

Notes on Chapter 1: Bidding Languages by N. Nisan

June 15, 2023

1 Bidding Languages

Most simply Bidding languages deals with the issue of the representation of bids in combinatorial auctions.

Bids can be seen as abstract elements drawn from a space of strategies defined by the auction. In practice, each implementation must determine how each bid can be represented in the program.

We see for a combinatorial auction of m items that there are $2^m - 1$ subsets and thus $2^m - 1$ ways to represent every bid. This is obviously inconvenient for auctions of even a small amount of time and thus there has to be additional thought that goes into bidding languages that can represent common bids succinctly. A different definition of bidding languages is that they simply are a mapping from a mathematical space of bids into the set of finite strings. What this space is does not matter, it could be ASCII characters, boolean sets, etc. Like other languages, we want our bidding language to assign a semantic meaning to each element.

A quick note is that typically for combinatorial auctions we deal with real numbers, and real numbers with infinite precision cannot be represented by any finite string of characters, for the sake of most practical bidding languages we can ignore this concern and instead use an appropriate level of fixed precision. For most practical and monetary purposes a few micro-pennys of precision is enough.

1.1 Bids vs Valuation

As we mentioned above each combinatorial auction bids are defined from some arbitrary space of possible bids. This space of bids is identical to the space of valuations when the mechanism is a direct revelation mechanism. More generally, we see that a player's bid will always be a function of his bid (though maybe not always equal). In the case of bidding languages, all well studied bidding languages are in fact a valuation language, a syntactic representation of valuations. The name bidding language is instead used as it makes clear the purpose of such languages.

Bidding languages offer an additional purpose they allow us to classify valuations according to the complexity. A collection of bidding languages can yield a classification according to which valuations can be succinctly represented. Alongside other semantic classification this becomes a helpful process in understanding bidding languages and determining which makes the most sense for the type of auction.

1.2 Goals of a Bidding Language

The goal of a bidding language is not to succinctly represent all valuations. (This is impossible) to do in length less than exponential to the number of items in our auction, instead it focuses on representing valuations that are *interesting*. Often time one of the largest factors for a bidding language is its ability to simplify valuations. In principle we see that there are two component that are desired for this simplification, first it must look simple to humans or to humans with reasonable help to a computation, and secondly they should be easy to compute and understand. One such case is that it should be easy for winning determination to be done when all the bids are presented.

Given the goals of a bidding language, let us next introduce basic notation. For a combinatorial auction we say it is being done on a set of m items. A valuation v provides the real value $v(S)$ for each subset S of the items. We assume that $v(\emptyset) = 0$ and that for $S \subseteq T$ that $v(S) < v(T)$.

1.3 Example Valuations

1.3.1 Symmetric Valuations

1. **Simple Additive Valuation** Bidder values any subset of k items at value k , $v(S) = |S|$.
2. **Simple Unit Demand Valuation** The bidder desires any single item, and only a single item, and it values it at 1, Thus $v(S) = 1$ for all $S \neq \emptyset$.
3. **Simple K-Budget Valuation** Each set of k items is valued at k as long as no more than k items are obtained. $v(S) = \min(K, |S|)$.
4. **Majority Valuation** Bidder values at 1 any majority ($> \frac{m}{2}$ sized set) of items and 0 for any smaller set.
5. **General Symmetric Valuation** Let $p_1, p_2, p_3, \dots, p_m$ be arbitrary non-negative numbers. The price p_j represents how much the bidder is willing to pay for the j th item won. Thus $v(S) = \sum_{j=1}^{|S|} p_j$.

1.3.2 Asymmetric Valuations

1. **Additive valuation** The bidder has a value v^j for each item j and values a subset of items in it. $v(S) = \sum_{j \in S} v^j$.
2. **Unit Demand Valuation** The bidder has a value v^j and for each item j but desires only a single item. Thus $v(S) = \max_{j \in S} v^j$.
3. **Monochromatic valuation** There are $\frac{m}{2}$ red items and $\frac{m}{2}$ blue items. The bidder requires items of the same color and values each item of the same color 1. Thus the valuation of any set of k blue items and l red items is $\max(k, l)$.
4. **One-of-each-kind valuation** There are $\frac{m}{2}$ pairs of items. The bidding wants one item for each pair and values at 1. Thus the valuation of a set S that contains k complete pairs and l singletons is $k + l$.

1.4 Exclusive and Non-Exclusive Bundle Bids

1.4.1 Basic Bidding Languages

1. **Atomic Bids** Each bidder can submit a pair (S, p) where S is a subset of the items and p is the price he is willing to pay for S . $v(T) = p$ for some $S \subseteq T$ and $v(T) = 0$ otherwise.
2. **OR Bids** Each bidder can submit an arbitrary number of atomic bids, a set of pairs (S_i, p_i) where S_i is a subset of items, and p_i is the maximum price for that subset. OR bid is equivalent to a set of separate atomic bids from different bidders. It is important to note that not all bids can be represented in the OR bidding language.

Definition 1.1. *OR Bids can represent all bids that don't have any substitutios (where $S \cap T = \emptyset$)*

One example that OR bids cannot represent is the simple unit demand valuation on even two items.

3. **XOR bids** Each bidder can submit an arbitrary number of pairs S_i, p_i where S_i is a subset of items and p_i is the maximum he is willing to pay for that subset. Note that this implies that he is willing to obtain at most one of these bids.

Definition 1.2. *XOR-bids can represent all valuation*

Definition 1.3. *The size of a bid is the number of atomic bids in it*

Any additive valuation on m items can be represented by OR bids of size m , the simple additive valuation requires XOR bids of size 2^m .

4. Combination of OR and XOR

OR-of-XOR is a arbitray number of *XOR* bids connected with OR bids, and XOR of OR is an arbitrary number of OR bids connected with XOR bids.

Lemma 1.1. *OR-of-XOR bids can express any downward sloping symmetric valuation on m items in size m^2 (Nisan 2000).*

Lemma 1.2. *The monochromatic valuation requires size of $2 \cdot 2^{\frac{m}{2}}$ in the OR-of-XOR bidding language (Nisan 2000).*

Theorem 1.3. *Fix $K = \sqrt{\frac{m}{2}}$. The K budget valuation requires size of atleast $2^{m^{\frac{1}{4}}}$ in the XOR-of-OR bid language*

1.4.2 OR/XOR

Each bidder may submit a OR/XOR formula specifying his bid. This formula is then used recursively in the usual way of XOR and OR operators to construct a bid. Note that the previous bidding types are just special cases of this.

1.5 OR Bids with Dummy items

Let us now pick our bidding language of our choice, consider OR. We simply allow the bidders to introduce dummy items that hold no value to any participants extending m . In doing so we can represent XOR bids as a series of OR bids with dummy elements.

1.5.1 OR* Bids

Each bidder i can submit an arbitrary number of pairs (S_i, p_i) where each $S_i \subseteq M \cup D$ and p_i is the maximum price he is willing to pay for that subset. Note that this implies he is willing to obtain any number of disjoint bids for the sum of the respective bids.

Theorem 1.4. *The majority valuation requires size of atleast $\binom{m}{\frac{m}{2}}$ in the OR* bid language*

1.6 Computational Complexities

1. **Expression** Given another representation of the valuation v , express v in the bidding language. This depends on this other representation, in most common ways this might be the non-formal ways humans describe this preference or bid.
2. **Winner Determination** Given a set of valuations $\{v_i\}$ find the allocation $\{S_i\}$ that maximizes the total value.
3. **Evaluation** Given a valuation v and a "query" answer the query, this could be the value query or the demand or some other simple query.

Note that even for the simplest bidding language, that only allows atomic bids, Winner determination is a NP-Complete Determination problem.

Overall, numerous bidding languages have been constructed that attempt to meet the general needs of modeling asymmetric and symmetric valuations while also providing simple and iterable results for use in computation like the winning determination, while such languages do exist, no such language meets all the requirements one might want and thus there is some degree of thought that must go into choosing a bidding language that meets the need of the combinatorial auction.