Algorithmic Game Theory

Truthful Mechanisms for Welfare Maximization

Vangelis Markakis
markakis@gmail.com

Designing welfare maximizing truthful auctions for single parameter environments

Single parameter auctions

- For the single-item case, we saw that the Vickrey auction is ideal
- We would like to achieve the same properties for any other type of auction
  - truthfulness and individual rationality [incentive guarantees]
  - welfare maximization [economic performance guarantees]
  - implementation in polynomial time [computational performance guarantees]
- Can we achieve all 3 properties for any single-parameter environment?

Knapsack auctions

- We will see an illustration for knapsack auctions
- k identical items for sale
- Each bidder i has a publicly known demand for w_i items
  - Inelastic demand
  - The mechanism should either give w_i items to the bidder or should not give him anything
- Each bidder i submits a bid b_i for his value per unit
- Real value per unit = v_i
- Assume the demands $(w_1, w_2, ..., w_n)$ are known to the mechanism
  - Say bidders have no incentive to lie about them
- Only private information to bidder i is v_i

Knapsack auctions

Alternative view of knapsack auctions

- The auctioneer has a resource of total capacity k (a knapsack)
- Each bidder requires size w_i , if he is served
- Each bidder has a value v_i w_i , if he is served
- The auctioneer needs to select a subset of bidders to serve so as not to exceed the capacity k

Feasible allocations:

- $(x_1, x_2, \ldots, x_n)$ with $x_i \in \{0, 1\}$, and $\Sigma_i w_i x_i <= k$
- Just like the feasible solutions of a knapsack problem

Knapsack auctions

Example

- Resource = the half-time break in the Champions League final
- Capacity k = total length of the break
- Each bidder corresponds to a company who wants to be advertised during the break
- The size w_i is the duration of the ad of bidder i
- The auctioneer needs to select a subset of bidders as winners and present their ads without exceeding the time capacity k

Knapsack auctions

- Let **b** = (b_1, b_2, ..., n_n) be the biding vector
- Need to decide the allocation and payment rule
- For the allocation rule:
  - Think of maximizing the social welfare
  - Then we have precisely the 0-1 Knapsack problem!

$$\max \sum_i b_i x_i$$

s.t.

$$\sum_i w_i x_i <= k$$

$$x_i \in \{0, 1\}, \text{ for } i = 1, \ldots, n$$

Knapsack auctions

Claim: The allocation rule that maximizes the social welfare is monotone

- Consider a winner and see what can happen if he increases his bid

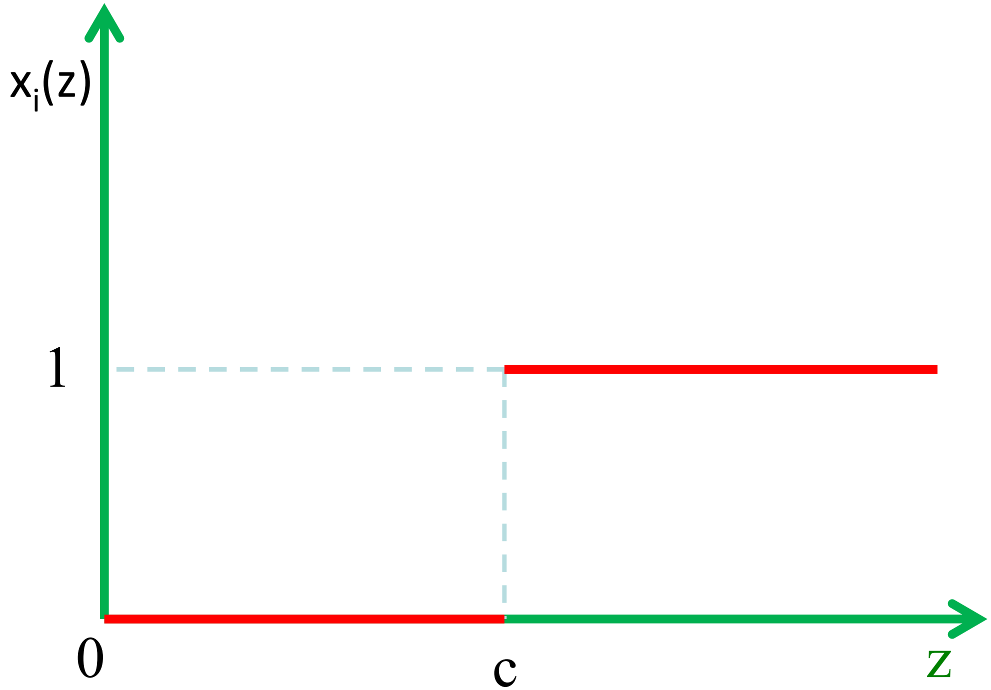
Hence, we can apply Myerson's lemma

How many jumps can we have for the allocation of a single player?

- At most one, a player can jump from being a loser ($x_i = 0$) to being a winner ($x_i = 1$)

Myerson's lemma and knapsack auctions

- The jump for a winner i happens at i's *critical bid*: the minimum he could bid and still be a winner, also known as *threshold bid*
- Generalization of the payment in Vickrey auction



Final mechanism:

- Solve the knapsack problem and find an optimal solution
- Give to each winner i, the requested number of items w_i
- Charge the winners their **critical bid**

Myerson's lemma and knapsack auctions

Does this mechanism achieve the desirable properties we wanted?

- truthfulness [YES]
- welfare maximization [YES]
- implementation in polynomial time [?]

•Knapsack is an NP-complete problem

•The properties can be enforced only for special cases where Knapsack is easy

- If highest bid or highest demand is polynomial in n (by dynamic programming)
- If weights form a super-increasing sequence

Algorithmic Mechanism Design

- The requirement for low complexity usually comes in conflict with the other criteria
- Goal of algorithmic mechanism design: explore the trade-offs between the 3 main properties (or any other properties that we may require in a given setting)
  - Truthfulness
  - welfare maximization
  - implementation in polynomial time
- Approach: relax one of the criteria and see if we can achieve the others
- For Knapsack and in general whenever welfare maximization is NP-complete: resort to approximation algorithms

Knapsack auctions

Goal for Knapsack:

- Find an approximation algorithm for the social welfare
- Prove that it is monotone

Recall:

Definition: An algorithm *A*, for a maximization problem, achieves an approximation factor of γ ($\gamma \leq 1$), if for every instance I of the problem, the solution returned by *A* satisfies:

$$SOL(I) \geq \gamma \, OPT(I)$$

Where OPT(I) is the value of the optimal solution for instance I

Knapsack auctions

- There are several heuristics and approximation algorithms for Knapsack, but not all of them are monotone
- A greedy ½-approximation:
  - For each bidder i, we care to evaluate the quantity b_i/w_i
  - Intuitively, we prefer bidders with small size/demand and large value
- Step 1: Sort and re-index the bidders so that

$$b_1/w_1 \geq b_2/w_2 \geq \ldots \geq b_n/w_n$$

- Step 2: Pick bidders in that order until the first time that adding someone exceeds the knapsack capacity
- Step 3: Return either the previous solution, or just the highest bidder if he achieves higher social welfare on his own

Knapsack auctions

- Why do we need the last step?
- Maybe there is a bidder with a very high value, but with a large demand as well
- The algorithm may not select this bidder in the first steps
- Step 3 ensures we do not miss out such highly-valued bidders

- Claim: This algorithm is monotone

- Theorem: Using Myerson's lemma, we can have a truthful polynomial time mechanism, that produces at least 50% of the optimal social welfare

Knapsack auctions

Going further

- Knapsack also admits an FPTAS (Fully Polynomial Time Approximation Scheme)
  - We can have a (1- ε)-approximation for any constant ε >0 [Ibarra, Kim '75]
  - But this is not a monotone algorithm
- [Briest, Krysta, Voecking '05]: A truthful FPTAS for Knapsack

- **Conclusion**: For a knapsack auction and any ε >0, we have a truthful mechanism that produces at least (1 – ε)-fraction of the optimal social welfare and runs in time polynomial in n and 1/ ε

General Approach

Suppose we have a single-parameter auction where the social welfare maximization problem is NP-hard

- ➢Check if any of the known approximation algorithms for the problem is monotone (usually not)
- ➢If not, then try to tweak it so as to make it monotone (sometimes feasible)
- ➢Or design a new approximation algorithm that is monotone (hopefully without worsening the approximation guarantee)

Single-minded bidders

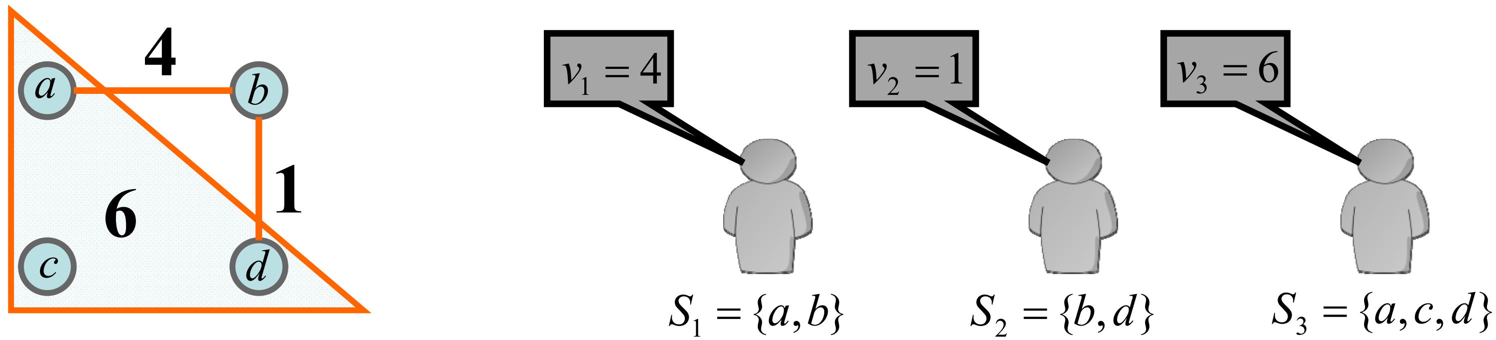
A single-paramerer auction with non-identical items

- The auctioneer has a set M of items for sale
- Each bidder i is interested in acquiring a specific subset of items, $S_i \subseteq M$ (known to the mechanism)
  - If the bidder does not obtain S_i (or a superset of it), his value is 0
- Each bidder submits a bid b_i for his value if he obtains the set

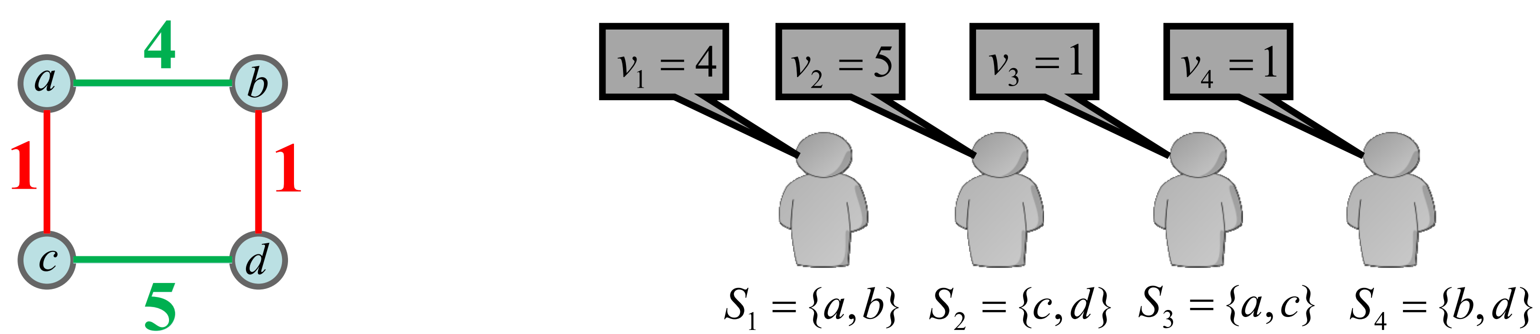
- Motivated by certain spectrum auctions
- Feasible allocations: the auctioneer needs to select winners who do not have overlapping sets

Single-minded bidders

Examples



- In the example above, the auctioneer can accept only 1 bidder as a winner
- In the example below, the auctioneer can accept up to 2 bidders as winners



Single-minded bidders

Social welfare maximization:

- Given the bids of the players, select a set of bidders with non-overlapping subsets, so as to maximize the sum of their bids
- It contains the SET PACKING problem, hence NP-hard
- Actually it gets even worse w.r.t. approximation

Theorem [Sandholm ’99]: Under certain complexity theory assumptions, we cannot have an algorithm with approximation factor better than 1/sqrt(m)

Q: Can we have a 1/sqrt(m)-approximation?

Single-minded bidders

$s_i = |S_i|$

[Lehmann, O' Callaghan, Shoham '01]:

- Order the bidders in decreasing order of b_i/sqrt(s_i)
- Accept each bidder in this order unless overlapping with previously accepted bidders
- Payment i: largest bid b_j for set S_j with nonempty intersection with S_i .

- This algorithm achieves
  - 1/sqrt(m)-approximation, where m = |M|
  - 1/d-approximation, where d = $\max_i s_i$
  - Monotonicity and truthfulness.

Final conclusion: truthful polynomial time mechanism with the best possible approximation to the social welfare

Single-minded bidders

$s_i = |S_i|$

- Order the bidders in decreasing order of $b_i/\text{sqrt}(s_i)$
- Accept each bidder in this order unless overlapping with previously accepted bidders

- *A* algorithm's solution (set of indices accepted by Greedy)
- O optimal solution (set of indices accepted by OPT)

Wlog. assume that $O \cap A = \emptyset$.

Partition O into O_i, $i \in A$, s.t. $j \in O_i$ if $j \in O$ and $S_i \cap S_j \neq \emptyset$.

$$\sum_{j \in O_i} v_j \leq \frac{v_i}{\sqrt{s_i}} \sum_{j \in O_i} \sqrt{s_j} \qquad \text{Greedy property}$$

$$\leq \frac{v_i}{\sqrt{s_i}} \sqrt{\sum_{j \in O_i} s_j \sqrt{|O_i|}} \qquad \text{Cauchy-Schwarz ineq.}$$

$$\leq \frac{v_i}{\sqrt{s_i}} \sqrt{m} \sqrt{s_i} \qquad |O_i| \leq s_i \text{ and } \sum_{j \in O_i} s_j \leq m$$

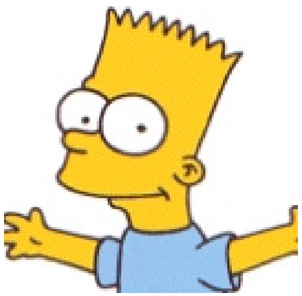
$$\leq v_i \sqrt{m}$$

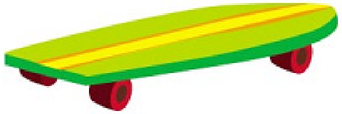
Multi-dimensional Bidders / Combinatorial Auctions

The model

Set of players
N = {1, 2, …, n}

Set of indivisible goods
M = {1, 2, …, m}





Combinatorial Auctions

- Any auction with multiple items for sale
- The players may be allowed to express interest / bids on various combinations of goods
- In practice very active field within the last 10-15 years
  - Spectrum licences
  - The FCC incentive auction:
    - https://www.fcc.gov/about-fcc/fcc-initiatives/incentive-auctions
  - Transportation routes
  - Logistics

Combinatorial auctions

- In practice, it seems economically more efficient and profitable to sell the items **together** than have a separate auction for each good
- Main challenges:
  - Algorithmic: How shall we design the allocation rule (especially if we have many overlaps in what the players want the most)?
  - Game-theoretic: Can we generalize Myerson's lemma to get truthful mechanisms?

Valuation functions

- So far we studied settings where a single parameter v_i determined all the information we needed for a player
- Most general scenario: consider that each player has a valuation function defined for every subset of the items
- $v_i : P(M) \rightarrow R$
  - where P(M) = powerset of M (all subsets of M)
  - For every $S \subseteq M$,
    - $v_i(S)$ = utility derived for player i if he acquires set S

      = maximum amount willing to pay for acquiring S
- We always assume **monotonicity** ("free-disposal"): for all $T \subseteq S$, $v_i(T) \leq v_i(S)$.

Examples of valuation functions

Additive valuation functions

- For every $S \subseteq M$, $v_i(S) = \sum_{j \in S} v_{ij}$
  - where v_{ij} = utility of acquiring item j
- Hence, the function can be completely determined by specifying the vector $(v_{i1}, v_{i2}, ..., v_{im})$
- m parameters for each bidder
- In such cases, the goods can be auctioned independently:
  - The value of an item is not affected by other items that a bidder may have already obtained

Examples of valuation functions

- In practice, the items may be interrelated with each other and additive valuations are not appropriate
- The value they add to a player may depend on the other items that the player has
- The items may exhibit
  - **Complementarity**: some items may be valuable only when they are sold together with other items (e.g. left and right shoe)
  - **Substitutability:** some items may be of similar type and should not be sold together to the same player (e.g. 2 cars with the same features)

Examples of valuation functions

Subadditive functions

- For any 2 disjoint subsets $S \subseteq M$, $T \subseteq M$,

$$v_i(S \cup T) \leq v_i(S) + v_i(T)$$

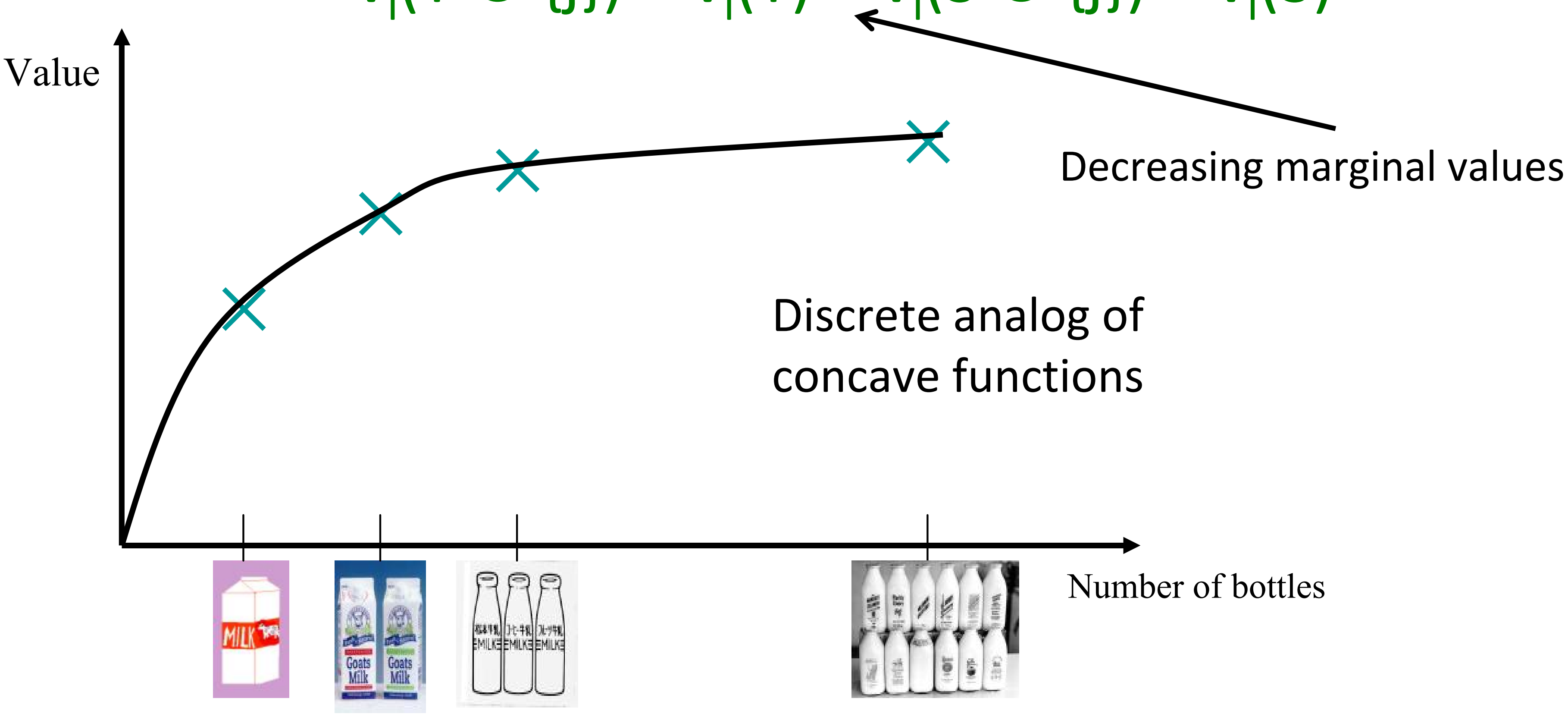
- In this case, we have substitutability among the goods
- They are also called complement-free functions (since we do not have complementarity)

Examples of valuation functions

Submodular functions

For any 2 subsets S, T, with $S \subseteq T \subseteq M$, and for every $j \notin T$

$$v_i(T \cup \{j\}) - v_i(T) \le v_i(S \cup \{j\}) - v_i(S)$$



Examples of valuation functions

- Submodular functions form a special class of subadditive valuations
- Hence, they also do not exhibit complementarity
- They play a key role in micro-economic theory
- Expressing the fact that utility gets “saturated” as we keep allocating substitutes to the same player

Examples of valuation functions

Symmetric submodular

- Special case of submodular functions, where all goods are identical
  - Hence, the final utility depends only on how many items the player receives
- Applicable for multi-unit auctions
  - E.g., auctions for government bonds fall under this framework
- For k identical items, such functions can be represented by a vector of k marginal values
  - $(m_i(1), m_i(2),\ldots, m_i(k))$ with $m_i(j) \geq m_i(j+1)$
  - Where $m_i(j)$ = additional utility to the player for obtaining the j-th unit, if the player already has j-1 units

Examples of valuation functions

Superadditive functions

- For any 2 disjoint subsets $S \subseteq M$, $T \subseteq M$,

$$v_i(S \cup T) \geq v_i(S) + v_i(T)$$

- In this case, we have complementarity
- For example, the items may not have any value if they are sold on their own, but only when sold in bundles with other goods
    - Single-minded bidders fall under this class

Relations between different classes of valuation functions