Bidding Languages and Winner Determination for Mixed Multi-unit Combinatorial Auctions

Ulle Endriss
Institute for Logic, Language and Computation
University of Amsterdam

[joint work with Jesús Cerquides, Andrea Giovannucci, and Juan A. Rodríguez-Aguilar (all Barcelona)]

Talk Outline

- Combinatorial Auctions and Mixed Auctions
- Bidding Languages: representing agent preferences
- Winner Determination: problem definition, computational complexity, integer programming formulation
- Conclusions and Future Work

Combinatorial Auctions

In a combinatorial auction, the auctioneer puts several goods on sale and the other agents submit bids for entire bundles of goods.

The number of possible bundles to bid for is exponential in the number of goods. So we need a *bidding language* that can succinctly represent valuations ($\leadsto$ knowledge representation problem).

Given certain bids, the *winner determination problem* (WDP) is the problem of deciding which bids to accept ($\leadsto$ algorithmic problem).

Mixed Auctions

Suppose the auctioneer would like to both sell and buy goods: integrate *direct* and *reverse* auctions.

The auctioneer may be able to *transform* goods: instead of buying a car he may choose to buy certain components and build the car by himself (at a cost). Generalising further, the auctioneer may even solicit *bids for transformations*.

We call the resulting model *mixed auctions*. Here we consider the *multi-unit* variant of the model: there may be several copies of each type of good available.

Transformations

Let G be a finite set of types of goods. A *transformation* is a pair of multisets over G: $(\mathcal{I}, \mathcal{O}) \in \mathbb{N}^G \times \mathbb{N}^G$

"I can deliver $\mathcal{O}$ *after* having received $\mathcal{I}$."

Bidders will be able to offer several such transformations; that is agents will negotiate over *multisets of transformations* $\mathcal{D} \in \mathbb{N}^{(\mathbb{N}^G \times \mathbb{N}^G)}$.

Example: $\{(\{\,\}, \{a\}), (\{b\}, \{c\})\}$ means that the agent in question is able to deliver a (no input required) and that it is able to deliver c if provided with b. Note that this is not the same as $\{(\{b\}, \{a, c\})\}$. In the former case, if another agent is able to produce b if provided with a, we can get c from nothing; in the latter case this would not work.

We define a *subsumption relation* over multisets of transformations: $\mathcal{D} \sqsubseteq \mathcal{D}'$ iff $\mathcal{D}'$ is "no worse than" $\mathcal{D}$ (see paper for precise definition).

Atomic Bids

An *atomic bid* $\text{BID}(\{(\mathcal{I}^1, \mathcal{O}^1), \ldots, (\mathcal{I}^n, \mathcal{O}^n)\}, p)$ specifies a finite multiset of finite transformations and a price.

Under the assumption of *free disposal* at the bidder's side, the bid $Bid = \text{BID}(\mathcal{D}, p)$ defines the following *valuation* function:

$$v_{Bid}(\mathcal{D}') = \begin{cases} p & \text{if } \mathcal{D} \sqsubseteq \mathcal{D}' \\ \perp & \text{otherwise (“undefined”)} \end{cases}$$

To obtain the valuation function defined by Bid without the free disposal assumption, simply replace $\sqsubseteq$ by equality.

The XOR-Language

A suitable *bidding language* should allow bidders to encode choices between alternative bids and the like.

In the paper, we introduce operators such as XOR, OR and AND to combine atomic bids to form complex bid expressions (there are minor but nevertheless interesting differences to the single-unit case).

In the *XOR-language* we only allow XOR-combinations of atomic bids. The intuitive semantics is that at most one of the atomic bids of each bidder can be selected by the auctioneer. Formal semantics:

$$\begin{aligned} v_{Bid}(\mathcal{D}) &= \max\{v_{Bid_i}(\mathcal{D}) \mid i \in \{1..n\}\} \\ &\text{where } Bid = Bid_1 \text{ XOR } \cdots \text{ XOR } Bid_n \end{aligned}$$

That is, XOR simply selects the atomic bid corresponding to the valuation giving maximum profit for the auctioneer.

Expressive Power

Any expression of the language can be translated into an equivalent XOR-combination of atomic bids (see paper). So the XOR-language is as expressive as the full bidding language.

Definition 1 (Finitely-peaked valuation) *A valuation v is called finitely-peaked iff v is only defined over finite multisets of pairs of finite multisets and $\{\mathcal{D} \in \mathbb{N}^{(\mathbb{N}^G \times \mathbb{N}^G)} \mid v(\mathcal{D}) \neq \bot\}$ is finite.*

Proposition 1 (No free disposal) *The XOR-language without free disposal can represent all finitely-peaked valuations, and only those.*

Definition 2 (Monotonic closure) *The monotonic closure $\hat{v}$ of a valuation v is defined as $\hat{v}(\mathcal{D}) = \max\{v(\mathcal{D}') \mid \mathcal{D}' \sqsubseteq \mathcal{D}\}$.*

Proposition 2 (Free disposal) *The XOR-language with free disposal can represent all valuations that are the monotonic closure of a finitely-peaked valuation, and only those.*

WDP: Informal Definition

The *input* to the winner determination problem (WDP) consists of the *bids* submitted by the bidders, a multiset $\mathcal{U}_{in}$ of goods the auctioneer holds initially, and a multiset $\mathcal{U}_{out}$ he is expected to end up with.

A *solution* will be a *sequence* (not a set!) of selected transformations. A *valid* solution has to meet two kinds of conditions:

(1) *Bidder constraints:* The transformations selected have to respect the bids submitted (e.g. the semantics of the XOR-operator).

(2) *Auctioneer constraints:* The sequence has to be implementable:
(a) $\mathcal{U}_{in}$ is a superset of the input of the first transformation;
(b) the set of goods held by the auctioneer after each transformation is a superset of the input of the next one;
(c) the auctioneer holds at least $\mathcal{U}_{out}$ in the end.

An *optimal* solution is a valid solution that maximises the sum of prices associated with the atomic bids selected.

WDP: Computational Complexity

Winner determination is NP-complete for standard CAs (shown via a reduction from SET PACKING). Although intuitively more complicated than standard CAs, mixed auctions are *no worse* than that:

Proposition 3 (Complexity) *Checking whether there exists a valid solution with revenue $\geq K$ is NP-complete for mixed auctions.*

Proof. Mixed auctions can simulate standard CAs $\Rightarrow$ *NP-hardness*. ✓
NP-membership follows from the fact that we can verify a supposed solution in polynomial time. All that is required is checking that the proposed solution is valid and then add up the prices. ✓ □

WDP: Integer Programming Formulation

For mixed auctions, the solution will be a *sequence* of transformations.

Key idea: introduce binary decision variables x^m_{ijk}, where $x^m_{ijk} = 1$ iff the *kth transformation* in the *jth atomic bid* of the *ith bidder* is selected for the *mth position* in the sequence. Additional decision variables with analogous meaning: x^m; x_{ijk}; x_{ij}.

▶ Solving the WDP amounts to solving an integer program:

$$\max \sum_{ij} x_{ij} \cdot p_{ij} \quad \text{subject to various constraints } (\leadsto \text{ next slide})$$

Here p_{ij} is the price of the jth atomic bid of the ith bidder.

Issue: quadratic number of decision variables (wrt. atomic bids)

WDP: Some of our Constraints

It's really a *sequence*! Each transformation can be selected at most once; and at most one transformation can be selected for any one position:

$$\forall ijk: \quad x_{ijk} = \sum_m x^m_{ijk} \qquad \qquad \forall m: \quad x^m = \sum_{ijk} x^m_{ijk}$$

The *bidder constraints* are satisfied! For instance, if the XOR-language is used, we can accept at most one atomic bid from each bidder:

$$\forall i: \quad \sum_j x_{ij} \leq 1$$

The *auctioneer constraints* are satisfied! For instance, sufficient copies of each good are available before each transformation:

$$\forall mg: \quad \mathcal{M}^{m-1}(g) \geq \sum_{ijk} x^m_{ijk} \cdot \mathcal{I}_{ijk}(g)$$

Here $\mathcal{M}^m(g)$ is an integer decision variable representing the number of copies of the good g held by the auctioneer after the mth transformation (can be defined in terms of x^m_{ijk}). The full IP formalisation is in the paper.

Conclusions

- A new auction model: (1) bidding for *transformations* of goods; (2) a solution is a *sequence* rather than a set of atomic bids
- Generalises single/multi-unit direct/reverse combinatorial auctions, supply chain formation auctions, combinatorial exchanges, . . .
- Bidding Languages: expressive completeness results with respect to finitely-peaked valuations
- Winner Determination: no increase in (theoretical) complexity; Integer Programming formulation difficult but possible
- How problematic is the issue with the quadratic number of variables really? Alternatives being explored:
  - Reformulate problem using Petri nets (AAMAS-2007)
  - Constraint programming approach (MSc thesis at UvA)