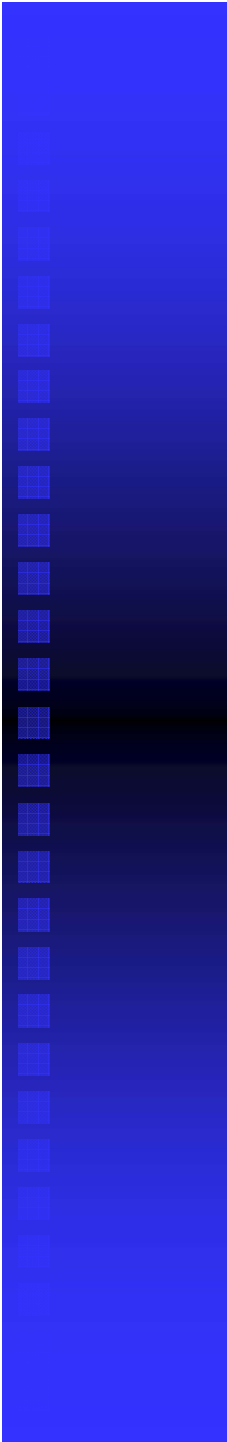
- Wireless spectrum auctions
 - ◆ E.g. frequencies for wireless cell phone communications
 - ◆ Synergies in adjacent bandwidths
 - ◆ Synergies in location

Examples and Computational Results

- Auctions of commodities with non-linear value functions
 - ◆ Economies of scale
 - ◆ Step functions
 - ◆ Piece-wise functional approximations

Examples and Computational Results

- Procurement auctions with capacity constrained suppliers
 - ◆ Suppliers' capacity constraints limit ability to satisfy multiple requests
 - ◆ These constraints can often be captured in standard form:
 - $Ax \leq b$



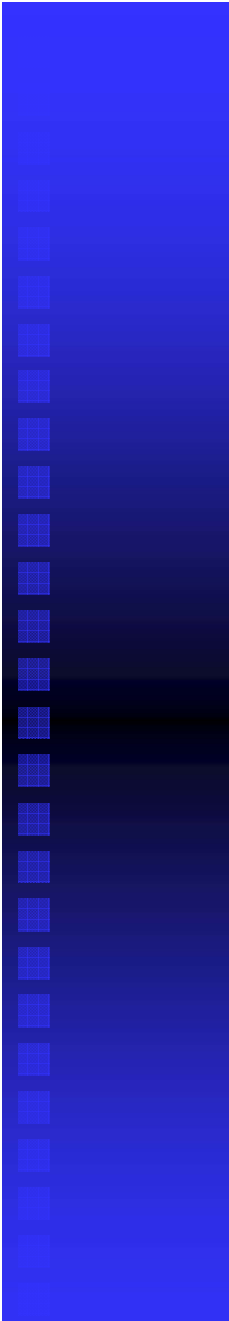
Conclusions and Future Research

Conclusions

- VCG auctions are powerful in theory but have been intractable in application because of their combinatorial explosion
- Using mathematical programming to embed bidders' true cost functions may allow for the implicit bidding on and consideration of *all* combinatorial bundles, without the computational cost of explicit bidding (under certain conditions on the true cost function)

Areas for Future Research

- Improvements in truckload procurement model to improve realism, instance size
- Identification of other applications appropriate for this approach
- Evaluation of the impact of stochasticity on the approach
- Consideration of other single-round closed bid auction mechanisms



Questions, Comments, and Concerns