

## Approximation Algorithms for Combinatorial Auctions

Today, we will consider a subclass of combinatorial auctions. Our goal will be to design mechanisms that are truthful, run in polynomial time, and yield good approximation guarantees.

The subclass will be a single-parameter settings, for which we already know by Myerson's Lemma that we have to confine ourselves to monotone outcome rules.

## 1 Combinatorial Auctions

Recall combinatorial auctions.

**Definition 12.1** (Combinatorial Auction). *In a combinatorial auction, a set of  $m$  items  $M$  shall be allocated to a set of  $n$  bidders  $\mathcal{N}$ . The bidders have private values for bundles of items. The goal is to maximize social welfare.*

- *Feasible allocations:*  $X = \{(S_1, \dots, S_n) \in (2^M)^n \mid S_i \cap S_j = \emptyset, i \neq j\}$
- *Valuation functions:*  $v_i: 2^M \rightarrow \mathbb{R}_{\geq 0}$ ,  $i \in \mathcal{N}$  (private)
- *Objective:* Maximize social welfare  $\sum_{i=1}^n v_i(S_i)$

We will generally assume free disposal, i.e.,  $v_i(S) \geq v_i(T)$  for  $T \subseteq S$ , and that valuations are normalized, i.e.,  $v_i(\emptyset) = 0$ .

We will focus on the case where each bidder is interested in a single bundle of items. We will call these bidders single minded.

**Definition 12.2** (Single-Minded Bidders). *Bidders are called single minded if, for every bidder  $i \in \mathcal{N}$ , there exists a bundle  $S_i^* \subseteq M$  and a value  $v_i^* \in \mathbb{R}_{\geq 0}$  such that*

$$v_i(T) = \begin{cases} v_i^* & \text{if } T \supseteq S_i^*, \\ 0 & \text{otherwise.} \end{cases}$$

We call a bidder that is granted his bundle a winner, and we say that this bidder wins the bundle.

We will further assume that the bundle  $S_i^*$  that bidder  $i$  is interested in is *public* and only the valuation  $v_i^*$  is *private*. This turns the problem into a single-parameter problem, to which our previous results apply.

**Example 12.3** (Single-Minded Combinatorial Auction). *There are two items  $a$  and  $b$  and three bidders Red, Green, and Blue. Red has a value of 10 for  $\{a\}$ , Green has a value of 14 for the set  $\{a, b\}$ , and Blue has a value of 8 for  $\{b\}$ . Social welfare is maximized by allocating  $\{a\}$  to Red and  $\{b\}$  to Blue.*

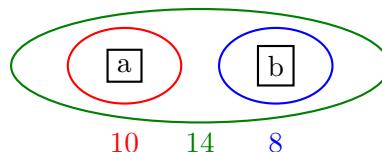


Figure 1: Single-minded CA instance from Example 12.3. The items are shown as black rectangles and the bundles as color-coded ellipses.

## 2 Hardness and Hardness of Approximation

A first observation is that we cannot hope to get an exact solution because the allocation problem is NP-hard. Note that this is purely an optimization question, the incentives do not matter at all.

**Theorem 12.4** (Lehmann, O'Callaghan, Shoham 1999). *The allocation problem among single-minded bidders is NP-hard.*

*Proof sketch.* We will prove the claim by reduction from independent set. Consider a graph  $G = (V, E)$ . Each node is represented by a bidder. Each edge is represented by an item. For bidder  $i$ , set  $S_i^* = \{e \in E \mid i \in e\}$  and  $v_i^* = 1$ .

Note that a set of bidders  $W$  corresponds to an independent set if and only if their sets  $S_i^*$  are disjoint, that is, if and only if  $W$  is a feasible set of winners. This implies that there is an independent set of size  $x$  if and only if there is an allocation (i.e. a set of winners  $W$ ) such that  $\sum_{i \in W} v_i^* = x$ .  $\square$

Due to this hardness result, we will consider approximation algorithms. We call an algorithm an  $\alpha$ -approximation, if for the solution  $x$  computed by the algorithm on input  $(v_i)_{i \in \mathcal{N}}$ , we have  $\sum_{i \in \mathcal{N}} v_i(x) \geq \frac{1}{\alpha} \max_{x^*} \sum_{i \in \mathcal{N}} v_i(x^*)$ .

Unfortunately, the same reduction actually implies a hardness of approximation result in terms of the number of items  $m$ . A more recent results shows a lower bound in terms of the maximum bundle size of any bidder,  $d = \max_i |S_i^*|$ .

**Theorem 12.5** (Lehmann, O'Callaghan, Shoham 1999; Håstad 1999). *There is no polynomial-time algorithm for approximating the optimal allocation among single-minded bidders to within a factor of  $m^{1/2-\epsilon}$ , for any  $\epsilon > 0$ , unless  $\text{NP} = \text{ZPP}$ .*

**Theorem 12.6** (Hazan et al. 2006). *Approximating the optimal allocation among single-minded bidders to within a factor of  $O\left(\frac{d}{\log d}\right)$ , is NP-hard.*

The class ZPP, for zero-error probabilistic polynomial time, is the subclass of NP consisting of those sets  $L$  for which there is some constant  $c$  and a probabilistic Turing machine  $\mathcal{M}$  that on input  $x$  runs in expected time  $O(|x|^c)$  and outputs 1 if and only if  $x \in L$ . More important for our purposes than the precise definition of the complexity class ZPP, is the fact that a conditional hardness result based on the assumption that  $\text{ZPP} \neq \text{NP}$  is considered strong evidence of computational intractability.

## 3 Greedy Mechanisms for Single-Minded CAs

A natural question in light of the hardness results is whether we can find polynomial-time algorithms that match the lower bounds. In particular, is there a separation between the best algorithm subject to polynomial-time and the best monotone algorithm?

The answer to this question due to Lehmann, O'Callaghan, and Shoham is one of the foundational results of the field Algorithmic Game Theory: With respect to both parameters, the total number of items and the maximum bundle size, simple monotone greedy algorithms yield optimal approximation results.

Both algorithms use a carefully designed scoring function to rank the bidders. They then go through the bidders and greedily accept the next bidder in the ranked list, removing all future bidders that conflict with it.

Before we discuss the two algorithms let us first recall what truthful payments in a monotone algorithm for a setting like ours should look like.

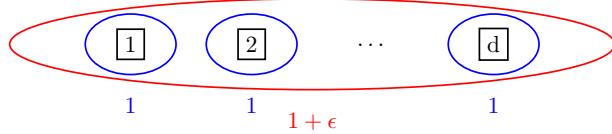


Figure 2: Challenge instance for Greedy-by-Value

**Definition 12.7** (Threshold Payments). *For an allocation rule for the single-minded CA problem denote by  $W(b)$  the set of winners when the bids are  $b$ . If the allocation rule is monotone we define the threshold bid  $\tau_i(b_{-i})$  for player  $i$  against bids  $b_{-i}$  of the bidders other than  $i$  as the smallest bid such that  $i \in W(b_i, b_{-i})$ , that is*

$$\tau_i(b_{-i}) = \inf\{b_i \mid i \in W(b_i, b_{-i})\} .$$

We first consider the algorithm that yields a good approximation with respect to the maximum bundle size  $d = \max_{i \in \mathcal{N}} |S_i^*|$ .

### Greedy-by-Value

1. Re-order the bids such that  $b_1^* \geq b_2^* \geq \dots \geq b_n^*$ .
2. Initialize the set of winning bidders to  $W := \emptyset$ .
3. For  $i = 1$  to  $n$  do: If  $S_i^* \cap \bigcup_{j \in W} S_j^* = \emptyset$ , then  $W := W \cup \{i\}$ .

**Example 12.8.** Consider the instance from Example 12.3. The ranking computed by Greedy-by-Value is Green, Red, Blue. Green is considered first and accepted, which leads to the removal of both Red and Blue. Green's threshold bid is 10.

**Proposition 12.9** (Folklore). *Greedy-by-Value is a  $d$ -approximation. It is monotone, so charging threshold bids yields a truthful mechanism.*

*Proof.* It is not difficult to see that the Greedy-by-Value algorithm is monotone. For every bidder  $i$  fixing the bids  $b_{-i}^*$  of the bidders other than  $i$ , player  $i$ 's outcome is determined by the position in the sorted list of bids of the other players. By increasing his bid  $b_i^*$ , bidder  $i$  can only move further to the front of the sorted list of all bids.

The approximation guarantee follows by a simple charging argument. Let  $W$  be the set of bidders selected by the algorithm and let  $OPT$  be the optimal solution. For  $i \in W$ , let

$$OPT_i = \{j \in OPT, j \geq i \mid S_i^* \cap S_j^* \neq \emptyset\} .$$

That is,  $OPT_i$  contains the indices of the bidders  $j \geq i$  that are in  $OPT$  and get blocked if we accept  $i$ . Note that if  $i \in OPT$  then  $OPT_i = \{i\}$ . Each  $j \in OPT$  is included in at least one set  $OPT_i$  for  $i \in W$  for the following reason: If  $j \notin OPT_i$  for  $i \in W$  with  $i < j$ , then all items in  $S_j^*$  are still available when reaching bidder  $j$  in the execution. So,  $j$  would be accepted by the algorithm and, hence,  $j \in W$  and  $j \in OPT_j$ . Therefore, we can write

$$\sum_{j \in OPT} b_j^* \leq \sum_{i \in W} \sum_{j \in OPT_i} b_j^* .$$

Next, we have  $|OPT_i| \leq |S_i^*| \leq d$ . This is due to the fact that the sets  $S_j^*$  for  $j \in OPT_i$  are disjoint but each have a non-empty intersection with  $S_i^*$ . Furthermore, by the greedy ordering  $b_j^* \leq b_i^*$  for  $j \in OPT_i$ . Therefore

$$\sum_{j \in OPT} b_j^* \leq \sum_{i \in W} \sum_{j \in OPT_i} b_j^* \leq \sum_{i \in W} d \cdot b_i^* = d \cdot \sum_{i \in W} b_i^* .$$

□

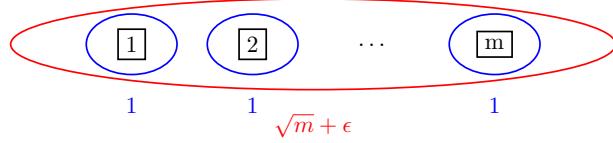


Figure 3: Challenge instance for Greedy-by-Sqrt-Value-Density

That the approximation guarantee can be as bad as  $d$  can be seen from examples such as the one in Figure 2. Assume w.l.o.g. that  $m$  is a multiple of  $d$ . Every set of  $d$  items is wanted by a distinct “big” bidder, who has a value of  $1 + \epsilon$  for it. Each of the  $d$  items this bidder is interested in is requested by a distinct “small” bidder, each of which has a value of 1. Greedy-by-Value will accept all the big bidders resulting in welfare  $m/d \cdot (1 + \epsilon)$ , while accepting all small bidders would have social welfare of  $m$ .

The same example that we used to establish a lower bound of  $d$  for Greedy-by-Value, also shows a lower bound of  $m$ . This is considerably worse than our lower bound of  $\sqrt{m}$  on what can be achieved with a polynomial-time algorithm.

Our next algorithm avoids the trap in which our Greedy-by-Value algorithm stepped by normalizing bids with their bundle size. More specifically, it divides each bid by the square root of the bundle size.

### Greedy-by-Sqrt-Value-Density

1. Re-order the bids such that  $\frac{b_1^*}{\sqrt{|S_1^*|}} \geq \frac{b_2^*}{\sqrt{|S_2^*|}} \geq \dots \geq \frac{b_n^*}{\sqrt{|S_n^*|}}$ .
2. Initialize the set of winning bidders to  $W := \emptyset$ .
3. For  $i = 1$  to  $n$  do: If  $S_i^* \cap \bigcup_{j \in W} S_j^* = \emptyset$ , then  $W := W \cup \{i\}$ .

**Example 12.10.** Consider again the instance from Example 12.3. The ranking computed by Greedy-by-Sqrt-Value-Density is  $10 \geq 14/\sqrt{2} \geq 8$ . So Red is considered first and accepted. This leads to the removal of Green. Afterwards Blue is accepted. The threshold bid for Red is  $14/\sqrt{2}$ , for Blue it is zero.

**Theorem 12.11** (Lehmann, O’Callaghan, Shoham 1999). *Greedy-by-Sqrt-Value-Density is a  $\sqrt{m}$ -approximation. It is monotone, so charging threshold bids makes it a truthful mechanism.*

*Proof.* That Greedy-by-Sqrt-Value-Density is monotone can be shown by essentially the same argument that showed that Greedy-by-Value is monotone. Holding a bidder and the bids of the other bidders fixed, the bidder faces a ranked list of bids. Its position in this sorted list determines whether he wins or not. A higher bid can only improve its position.

To establish an upper bound on the approximation guarantee we again write  $W$  and  $OPT$  for the set of winners selected by the algorithm and the optimal one. Again, we define

$$OPT_i = \{j \in OPT, j \geq i \mid S_i^* \cap S_j^* \neq \emptyset\} .$$

And we can write

$$\sum_{j \in OPT} b_j^* \leq \sum_{i \in W} \sum_{j \in OPT_i} b_j^* .$$

So, if we can show  $\sum_{j \in OPT_i} b_j^* \leq \sqrt{m} \cdot b_i^*$ , we are done.

As  $b_j^* \leq \sqrt{|S_j^*|} \cdot b_i^* / \sqrt{|S_i^*|}$ , for  $j \in OPT_i$ , we obtain

$$\sum_{j \in OPT_i} b_j^* \leq \frac{b_i^*}{\sqrt{|S_i^*|}} \cdot \sum_{j \in OPT_i} \sqrt{|S_j^*|}$$

Next we will show that  $\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{m} \cdot \sqrt{|S_i^*|}$ . To this end, we will use that the function  $\sqrt{\cdot}$  is concave. Therefore, by Jensen's inequality, we have for all  $y_1, \dots, y_\ell \geq 0$  that  $\frac{1}{\ell} \sum_{k=1}^\ell \sqrt{y_k} \leq \sqrt{\frac{1}{\ell} \sum_{k=1}^\ell y_k}$  and therefore  $\sum_{k=1}^\ell \sqrt{y_k} \leq \sqrt{\ell} \sum_{k=1}^\ell y_k$ . So, we get

$$\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{|OPT_i| \cdot \sum_{j \in OPT_i} |S_j^*|}.$$

Now  $|OPT_i| \leq |S_i^*|$  since every  $S_j^*$ , for  $j \in OPT_i$ , intersects  $S_i^*$  and these intersections are disjoint. Furthermore,  $\sum_{j \in OPT_i} |S_j^*| \leq m$  since  $OPT_i$  is an allocation.

We obtain,

$$\sum_{j \in OPT_i} b_j^* \leq \frac{b_i^*}{\sqrt{|S_i^*|}} \cdot \sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \frac{b_i^*}{\sqrt{|S_i^*|}} \cdot \sqrt{|OPT_i|} \cdot \sqrt{\sum_{j \in OPT_i} |S_j^*|} \leq b_i^* \sqrt{m} .$$

To obtain a lower bound of  $\sqrt{m}$  on the approximation guarantee we consider instances such as the one given in Figure 3. There is one “big” bidder with a bundle size of  $m$  and a value of  $\sqrt{m} + \epsilon$  and  $m$  bidders, one for each item, with a bundle size and a value of 1. Greedy-by-Sqrt-Value-Density accepts the big bidder for a social welfare of  $\sqrt{m} + \epsilon$ , while accepting all small bidders would have led to a social welfare of  $m$ .  $\square$

We conclude that with respect to both quality measures, number of items  $m$  and maximum bundle size  $d = \max_i |S_i^*|$ , insisting on monotonicity did not lower our ability to obtain a near optimal outcome.

## References

- D. Lehmann, L. I. O’Callaghan, Y. Shoham. Truthful Revelation in Approximately Efficient Combinatorial Auctions. STOC 1999 and JACM 2002. (Greedy mechanism for single-minded CAs)
- J. Håstad. Clique is hard to approximate withing  $n^{1-\epsilon}$ . Acta Math., 182:105–142, 1999. (Hardness of approximation for single-minded CAs when the parameter is number of items)
- E. Hazan, S. Safra, O. Schwartz. On the complexity of approximating k-set packing. Computational Complexity, 15(1):20–39, 2006. (Hardness for single minded CAs with bounded bundle size)