

Algorithmic Game Theory Notes

Jingyuan Hu

Disclaimer: This note is entirely for personal purpose. Please feel free to email me at juanh96@gmail.com if you find any mistakes or you have any suggestions. The majority of the contents/materials are based on Lecture Notes from Stanford CS364A (Algorithmic Game Theory), UBC CPSC536F (Algorithmic Game Theory), which is meant to be accessible to a Masters or 1st-year PhD student with an affinity for algorithms and NP-completeness. Notice that this is no substitute for a traditional game theory or microeconomics economics course where *A Course in Game Theory* (Osborne, Rubinstein) and *Microeconomic Theory* (MWG) are usually used as textbooks.

Reference textbooks:

Twenty Lectures on Algorithmic Game Theory (Tim Roughgarden)

Algorithmic Game Theory (Tim Roughgarden, Eva Tardos)

Contents

1	Introduction	1
2	Mechanism Design Basics	2
2.1	Single-item Auctions	2
2.2	Sealed-bid Auctions	2
2.3	First-price Auctions	2
2.4	Second-price Auctions and Dominant Strategies	2
2.5	Ideal Auctions	3
2.6	Case Study: Sponsored Search Auctions	3
3	Myerson's Lemma	4
3.1	Single-Parameter Environments	4
3.2	Allocation and Payment Rules	4
3.3	Statement of Myerson's Lemma	5
3.4	Applying the Payment Formula	6
4	Algorithmic Mechanism Design	6
5	Revenue-Maximizing Auctions	6
6	Simple Near-Optimal Auctions	6
7	Multi-parameter Mechanism Design	6

1 Introduction

- *Braess's Paradox*: Adding a superfast link to a network can negatively impact all of the traffic.
- *Price of anarchy (POA)* is the ratio between the system performance with strategic players and the best possible system performance. When the POA is close to 1, selfish behaviour is largely benign.

- A *game* is specified by a set of players, a strategy set for each player, and a payoff to each player in each outcome.

Theorem 1.1. (*Nash's Theorem*) Every finite two-player game has a Nash Equilibrium

2 Mechanism Design Basics

2.1 Single-item Auctions

Each bidder i has a nonnegative *valuation* v_i (maximum willingness-to-pay) for the item being sold. We assume this valuation is *private*, meaning it is unknown to the seller and to the other bidders. Our bidder utility model (*Quasi-linear utility model*): if bidder i loses an auction, her utility is 0; if bidder wins at a price p , her utility is $v_i - p$.

2.2 Sealed-bid Auctions

Sealed-bid auctions:

1. Each bidder i privately communicates a bid b_i to the seller in a sealed envelope, if you like.
2. The seller decides who gets the item (if anyone).
3. The seller decides on a selling price.

2.3 First-price Auctions

First-price auctions: the winning bidder pays her bid.

2.4 Second-price Auctions and Dominant Strategies

- *Second-price (Vickrey) auctions:* A sealed-bid auction in which the highest bidder wins and pays a price equal to the second-highest bid.
- *Dominant strategy* is guaranteed to maximize a bidder's utility, no matter what the other bidders do.

Proposition 2.1. (Incentives in Second-Price Auctions) In a second-price auction, every bidder i has a dominant strategy: set the bid b_i equal to her private valuation.

Proof. Fix an arbitrary bidder i with valuation v_i , and the bids \mathbf{b}_{-i} of other bidders. We need to show that the bidder i 's utility is maximized by setting $b_i = v_i$. Let $B = \max_{j \neq i} b_j$ denote the highest bid by other bidders. If $b_i < B$, then i loses and receives utility 0. If $b_i \geq B$, then i wins at price B and receives utility $v_i - B$. Therefore, if $v_i < B$, the maximum utility that bidder i can obtain is $\max\{0, v_i - B\} = 0$ and is achieved by bidding truthfully (and losing); if $v_i \geq B$, the maximum utility that bidder i can obtain is $\max\{0, v_i - B\} = v_i - B$ and is also achieved by bidding truthfully (and winning). \square

Proposition 2.2. In a second-price auction, every truthful bidder is guaranteed nonnegative utility.

Proof. Losers receives utility 0. For winner, say bidder i , his utility will be $v_i - p$ where p is the second-highest bid. Since i is the winner (and bid his true valuation), we have $p \leq v_i \Rightarrow v_i - p \geq 0$. \square

Note: (*Exercise*) Truthful bidding is the unique dominant strategy for a bidder in a second-price auction

2.5 Ideal Auctions

We first start with a couple of definitions:

Definition 2.1. (Dominant-Strategy Incentive Compatible) An auction is *dominant-strategy incentive compatible (DSIC)* if truthful bidding is always a dominant strategy for every bidder and if truthful bidders always obtain nonnegative utility.

Note: The condition that truthful bidders obtain nonnegative utility is traditionally considered a separate requirement, called *individual rationality* or *voluntary participation*. (Bidder with an obvious dominant strategy will play it)

Definition 2.2. (Social welfare) The social welfare of an outcome of a single-item auction is $\sum_{i=1}^n v_i x_i$, where $x_i = 1_{\{i \text{ wins}\}}$. (Since there's only one item, we have feasibility constraint: $\sum_{i=1}^n x_i \leq 1$, in this case the social welfare is just the valuation of the winner, or 0 if there is no winner.)

Note: Sale price does not appear in the definition of social welfare, since we can think of the seller as an agent whose utility is the revenue she earns, then it will cancel out the utility lost by the auction winner from paying for the item. (Suppose the price is 10, sell at 8: seller agent earn -2 , buyer earn 2, welfare is the winner (buyer); If sell at 12, seller agent earn 2, buyer earn -2 , welfare is the winner (seller))

Second-price single item auctions are 'ideal' in that they enjoy three quite different and desirable properties:

Theorem 2.1. (Second-Price Auction Are Ideal) A second-price single item auction satisfies the following:

- (Strong incentive guarantees) It is a DSIC auction
- (Strong performance guarantees) It is welfare maximizing
- (Computational efficiency) It can be implemented in time polynomial (indeed, linear) in the size of the input, meaning the number of bits necessary to represent the numbers v_1, \dots, v_n .

Note: DSIC property makes bidder and seller decision easier; Welfare maximization property let the auction identifies the bidder with the highest valuation even though the valuations are a priori unknown to the seller; Computational efficiency assures practical utility (e.g. auctions for online advertising)

2.6 Case Study: Sponsored Search Auctions

- The items for sale are k "slots" for sponsored links on a search results page
- Quantify different slots using *click through rates (CTRs)*: the CTR α_j of a slot j represents the probability that the end users clicks on the slot. (Reasonable assumption: $\alpha_1 \geq \dots \geq \alpha_k$; Another strong assumption: CTR of a slot is independent of its occupant, in general they are related based on some 'quality score' of the advertiser)
- Advertiser has a private valuation v_i for each click on her link, hence the expected value by advertiser i from slot j is $v_i \alpha_j$
- We want (1)DSIC, (2)Social welfare maximization, (3)Computational efficiency.
- Need to design: choice of who wins & choice of who pays what
 - Step 1: Assume, without justification, bidders bid truthfully. How should we *assign bidders to slots* so that the above properties (2) & (3) hold
 - Step 2: Given our answer in step 1, how should we set *selling price* s.t. the property (1) holds

3 Myerson's Lemma

3.1 Single-Parameter Environments

- n agents, each agent i has a private nonnegative valuation v_i (her value 'per unit of stuff' that she acquires)
- There is a feasible set X , each element of X is a nonnegative n -vector (x_1, \dots, x_n) where x_i denotes the 'amount of stuff' given to agent i

Now we present some examples

Example 3.1. (Single-Item Auction) In a single-item auction, X is the set of $0-1$ vectors that have at most one 1, that is, $\sum_{i=1}^n x_i \leq 1$

Example 3.2. (k-Unit Auction) With k identical items and the constraint that each bidder gets at most one, the feature set is the set of $0-1$ vectors that satisfies $\sum_{i=1}^n x_i \leq k$.

Example 3.3. (Sponsored Search Auction) X is the set of n -vectors corresponding to assignments of bidders to slots, where each slot is assigned to at most one bidder and each bidder is assigned to at most one slot. If bidder i is assigned to slot j , then the component x_i equals the click-through rate α_j of her slot.

Example 3.4. (Public Project) Deciding whether or not to build a public project that can be used by all, can be modeled by set $X = \{(0, \dots, 0), (1, \dots, 1)\}$

3.2 Allocation and Payment Rules

In sealed-bid auction we need to design: choice of who wins (**allocation rule**) & choice of who pays what (**payment rule**), this leads to the three steps:

1. Collect bids $\mathbf{b} = (b_1, \dots, b_n)$ from all agents, the vector is called the *bid vector* or *bid profile*.
2. (**Allocation Rule**) Choose a feasible allocation $x(\mathbf{b}) \in X \subseteq \mathbb{R}^n$ as a function of their bids.
3. (**Payment Rule**) Choose payments $p(\mathbf{b}) \in \mathbb{R}^n$ as a function of the bids.

Note: Procedure of this type are called *direct-revelation mechanisms*, since in the first step agents are asked to reveal directly their private valuations.

Then, with this quasilinear utility model, in a mechanism with allocation and payment rules \mathbf{b}, \mathbf{p} respectively, agent i receives utility $u_i(\mathbf{b}) = v_i x_i(\mathbf{b}) - p_i(\mathbf{b})$ given the bid profile is \mathbf{b} .

We now only focus on payment rules that satisfy $p_i(\mathbf{b}) \in [0, b_i x_i(\mathbf{b})]$ for every agent i and bid profile \mathbf{b} :

- $p_i(\mathbf{b}) \geq 0$: prohibiting the seller from paying the agents. (p_i is the amount that the agent i need to pay)
- $p_i(\mathbf{b}) \leq b_i x_i(\mathbf{b})$: ensures a truthful agent receives nonnegative utility. (the amount that the agent need to pay under the mechanism is at most *how much he bid per unit* \times *how much unit he got allocated*, if he pays more than he bid (valuation, since he is truthful), he will have negative utility)

3.3 Statement of Myerson's Lemma

Definition 3.1. (Implementable Allocation Rule) An allocation rule \mathbf{x} for a single-parameter environment is *implementable* if there is a payment rule \mathbf{p} s.t. the direct-revelation mechanism (\mathbf{x}, \mathbf{p}) is DSIC.

Note: In other words, the projection of DSIC mechanisms onto their allocation rules is the set of implementable rules.

Definition 3.2. (Monotone Allocation Rule) An allocation rule \mathbf{x} for a single-parameter environment is *monotone* if for every agent i and bids \mathbf{b}_{-i} by the other agents, the allocation rule $x_i(z, \mathbf{b}_{-i})$ to i is nondecreasing in her bid z .

Note: The single-item auction allocation rule that awards the item to the highest bidder is monotone: if you're the winner and keep raising your bid (other bids fixed), you continue to win; Awarding to the second-highest is not monotone: if you're the winner (second highest) and keep raising, eventually you will become the highest and lose. (**Notice:** $x_i(z, \mathbf{b}_{-i})$ is the "amount of stuff" not the actual value, otherwise awarding to the highest will clearly not monotone.)

Theorem 3.1. (Myerson's Lemma) Fix a single-parameter environment, then

- (a) An allocation rule \mathbf{x} is implementable iff it is monotone.
- (b) If \mathbf{x} is monotone, then there exist an unique payment rule for which the direct-revelation mechanism (\mathbf{x}, \mathbf{p}) is DSIC and $p_i(\mathbf{b}) = 0$ whenever $b_i = 0$
- (c) The payment rule in (b) is given by an explicit formula

Proof. Fix a single-parameter environment and consider an allocation rule \mathbf{x} , fix i and \mathbf{b}_{-i} arbitrarily. Write $x(z)$ and $p(z)$ for the allocation $x_i(z, \mathbf{b}_{-i})$ and payment $p_i(z, \mathbf{b}_{-i})$ of agent i when she bids z . We invoke the DSIC constraint via a simple trick: suppose (\mathbf{x}, \mathbf{p}) is DSIC, and consider any arbitrary $0 \leq y < z$. Because agent i might well have private valuation z and is free to submit the false bid y , DSIC demands that utility of bidding z is greater than utility of bidding y .

$$\underbrace{z \cdot x(z)}_{\text{bid(valuation)} \times \text{allocation}} - \underbrace{p(z)}_{\text{need to pay}} \geq z \cdot x(y) - p(y)$$

Similarly, agent i might well have private valuation y and could submit the false bid z , (\mathbf{x}, \mathbf{p}) must satisfy

$$y \cdot x(y) - p(y) \geq y \cdot x(z) - p(z)$$

hence we have $z[x(y) - x(z)] \leq p(y) - p(z) \leq y[x(y) - x(z)]$, which implies every implementable allocation rule is monotone. Thus we can assume for the rest of the proof that \mathbf{x} is monotone.

Next, consider the case where x is a piecewise constant function, where the graph of x is flat except for a finite number of "jumps". Now taking $\lim_{y \downarrow z}$ (**Isn't $y < z$??**) on $z[x(y) - x(z)] \leq p(y) - p(z) \leq y[x(y) - x(z)]$, we notice that LHS and RHS will be 0 if there's no jumps in x at z . If there is a jump of h at z , then LHS and RHS both tend to $z \cdot h$, which means the following constraint on p : for every z we have

$$\text{jump in } p \text{ at } z = z \cdot [\text{jump in } x \text{ at } z]$$

Combine this with the initial condition $p(0) = 0$, we have the payment formula: for every agent i , bids \mathbf{b}_{-i} by other agents, and bid b_i by i :

$$p_i(b_i, \mathbf{b}_{-i}) = \sum_{j=1}^{\ell} z_j [\text{jump in } x_i(\cdot, \mathbf{b}_{-i}) \text{ at } z_j]$$

where z_1, \dots, z_ℓ are the breakpoints of the allocation function $x_i(\cdot, \mathbf{b}_{-i})$ in the range $[0, b_i]$.

Similar argument applies even when x is monotone but not piecewise constant, for example, x is differentiable. dividing the sandwich inequality by $y - z$, take limit as $y \downarrow z$ yields the constraint $p'(z) = z \cdot x'(z)$, recall $p(0) = 0$, then

$$p_i(b_i, \mathbf{b}_{-i}) = \int_0^{b_i} z \cdot \frac{d}{dz} x_i(z, \mathbf{b}_{-i}) dz$$

Thus, for every allocation rule \mathbf{x} , there is at most one payment rule \mathbf{p} s.t. (\mathbf{x}, \mathbf{p}) is DSIC. But we still to check that this payment rule always works provided \mathbf{x} is monotone. (We already know that even this payment rule cannot extend a non-monotone allocation rule to a DSIC mechanism) We proof this by picturing ... \square

Myerson's Lemma is the foundation of most mechanism design theory:

- (a) states that Implementable Allocation Rule and Monotone Allocation Rule defines exactly the same class of allocation rules.
- (b) states that when an allocation rule is implementable, there is no ambiguity in how to assign payments to achieve the DSIC property. (there's only one way to do it)
- (c) states that there is a relatively simple and explicit formula for the payment rule in (b).

3.4 Applying the Payment Formula

4 Algorithmic Mechanism Design

5 Revenue-Maximizing Auctions

6 Simple Near-Optimal Auctions

7 Multi-parameter Mechanism Design