Algorithmic Game Theory | Autumn 2021, Week 7

Incentives vs Computation

ETH Zürich | Paolo Penna

Previous lecture: VCG mechanisms cannot solve all problems. For example, minimizing the makespan in related machines can be done with the one-parameter/monotone characterization of truthful mechanisms. For other problems, unrelated machines, we cannot attain optimal solutions with truthful mechanisms.

This lecture: In this lecture we consider combinatorial auctions, a class of problems where truthfulness by itself is not a problem (exact mechanisms exist using VCG). However, the problem is computationally difficult, so if we want *polynomial running time*, we have to use *approximation algorithms*. What we study today is whether such algorithms lead to a truthful mechanism.

Can we get *truthful approximation mechanisms?*

We use the characterization for one-parameter setting (Myerson's Lemma) which says that we need *monotone approximation algorithms*, and this is enough. Does the monotonicity requirement limit our ability to achieve near-optimal outcomes in polynomial time?

1 Combinatorial Auctions

We will study the tradeoff between incentives and computation through one of the canonical problems in mechanism design.

Definition 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:* $\mathcal{A} = \{(S_1, \ldots, S_n) \subseteq M^n \mid S_i \cap S_j = \emptyset, i \neq j\}$
- *Valuation functions (private):* $v_i : 2^M \to \mathbb{R}_{\geq 0}$*, for every player* i
- *Objective: Maximize social welfare* $\sum_{i=1}^n v_i(S_i)$

We also make two natural assumptions: 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:

Definition 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 **one-parameter** problem, to which our previous results apply.

Example 3 (single-minded CA)**.** *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.*

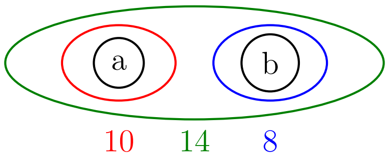


Figure 1: Single-minded CA instance from Example 3. The items are shown as black circles and the bundles as color-coded ellipses.

Exact Mechanisms Exist (VCG) The CA problem (Definition 1) asks to **maximize** the **sum** of all players **valuations**,

$$SW(a, v) := \sum_i v_i(a)$$

where $v_i(a)$ is the valuation of player i for allocation $a \in \mathcal{A}$. The truthful VCG mechanisms we presented in the previous lectures for players with private costs, is actually maximizing the social welfare if we consider

- cost $\leftrightarrow$ valuation
- minimize sum costs (social cost) $\leftrightarrow$ maximize sum valuations (social welfare)

A rewriting of the mechanism for valuations is in Appendix A.

Observation 4. *VCG mechanisms maximize the social welfare and are truthful, even for the (general) CA in Definition 1.*

2 Hardness and Hardness of Approximation

A first observation is that the VCG mechanism, which maximizes social welfare is not a viable solution. The reason is that its algorithm should maximize (exactly) the social welfare, which turns out to be an NP-hard problem.

Theorem 5 (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 a good.
- For bidder i, set $S_i^* = \{e \in E \mid i \in e\}$ and $v_i^* = 1$.

This way, winning bidders correspond to nodes in an independent set. □

The same reduction actually implies a **hardness of approximation** result in terms of the number of items m. A more recent result shows a lower bound in terms of the maximum bundle size of any bidder,

$$d := \max_i |S_i^*|.$$

Theorem 6 (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 $\mathsf{NP} = \mathsf{ZPP}$.*

Theorem 7 (Hazan et al. 2006)**.** *Approximating the optimal allocation among single-minded bidders to within a factor of $\Omega\left(\frac{d}{\log d}\right)$, is NP-hard.*

Intuitively, $\mathsf{NP} = \mathsf{ZPP}$ means that we have an efficient *randomized* algorithm for every problem in NP, which is considered almost as unlikely as $\mathsf{NP} = \mathsf{P}$.[1] For our purposes, this means that there is a strong evidence that we cannot approximate the social welfare.

Goal 1: $O(d)$-approximation (trufhful + polynomial time)?
Goal 2: $O(\sqrt{m})$-approximation (trufhful + polynomial time)?

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?

3 Greedy Mechanisms for Single-Minded CAs

We show that with respect to both parameters, the total number of items and the maximum bundle size, simple monotone greedy algorithms yield optimal approximation results. Recall that our restricted version of CA is a one-parameter problem, so this is enough to get truthful mechanism:

Truthful ⇔ Monotone

[1] 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 $x \in L$.

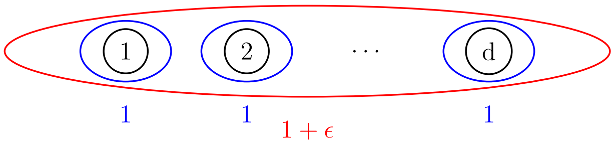
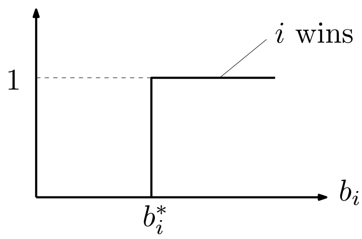


Figure 2: A "bad" instance for Greedy-by-Value

Here monotone means that, if bidder i wins for a bid b_i, then he/she still win when increasing his/her bid and the others do not change their bids:



(The y-axis indicates whether i wins or not.) Given any such monotone algorithm, the payments to obtain a truthful mechanism are very simple.

Definition 8 (threshold payments)**.** *For an allocation rule (algorithm) 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 b_i^* *for player i against bids b_{-i} of the bidders other than i as the smallest bid such that $i \in W(b_i^*, b_{-i})$.*

Example 9. *Suppose we want to allocate k identical items to n bidders, each bidder is interested in a single copy of the item. The k-highest bids are the winners (each of these bidders get one item), and each of them pays the $k+1$-highest bid.*

Both greedy algorithms below 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.

3.1 Truthful $O(d)$-approximation

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 $v_1^* \geq v_2^* \geq \cdots \geq v_n^*$.
2. Initialze 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 10. *Consider the instance from Example 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.

Theorem 11 (folklore)**.** *Greedy-by-Value is a $\Theta(d)$ approximation. It is monotone, so charging threshold bids yields a truthful mechansm.*

Proof of Monotonicity. For every bidder i fixing the bids v^*_{-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/her bid v^*_i, bidder i can only move further to the front of the sorted list of all bids. That is, if i wins for v^*_i, then he/she also wins for $v'_i > v^*_i$.

More in detail, consider the execution of the algorithm for v^*_i. The set of bidders added to the winning set $W = W(v^*_i, v^*_{-i})$ are

$$\mathsf{w}_1, \mathsf{w}_2, \ldots, \mathsf{w}_{t-1}, \underbrace{\mathsf{w}_t}_{i}, \ldots$$

and each time we add some winner w_k we also "discard" a subset of conflicting bidders from the list

$$D_{\mathsf{w}_1}, D_{\mathsf{w}_2}, \ldots, D_{\mathsf{w}_{t-1}}$$

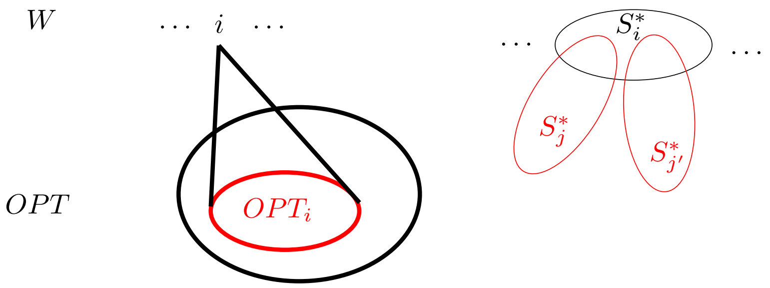
where each D_{w_k} contains those r such that S^*_r intersects $S^*_{\mathsf{w}_k}$ (see Step 3 of the algorithm).

Note that i does not conflict with any of $\mathsf{w}_1, \mathsf{w}_2, \ldots, \mathsf{w}_{t-1}$. So, when i increases her valuation to some $v'_i > v^*_i$, its position moves further to the front of the list, and the algorithm will include some of these bidders:

$$\mathsf{w}_1, \mathsf{w}_2, \ldots, \mathsf{w}_{t'-1}, \underbrace{\mathsf{w}_t}_{i}, \mathsf{w}_{t'} \ldots, \mathsf{w}_{t-1}, \ldots,$$

That is, if i wins for v^*_i then he/she also wins for $v'_i > v^*_i$. □

Proof of Approximation. The approximation guarantee follows by a simple charging argument. Every bidder $i \in W$ can block at most d bidders $j \in OPT$, because OPT is a feasible allocation:



Since we are ranking by non-increasing value each such bidder i must have a value v^*_i that is at least as high as the value v^*_j of the bidders $j \in OPT_i$ that it blocks, thus implying

$$d \cdot v^*_i \geq \sum_{j \in OPT_i} v^*_j$$

Every element $j \in OPT$ is either blocked by some i, or it is also in W. Therefore,

$$d \cdot \sum_{i \in W} v^*_i \geq \sum_{j \in OPT} v^*_j \Leftrightarrow \sum_{i \in W} v^*_i \geq \frac{1}{d} \cdot \sum_{j \in OPT} v^*_j \ .$$

□

Observation 12. *Greedy-by-Value is not better than m-approximate (see Figure 2).*

3.2 Truthful $O(\sqrt{m})$-approximation

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-Value-Density

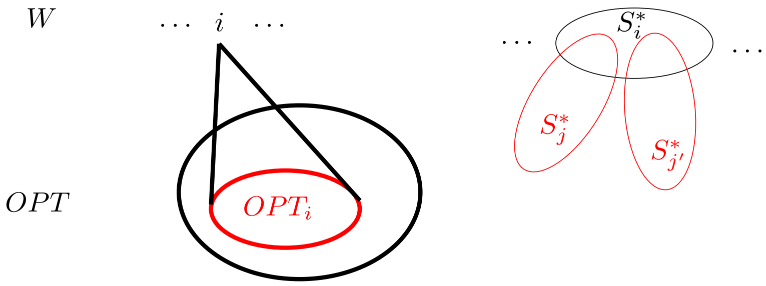
1. Re-order the bids such that $\frac{v_1^*}{\sqrt{|S_1^*|}} \geq \frac{v_2^*}{\sqrt{|S_2^*|}} \geq \cdots \geq \frac{v_n^*}{\sqrt{|S_n^*|}}$.
2. Initialze the set of winning bidders to $W = \emptyset$.
3. For $i = 1$ to n do: If $S_i^* \cap \bigcup_{j \in W} S_j^* = \emptyset$, then $W = W \cup \{i\}$.

Example 13. *Consider again the instance from Example 3. The ranking computed by Greedy-by-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 14 (Lehmann, O'Callaghan, Shoham)**.** *Greedy-by-Value-Density is a* $\Theta(\sqrt{m})$ *approximation. It is monotone, so charging threshold bids make it a truthful mechanism.*

Proof of Monotonicity. We can use 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. □

Proof of Approximation. Like in the proof of Theorem 11, we start by observing that every $i \in W$ blocks some subset OPT_i of bidders in OPT:



The algorithm includes i instead of $j \in OPT_i$ because $v_j^* \leq \sqrt{|S_j^*|} \cdot v_i^* / \sqrt{|S_i^*|}$. Therefore we obtain

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

Next we will show that $\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{m} \cdot \sqrt{|S_i^*|}$. By the Cauchy-Schwarz inequality,[2]

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

[2] In general, $(a_1 b_1 + \cdots + a_k b_k)^2 \leq (a_1^2 + \cdots + a_k^2)(b_1^2 + \cdots + b_k^2)$. Thus, $(a_1 + \cdots + a_k)^2 \leq (a_1^2 + \cdots + a_k^2) \cdot k$.

Since OPT_i is a feasible allocation $\sum_{j \in OPT_i} |S_j^*| \leq m$, because these sets S_j^* are pairwise disjoint. For the same reason, we can obtain $|OPT_i| \leq |S_i^*|$ as follows: every S_j^* intersects S_i^* and these intersections are disjoint (see upper right picture). This proves

$$\sum_{j \in OPT_i} v_j^* \leq v_i^* \sqrt{m} \ .$$

As element $j \in OPT$ is either blocked by some i, or it is also in W, we get

$$\sum_{j \in OPT} v_j^* \leq \sum_{i \in W} \sum_{j \in OPT_i} v_j^* \leq \sqrt{m} \cdot \sum_{i \in W} v_i^* \ .$$

This prove the $O(\sqrt{m})$ upper bound on the approximation ratio of Greedy-by-Value-Density. This is also a lower bound (**Exercise** 1), so the approximation is precisely $\Theta(\sqrt{m})$. □

Exercise 1. *Show that the analysis of the theorem above is tight. That is, the approximation ratio of Greedy-by-Value-Density is at least* $\Omega(\sqrt{m})$.

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.

Best **truthful** approximation $\simeq$ Best approximation

The two monotone greedy algorithms answer a fundamental question in mechanism design for the CAs we considered, namely, if *truthfulness* prevents us from obtaining the same results that any polynomial-time algorithm can achieve.

Related Literature

The results in this lecture appeared in these works:

- 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)

There is also a new family of mechanisms, called *deferred acceptance auctions*, which use the idea of ranking bidders (in some way). Instead of adding bidders to extend a feasible solution, these mechanisms start from an unfeasible solution and remove bidders until the winners form a feasible set. We shall see these mechanismsin future lectures.

A significant part of this notes is from previous notes by Paul Dütting available here:

- http://www.cadmo.ethz.ch/education/lectures/HS15/agt_HS2015/

which also include the definition and examples of deferred acceptance auctions.

A VCG mechanisms (again)

In auctions it is natural to speak about **valuations** and **bids**, instead of true costs and reported costs. So here we restate the construction of truthful (VCG) mechanisms in this terminology.

A **mechanism** is a pair (A, P) which on input the bids $b = (b_1, \ldots, b_n)$ reported by the players (bidders), outputs

- A solution $A(b) \in \mathcal{A}$;
- A payment $P_i(b)$ that player i pays to the mechanism.

The corresponding **utility** for each agent i is

$$u_i(b|v_i) := v_i(A(b)) - P_i(b).$$

Instead of minimizing the social cost, we say we want to **maximize** the **social welfare**:

$$SW(a, v) := \sum_i v_i(a)$$

and the optimum social welfare is

$$opt_{SW}(v) := \max_{a \in \mathcal{A}} SW(a, v)$$

A **VCG mechanism** is a pair (A, P) such that

- A in an optimal algorithm:
$$SW(A(b), b) = opt_{SW}(b) \qquad \text{for all } b;$$
- P is of the following form:
$$P_i(b) = Q_i(b_{-i}) - \sum_{j \neq i} b_j(A(b))$$
where Q_i is an arbitrary function independent of b_i.

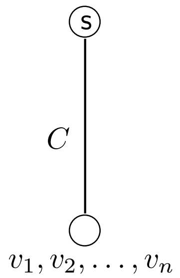
Once again, VCG mechanisms are **truthful**. Either re-do the proof in the previous lectures, or observe that $c_i(a) = -v_i(a)$ and minimize the social cost is the same as maximize the social welfare.

Remark 1. *This version of VCG is the standard one you find in the literature. We presented the version for costs first to explain the main ideas using the shortest-path problem.*

Exercises
(during next exercise class - 9.11.2021)

We shall discuss and solve together these two exercises.

Exercise 2 (Exercise 3 from Set 5 reloaded). *Consider the following problem: There are n users (players) potentially interested in receiving a TV transmission, and the transmission is sent from a server* s *over this simple network:*



*All users are located in the same node, and $C > 0$ is the **cost** for sending the transmission to one or more users:*

> The server s can select which users receive the transmission (and which do not). If one or more receive, s has cost exactly C in total (the use of the link). If no user receives, the cost for s is 0.

*Each user has a **private valuation** v_i, that is, how much he/she is willing to pay for the transmission. Users can chet and report a different valuation r_i. The **utility** of user i is given by the difference between the valuation and the amount of money he/she has to pay:*

$$-P_i + \begin{cases} v_i & \text{if } i \text{ receives the transmission} \\ 0 & \text{otherwise} \end{cases}$$

*Let us consider the variant in which the service can be **provided to several users** (a subset S of users receives the transmission). Natural questions:*

1. *Who should get the service?*
2. *How much should we charge?*

If you give the service to one or more users, the cost is C as above, and it is 0 if none gets served. Two natural things we may want to have:

> *I) Maximize the **(economic) efficiency** of our solution: Sum of players valuations - cost for providing the service.*
> *II) **Budget-balance**: Sum of payments = cost of transmission.*

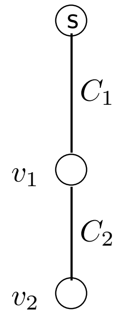
*Let us try to build **truthful** mechanisms that achieve one or both the above goals. Truthful means that reporting the true valuation v_i maximizes i's utility (no matter how we fix the reports r_j of the other users).*

Perhaps the following conditions are also natural

> *III) **No positive transfer**: We do not pay the players.*
> *IV) **Consumer sovereignty**: We do not exclude a priory any player (if i bids high enough gets serviced).*
> *V) **Voluntary partecipation**: We never charge more than (reported) valuation*

Can you also get these conditions?

Exercise 3. *Reconsider truthful mechanisms and **economic efficiency** for this variant:*



Now to serve player 2 (or both) the cost is $C_1 + C_2$, while to service player 1 only the cost is C_1.

- *What does the algorithm optimize?*
- *How can we do that?*
- *Describe the payments to get a truthful mechanism.*