Economics and Computation

Yingkai Li

EC4501/EC4501HM Semester 2, AY2024/25

Logistics

Instructor: Yingkai Li

Office: AS2 05-21

Office hour: by appointment.

No course for the Chinese New Year.

Schedule a make-up class for Feb. 6th.

Course Philosophy

Economic analysis using algorithmic tools.

- approximation analysis: design and analysis of simple mechanisms in complex environments where finding the optimal is infeasible or undesirable.
- robust analysis: design robust mechanisms in the absence of detailed knowledge about the environment.
- data analysis: how to design good mechanisms with access to historical data.

Course Philosophy

Economic analysis using algorithmic tools.

- approximation analysis: design and analysis of simple mechanisms in complex environments where finding the optimal is infeasible or undesirable.
- robust analysis: design robust mechanisms in the absence of detailed knowledge about the environment.
- data analysis: how to design good mechanisms with access to historical data.

Goal: understand the design of good mechanisms in practical applications.

- online platforms (Google/Meta);
- resource allocations (FCC Spectrum/Land Resource/Cloud Computing);
- blockchains and cryptocurrencies (Bitcoin);
- recommendation system (Yelp/Netflix);
- etc.

Reading Lists

1. Jason Hartline. *Mechanism Design and Approximation.* https://jasonhartline.com/MDnA/
2. Tim Roughgarden. *Twenty Lectures on Algorithmic Game Theory.* https://timroughgarden.org/notes.html
3. Aleksandrs Slivkins. *Introduction to Multi-Armed Bandits.* https://arxiv.org/abs/1904.07272

Additional readings:

- Noam Nisan, Tim Roughgarden, Éva Tardos, Vijay V. Vazirani. *Algorithmic Game Theory.* Cambridge University Press.
- Federico Echenique, Nicole Immorlica, Vijay V. Vazirani. *Online and Matching-Based Market Design.* Cambridge University Press.

Prerequisite

Required: Basics in probabilities, calculus, and how to prove formal theorems.

Not required: solid background knowledge about algorithm design (CS), mechanism design (Econ), or game theory (Econ). Coding is also not required.

Evaluations

- Two assignments (40%); due on March 10th, April 11th.
- Course project (30%); due on April 11th, mid-term review on March 14th.
- Final exam (30%); scheduled on May 6th, 5pm.
- Survey paper (25%); due on April 4th; only for HM students.

Syllabus

Week 1:	Preview of the course
Week 2/3/4:	Auctions: welfare and revenue maximization
Week 5/6:	Prior-independent and prior-free analysis
Week 7/8/9:	Learning agents and mechanism design under learning
Week 10:	Contracts and moral hazard
Week 11/12:	Topic courses: fairness, privacy, etc. Details depend on interests.
Week 13:	Project presentation by students

Basics on Game Theory

Incomplete Information Games

A static game with incomplete information is denoted as $\Gamma_I = \left(N, (A_i)_{i\in N}, (u_i)_{i\in N}, (\Theta_i)_{i\in N}, \mu\right)$ where

- N is the set of players;
- A_i is the set of player i's actions; (what the agents can do)
- Θ_i is the set of player i's "types" where $\theta_i \in \Theta_i$ is private information of i; (what the agents know)
- $u_i : A \times \Theta \to \mathbb{R}$ is player i's payoff function (where $A = \times_{i\in N} A_i$, and $\Theta = \times_{i\in N}\Theta_i$).
- $\mu(\theta)$ is the probability that a type profile $\theta \in \Theta$ occurs.

Incomplete Information Games

A static game with incomplete information is denoted as $\Gamma_I = \left(N, (A_i)_{i \in N}, (u_i)_{i \in N}, (\Theta_i)_{i \in N}, \mu\right)$ where

- N is the set of players;
- A_i is the set of player i's actions; (what the agents can do)
- Θ_i is the set of player i's "types" where $\theta_i \in \Theta_i$ is private information of i; (what the agents know)
- $u_i : A \times \Theta \to \mathbb{R}$ is player i's payoff function (where $A = \times_{i \in N} A_i$, and $\Theta = \times_{i \in N} \Theta_i$).
- $\mu(\theta)$ is the probability that a type profile $\theta \in \Theta$ occurs.

μ is called a common prior.

- Let μ_i denote the marginal distribution of μ on Θ_i, i.e., $\mu_i(\theta_i) \equiv \sum_{\theta_{-i} \in \Theta_{-i}} \mu(\theta_i, \theta_{-i})$.
- Let $\mu(\theta_{-i} | \theta_i)$ be the belief of agent i over θ_{-i} conditional on his type being θ_i.

Strategies and Bayesian Nash Equilibrium

A strategy of player i in Γ_I is a mapping $s_i : \Theta_i \to \Delta(A_i)$.

- s_i is a pure strategy if the mapping is deterministic, i.e., $s_i : \Theta_i \to A_i$. Let S_i be the set of pure strategies for i.

Strategies and Bayesian Nash Equilibrium

A strategy of player i in Γ_I is a mapping $s_i : \Theta_i \to \Delta(A_i)$.

- s_i is a pure strategy if the mapping is deterministic, i.e., $s_i : \Theta_i \to A_i$. Let S_i be the set of pure strategies for i.

Definition (BNE)

A strategy profile s is a Bayesian Nash Equilibrium if for any agent i and any type θ_i (such that $\mu_i(\theta_i) > 0$), for any action a_i^* in the support of $s_i(\theta_i)$, we have

$$a_i^* \in \operatorname*{argmax}_{a_i \in A_i} \sum_{\theta_{-i} \in \Theta_{-i}} \mu(\theta_{-i}|\theta_i) \cdot \mathbf{E}_{a_{-i} \sim s_{-i}(\theta_{-i})}[u_i(a_i, a_{-i}, \theta)] .$$

Informal definition of BNE: all agents are doing the best they can given what they think others are doing.

Historical Review: Selfish Routing

Selfish Routing

Suppose now the government has the funding to build another road in the current system. Should the government do it?

Selfish Routing

Suppose now the government has the funding to build another road in the current system. Should the government do it? Sometime no, not even when it's costless to do so!

Selfish Routing

Suppose now the government has the funding to build another road in the current system. Should the government do it? Sometime no, not even when it's costless to do so!

Braess's paradox [Pigou '20; Braess '68]

- adding more roads could lead to more severe congestion.

Selfish Routing

Suppose now the government has the funding to build another road in the current system. Should the government do it? Sometime no, not even when it's costless to do so!

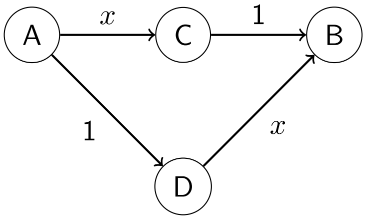
Braess's paradox [Pigou '20; Braess '68]

- adding more roads could lead to more severe congestion.

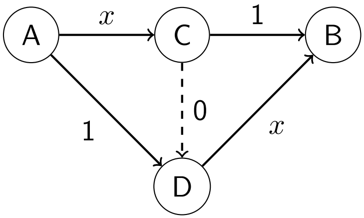
Example: agents travel from A to B.

- A $\to$ C, D $\to$ B: travel time x, fraction of travelers.
- A $\to$ D, C $\to$ B: travel time 1.
- New road in network: open a portal from C to D with zero travel time.

Network Before Adding Shortcut

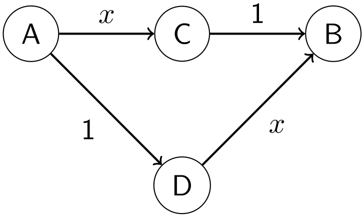


Network After Adding Shortcut

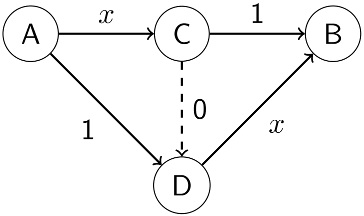


Braess's Paradox

Network Before Adding Shortcut

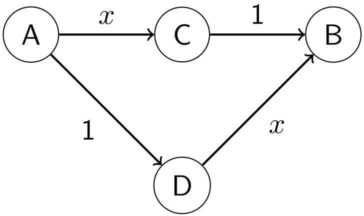


Network After Adding Shortcut

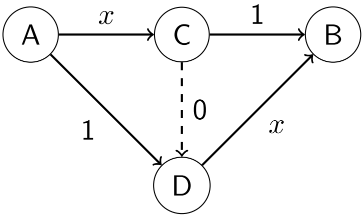


Braess's Paradox

Network Before Adding Shortcut



Network After Adding Shortcut

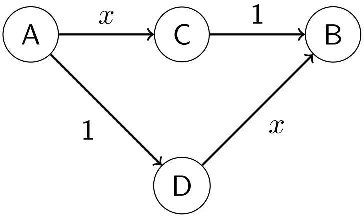


Equilibrium before shortcut: $\frac{1}{2}$ chooses A $\rightarrow$ C $\rightarrow$ B, $\frac{1}{2}$ chooses A $\rightarrow$ D $\rightarrow$ B.

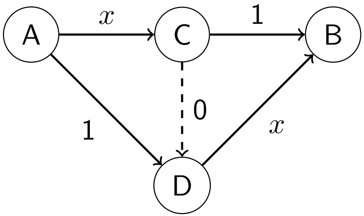
- total travel time is $\frac{3}{2}$ for all agents.

Braess's Paradox

Network Before Adding Shortcut



Network After Adding Shortcut



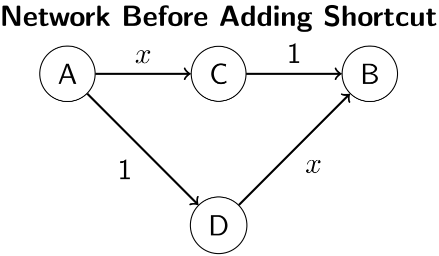
Equilibrium before shortcut: $\frac{1}{2}$ chooses A $\rightarrow$ C $\rightarrow$ B, $\frac{1}{2}$ chooses A $\rightarrow$ D $\rightarrow$ B.

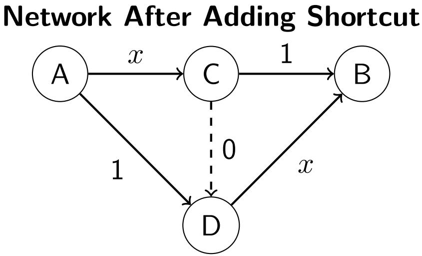
- total travel time is $\frac{3}{2}$ for all agents.

Equilibrium after shortcut: all agents choose A $\rightarrow$ C $\rightarrow$ D $\rightarrow$ B.

- total travel time is 2 for all agents.

Braess's Paradox





Equilibrium before shortcut: $\frac{1}{2}$ chooses A $\to$ C $\to$ B, $\frac{1}{2}$ chooses A $\to$ D $\to$ B.

- total travel time is $\frac{3}{2}$ for all agents.

Equilibrium after shortcut: all agents choose A $\to$ C $\to$ D $\to$ B.

- total travel time is 2 for all agents.

$2 > \frac{3}{2}$: everyone suffers from having an additional shortcut!

Selfish Routing

In the previous example, the efficiency loss due to strategic behavior is $2/\frac{3}{2} = \frac{4}{3}$.

- not so bad compared to the first-best scenario.

Selfish Routing

In the previous example, the efficiency loss due to strategic behavior is $2/\frac{3}{2} = \frac{4}{3}$.

- not so bad compared to the first-best scenario.

Question: can we quantify the worst-case efficiency loss due to strategic behavior (Price of Anarchy (PoA))?

- if PoA is small in selfish routing, building more roads is always approximately optimal since it always improves the first best.

Selfish Routing

In the previous example, the efficiency loss due to strategic behavior is $2/\frac{3}{2} = \frac{4}{3}$.

- not so bad compared to the first-best scenario.

Question: can we quantify the worst-case efficiency loss due to strategic behavior (Price of Anarchy (PoA))?

- if PoA is small in selfish routing, building more roads is always approximately optimal since it always improves the first best.

Table: The worst-case POA with cost functions that are polynomials with nonnegative coefficients and degree at most d. See https://theory.stanford.edu/~tim/f13/l/l11.pdf

Description	**Typical Representative**	**Price of Anarchy**
Linear	$ax + b$	$\frac{4}{3}$
Quadratic	$ax^2 + bx + c$	$\frac{\sqrt[3]{3}\sqrt[3]{3}-2}{3\sqrt[3]{3}-2} \approx 1.6$
Cubic	$ax^3 + bx^2 + cx + d$	$\frac{\sqrt[4]{4}\sqrt[4]{4}-3}{4\sqrt[4]{4}-3} \approx 1.9$
Polynomials of degree $\leq d$	$\sum_{i=0}^{d} a_i x^i$	$\frac{(d+1)\sqrt[d+1]{d+1}}{(d+1)\sqrt[d+1]{d+1}-d} \approx \frac{d}{\ln d}$

Auctions and Welfare Analysis

Auctions

Auctions: a single item, n agents.

- each agent i has value $v_i \sim F_i$;

Auctions

Auctions: a single item, n agents.

- each agent i has value $v_i \sim F_i$;

How to allocate the item efficiently?

- ideally we want to give the item to the agent with highest v_i.

Auctions

Auctions: a single item, n agents.

- each agent i has value $v_i \sim F_i$;

How to allocate the item efficiently?

- ideally we want to give the item to the agent with highest v_i.

Use transfers to discipline the agent:

- each agent i has utility $u_i = v_i x_i - p_i$.

Second-price Auction

Second-price Auction: Each bidder i place a bid $b_i \geq 0$ in the auction.

- highest bidder wins where ties are broken uniform randomly;
- winner pays the second highest bid.

Second-price Auction

Second-price Auction: Each bidder i place a bid $b_i \geq 0$ in the auction.

- highest bidder wins where ties are broken uniform randomly;
- winner pays the second highest bid.

Truthful bidding as equilibrium: all bidders maximize their utilities by bidding $b_i(v_i) = v_i$.

Second-price Auction

Second-price Auction: Each bidder i place a bid $b_i \geq 0$ in the auction.

- highest bidder wins where ties are broken uniform randomly;
- winner pays the second highest bid.

Truthful bidding as equilibrium: all bidders maximize their utilities by bidding $b_i(v_i) = v_i$.

- if $\max_{j \neq i} b_j \geq v_i$: bidder i does not gain by bidding higher to win;

Second-price Auction

Second-price Auction: Each bidder i place a bid $b_i \geq 0$ in the auction.

- highest bidder wins where ties are broken uniform randomly;
- winner pays the second highest bid.

Truthful bidding as equilibrium: all bidders maximize their utilities by bidding $b_i(v_i) = v_i$.

- if $\max_{j \neq i} b_j \geq v_i$: bidder i does not gain by bidding higher to win;
- if $\max_{j \neq i} b_j < v_i$: bidder i does not gain by bidding lower since the payment won't decrease conditional on winning, and losing is worse.

Second-price Auction

Second-price Auction: Each bidder i place a bid $b_i \geq 0$ in the auction.

- highest bidder wins where ties are broken uniform randomly;
- winner pays the second highest bid.

Truthful bidding as equilibrium: all bidders maximize their utilities by bidding $b_i(v_i) = v_i$.

- if $\max_{j \neq i} b_j \geq v_i$: bidder i does not gain by bidding higher to win;
- if $\max_{j \neq i} b_j < v_i$: bidder i does not gain by bidding lower since the payment won't decrease conditional on winning, and losing is worse.

Efficiency in equilibrium: in second-price auction, highest value agent always wins the item in the truthful bidding equilibrium.

Second-price Auction

Even with strong efficiency guarantees, second-price auction is still not adopted in many practical applications.

- second-price auction is not credible: the seller may attempt to get more revenue by misreporting the second highest bid.
- equilibrium selection.

Second-price Auction

Even with strong efficiency guarantees, second-price auction is still not adopted in many practical applications.

- second-price auction is not credible: the seller may attempt to get more revenue by misreporting the second highest bid.
- equilibrium selection.

Understand the efficiency guarantee of simple and practical mechanisms.

- Posted pricing mechanisms: offer price p_i to agent i. The item is sold to the first agent who is willing to purchase.
- First-price auction: each bidder i places a bid $b_i \geq 0$ in the auction. Highest bid wins and the winner pays his bid.

Worst-case Approximations

Given a distribution F over values, denote the optimal welfare as

$$\mathrm{Wel}(F) = \mathbf{E}_{v \sim F}\left[\max_i v_i\right].$$

Worst-case Approximations

Given a distribution F over values, denote the optimal welfare as

$$\mathrm{Wel}(F) = \mathbf{E}_{v \sim F}\left[\max_i v_i\right].$$

For any mechanism M, let $M(F)$ be the social welfare achieved in mechanism M given distribution F. The worst-case approximation of mechanism M is

$$\mathrm{APX}(M) \triangleq \max_F \frac{\mathrm{Wel}(F)}{M(F)}.$$

Worst-case Approximations

Given a distribution F over values, denote the optimal welfare as

$$\mathrm{Wel}(F) = \mathbf{E}_{v \sim F}\left[\max_i v_i\right].$$

For any mechanism M, let $M(F)$ be the social welfare achieved in mechanism M given distribution F. The worst-case approximation of mechanism M is

$$\mathrm{APX}(M) \triangleq \max_F \frac{\mathrm{Wel}(F)}{M(F)}.$$

What are the worst-case approximations for posted pricing mechanisms and first-price auction?

Job Hiring Process

Its connection to posted pricing mechanisms will be illustrated later.

Job Hiring Process

Its connection to posted pricing mechanisms will be illustrated later.

A firm want to hire for a vacant position.

- optimal policy: interview all the candidates, and selects the best one after the interviews.
- may not be feasible in certain scenarios, e.g., some candidates cannot wait long for the decisions.

Job Hiring Process

Its connection to posted pricing mechanisms will be illustrated later.

A firm want to hire for a vacant position.

- optimal policy: interview all the candidates, and selects the best one after the interviews.
- may not be feasible in certain scenarios, e.g., some candidates cannot wait long for the decisions.

Online Interview Process

- the candidates arrive in an online order;

Job Hiring Process

Its connection to posted pricing mechanisms will be illustrated later.

A firm want to hire for a vacant position.

- optimal policy: interview all the candidates, and selects the best one after the interviews.
- may not be feasible in certain scenarios, e.g., some candidates cannot wait long for the decisions.

Online Interview Process

- the candidates arrive in an online order;
- the firm observes the true quality of the current candidate, but not the quality of future candidates;
- the firm needs to make an immediate hiring decision for each candidate.

Job Hiring Process

Its connection to posted pricing mechanisms will be illustrated later.

A firm want to hire for a vacant position.

- optimal policy: interview all the candidates, and selects the best one after the interviews.
- may not be feasible in certain scenarios, e.g., some candidates cannot wait long for the decisions.

Online Interview Process

- the candidates arrive in an online order;
- the firm observes the true quality of the current candidate, but not the quality of future candidates;
- the firm needs to make an immediate hiring decision for each candidate.

Question: how to design good online hiring policies? What is the loss of adhering to online policies?

Online Selection Problems

Problem: n items arriving online.

- item i has value $v_i \sim F_i$;
- the agent knows $F_1, \ldots, F_n$ at time 0.
- at time $i \leq n$, the agent observes value v_i and decides whether to select item i (if the selection has not been made).

Note: the arrival order of the items is unknown to the agent.

Prophet Inequalities

How to evaluate the performance of an online policy?

Prophet Inequalities

How to evaluate the performance of an online policy?

Compare to a prophet who can foresee all future values.

- the prophet can guarantee an expected value of $\mathbf{E}[\max_i v_i]$.

Prophet Inequalities

How to evaluate the performance of an online policy?

Compare to a prophet who can foresee all future values.

- the prophet can guarantee an expected value of $\mathbf{E}[\max_i v_i]$.

Question: what is the performance guarantees using online policies compared to the prophet?

Prophet Inequalities

How to evaluate the performance of an online policy?

Compare to a prophet who can foresee all future values.

- the prophet can guarantee an expected value of $\mathbf{E}[\max_i v_i]$.

Question: what is the performance guarantees using online policies compared to the prophet?

Naive solution: randomly select a value (RS).

- the probability of choosing the highest value is $\frac{1}{n} \Rightarrow \mathrm{APX}(\mathrm{RS}) = n$.
- can we do better?

Prophet Inequalities

How to evaluate the performance of an online policy?

Compare to a prophet who can foresee all future values.

- the prophet can guarantee an expected value of $\mathbf{E}[\max_i v_i]$.

Question: what is the performance guarantees using online policies compared to the prophet?

Naive solution: randomly select a value (RS).

- the probability of choosing the highest value is $\frac{1}{n} \Rightarrow \mathrm{APX}(\mathrm{RS}) = n$.
- can we do better?

The designer cannot foresee the future values. How would she know whether to select the current value or not?

Threshold Policies

The designer knows the distribution of values and can predict the expected gain from the future if the current value is not selected.

- Intuitively, the designer should stop if the current value exceeds the predicted future value.

Threshold Policies

The designer knows the distribution of values and can predict the expected gain from the future if the current value is not selected.

- Intuitively, the designer should stop if the current value exceeds the predicted future value.

Simple policy in practice: threshold policies

- set threshold τ;
- at time i, selects item i if and only if $v_i \geq \tau$.

τ is an approximation of what the designer can gain in the future.

Prophet Inequalities

Theorem

There exists a threshold policy that achieves a 2*-approximation, i.e., it achieves expected value at least* $\frac{1}{2}\mathbf{E}[\max_i v_i]$.

Prophet Inequalities

Theorem

There exists a threshold policy that achieves a 2-approximation, i.e., it achieves expected value at least $\frac{1}{2}\mathbf{E}[\max_i v_i]$.

Consider threshold τ and let p_τ be the probability that an item is selected given τ.

Prophet Inequalities

Theorem

There exists a threshold policy that achieves a 2-approximation, i.e., it achieves expected value at least $\frac{1}{2}\mathbf{E}[\max_i v_i]$.

Consider threshold τ and let p_τ be the probability that an item is selected given τ.
The expected performance of the algorithm is

$$
\begin{aligned}
\mathrm{ALG}_\tau &= p_\tau \cdot \tau + \sum_{i \le n} \Pr[v_j < \tau, \forall j < i] \cdot \mathbf{E}\left[(v_i - \tau)^+\right] \\
&\ge p_\tau \cdot \tau + (1 - p_\tau) \cdot \sum_{i \le n} \mathbf{E}\left[(v_i - \tau)^+\right] \\
&\ge p_\tau \cdot \tau + (1 - p_\tau) \cdot \left(\mathbf{E}\left[\max_i v_i\right] - \tau \right)
\end{aligned}
$$

Last inequality holds since $\max_i v_i \le \tau + \max_i (v_i - \tau)^+ \le \tau + \sum_i (v_i - \tau)^+$.

Prophet Inequalities

Theorem

There exists a threshold policy that achieves a 2-approximation, i.e., it achieves expected value at least $\frac{1}{2}\mathbf{E}[\max_i v_i]$.

$$\mathrm{ALG}_\tau \geq p_\tau \cdot \tau + (1 - p_\tau) \cdot \left(\mathbf{E}\left[\max_i v_i\right] - \tau \right).$$

Prophet Inequalities

Theorem

There exists a threshold policy that achieves a 2*-approximation, i.e., it achieves expected value at least* $\frac{1}{2}\mathbf{E}[\max_i v_i]$.

$$\mathrm{ALG}_\tau \geq p_\tau \cdot \tau + (1 - p_\tau) \cdot \left(\mathbf{E}\left[\max_i v_i\right] - \tau \right).$$

- Mean Rule: Let $\tau = \frac{1}{2}\mathbf{E}[\max_i v_i]$. We have

$$\mathrm{ALG}_\tau \geq p_\tau \cdot \frac{1}{2}\mathbf{E}\left[\max_i v_i\right] + (1 - p_\tau) \cdot \frac{1}{2}\mathbf{E}\left[\max_i v_i\right] = \frac{1}{2}\mathbf{E}\left[\max_i v_i\right].$$

Prophet Inequalities

Theorem

There exists a threshold policy that achieves a 2-approximation, i.e., it achieves expected value at least $\frac{1}{2}\mathbf{E}[\max_i v_i]$.

$$\mathrm{ALG}_\tau \geq p_\tau \cdot \tau + (1 - p_\tau) \cdot \left(\mathbf{E}\left[\max_i v_i\right] - \tau \right).$$

- Mean Rule: Let $\tau = \frac{1}{2}\mathbf{E}[\max_i v_i]$. We have

$$\mathrm{ALG}_\tau \geq p_\tau \cdot \frac{1}{2}\mathbf{E}\left[\max_i v_i\right] + (1 - p_\tau) \cdot \frac{1}{2}\mathbf{E}\left[\max_i v_i\right] = \frac{1}{2}\mathbf{E}\left[\max_i v_i\right].$$

- Median Rule: Let τ such that $p_\tau = \frac{1}{2}$. We have

$$\mathrm{ALG}_\tau \geq \frac{1}{2}\tau + \frac{1}{2}\left(\mathbf{E}\left[\max_i v_i\right] - \tau \right) = \frac{1}{2}\mathbf{E}\left[\max_i v_i\right].$$

Hard Instances

Can we do better than 2? No!

Hard Instances

Can we do better than 2? No!

Example: two items.

- Item 1: $v_1 = 1$ with probability 1.
- Item 2: $v_2 = z$ w.p. $\frac{1}{z}$, and 0 otherwise.

Hard Instances

Can we do better than 2? No!

Example: two items.

- Item 1: $v_1 = 1$ with probability 1.
- Item 2: $v_2 = z$ w.p. $\frac{1}{z}$, and 0 otherwise.

Any Online Policy:

- If item 1 is chosen, the expected value is $v_1 = 1$.
- If item 1 is not chosen, the expected value is at most $\mathbf{E}[v_2] = 1$.

Hard Instances

Can we do better than 2? No!

Example: two items.

- Item 1: $v_1 = 1$ with probability 1.
- Item 2: $v_2 = z$ w.p. $\frac{1}{z}$, and 0 otherwise.

Any Online Policy:

- If item 1 is chosen, the expected value is $v_1 = 1$.
- If item 1 is not chosen, the expected value is at most $\mathbf{E}[v_2] = 1$.

Prophet: select item 1 if and only if $v_2 = 0$. The expected value of the prophet is $z \cdot \frac{1}{z} + (1 - \frac{1}{z}) \cdot 1 = 2 - \frac{1}{z}$.

Hard Instances

Can we do better than 2? No!

Example: two items.

- Item 1: $v_1 = 1$ with probability 1.
- Item 2: $v_2 = z$ w.p. $\frac{1}{z}$, and 0 otherwise.

Any Online Policy:

- If item 1 is chosen, the expected value is $v_1 = 1$.
- If item 1 is not chosen, the expected value is at most $\mathbf{E}[v_2] = 1$.

Prophet: select item 1 if and only if $v_2 = 0$. The expected value of the prophet is $z \cdot \frac{1}{z} + (1 - \frac{1}{z}) \cdot 1 = 2 - \frac{1}{z}$.

The gaps is 2 when $z \to \infty$.

Connection to Auctions

Prophet inequality: n items

- value distributions $F = F_1 \times \cdots \times F_n$;
- threshold τ;
- arrival order π.

Posted pricing mechanism: n agents

- value distributions $F = F_1 \times \cdots \times F_n$;
- price $p_i = \tau$ for each agent i;
- tie breaking rule π.

Connection to Auctions

Prophet inequality: n items

- value distributions $F = F_1 \times \cdots \times F_n$;
- threshold τ;
- arrival order π.

Posted pricing mechanism: n agents

- value distributions $F = F_1 \times \cdots \times F_n$;
- price $p_i = \tau$ for each agent i;
- tie breaking rule π.

Given any valuation profile $v = (v_1, \ldots, v_n)$, the selected value and the optimal value in both problems are the same.

- Posted pricing mechanism has a 2-approximation to the optimal welfare.

Connection to Auctions

Prophet inequality: n items

- value distributions $F = F_1 \times \cdots \times F_n$;
- threshold τ;
- arrival order π.

Posted pricing mechanism: n agents

- value distributions $F = F_1 \times \cdots \times F_n$;
- price $p_i = \tau$ for each agent i;
- tie breaking rule π.

Given any valuation profile $v = (v_1, \ldots, v_n)$, the selected value and the optimal value in both problems are the same.

- Posted pricing mechanism has a 2-approximation to the optimal welfare.

Question: how do we evaluation this approximation?

- is 2 a good approximation or a bad approximation?

Asymptotic Analysis

- $f(n) = O(g(n)) : \lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$;
- $f(n) = \Omega(g(n)) : \lim_{n\to\infty} \frac{f(n)}{g(n)} > 0$.
- $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$;
- $f(n) = o(g(n))$ if $f(n) = O(g(n))$ and $f(n) \neq \Omega(g(n))$;
- $f(n) = \omega(g(n))$ if $f(n) \neq O(g(n))$ and $f(n) = \Omega(g(n))$;

In the context of auctions, n can be viewed as the number of agents.

Asymptotic Analysis

- $f(n) = O(g(n)) : \lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$;
- $f(n) = \Omega(g(n)) : \lim_{n\to\infty} \frac{f(n)}{g(n)} > 0$.
- $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$;
- $f(n) = o(g(n))$ if $f(n) = O(g(n))$ and $f(n) \neq \Omega(g(n))$;
- $f(n) = \omega(g(n))$ if $f(n) \neq O(g(n))$ and $f(n) = \Omega(g(n))$;

In the context of auctions, n can be viewed as the number of agents.

Example:

- $2n^2 + 8n + 100 = O(n^2)$;
- $16n^3 = o(2^n)$.
- $4n - 32 = \Theta(n)$.
- $\log(n) = o(n^\epsilon)$ for any constant $\epsilon > 0$.

Asymptotic Analysis

A mechanism M has a constant approximation if $\mathrm{APX}(M) = O(1)$.

- usually we view constant approximation as a good approximation since the worst-case performance does not degrade as the problem instance grows large ($n \to \infty$).

Asymptotic Analysis

A mechanism M has a constant approximation if $\mathrm{APX}(M) = O(1)$.

- usually we view constant approximation as a good approximation since the worst-case performance does not degrade as the problem instance grows large ($n \to \infty$).

Usually an approximation is not ideal if it is a super-constant, i.e., $\mathrm{APX}(M) = \omega(1)$.

- E.g., $\mathrm{APX}(M) = \Theta(\log(n))$, or $\mathrm{APX}(M) = \Theta(n^2)$.

Asymptotic Analysis

A mechanism M has a constant approximation if $\mathrm{APX}(M) = O(1)$.

- usually we view constant approximation as a good approximation since the worst-case performance does not degrade as the problem instance grows large ($n \to \infty$).

Usually an approximation is not ideal if it is a super-constant, i.e., $\mathrm{APX}(M) = \omega(1)$.

- E.g., $\mathrm{APX}(M) = \Theta(\log(n))$, or $\mathrm{APX}(M) = \Theta(n^2)$.

Posted pricing mechanism is a 2-approximation to the optimal welfare: great!

First-Price Auction

First-price Auction: Each bidder i places a bid $b_i \geq 0$ in the auction.

- highest bidder wins where ties are broken uniform randomly;
- winner pays his bid.

First-Price Auction

First-price Auction: Each bidder i places a bid $b_i \geq 0$ in the auction.

- highest bidder wins where ties are broken uniform randomly;
- winner pays his bid.

No simple characterization of equilibrium behaviors.

- the equilibrium outcome is often inefficient.

First-Price Auction

First-price Auction: Each bidder i places a bid $b_i \geq 0$ in the auction.

- highest bidder wins where ties are broken uniform randomly;
- winner pays his bid.

No simple characterization of equilibrium behaviors.

- the equilibrium outcome is often inefficient.

Question: what is the maximum inefficiency of first-price auction.

First-Price Auction

Theorem (Jin and Lu '22)

The first-price auction is an $\frac{e^2}{e^2-1} \approx 1.16$ *approximation to optimal welfare.*

First-Price Auction

Theorem (Jin and Lu '22)

The first-price auction is an $\frac{e^2}{e^2-1} \approx 1.16$ *approximation to optimal welfare.*

A simpler proof to show that the first price auction is a 2-approximation to optimal welfare.

First-Price Auction

Theorem (Jin and Lu '22)

The first-price auction is an $\frac{e^2}{e^2-1} \approx 1.16$ *approximation to optimal welfare.*

A simpler proof to show that the first price auction is a 2-approximation to optimal welfare.

Intuition: we don't know how the agents behave, but we know they should not perform too bad in equilibrium.

2-approximation of First-Price Auction

For each agent i, one possible strategy is to bid $b_i^* = \frac{v_i}{2}$ regardless of the opponents' strategy.

$$u_i(b_i^*, \mathbf{b}_{-i}; v_i) \geq \frac{1}{2} v_i - p(\mathbf{b}).$$

since the bidder either wins and obtains utility $v_i - b_i^* = v_i - \frac{1}{2}v_i = \frac{1}{2}v_i \geq \frac{1}{2}v_i - p(\mathbf{b})$, or loses and obtains utility $0 \geq \frac{1}{2}v_i - p(\mathbf{b})$.

2-approximation of First-Price Auction

For each agent i, one possible strategy is to bid $b_i^* = \frac{v_i}{2}$ regardless of the opponents' strategy.

$$u_i(b_i^*, \mathbf{b}_{-i}; v_i) \geq \frac{1}{2}v_i - p(\mathbf{b}).$$

since the bidder either wins and obtains utility $v_i - b_i^* = v_i - \frac{1}{2}v_i = \frac{1}{2}v_i \geq \frac{1}{2}v_i - p(\mathbf{b})$, or loses and obtains utility $0 \geq \frac{1}{2}v_i - p(\mathbf{b})$.

Let x_i^* be the welfare optimal allocation. Since the bid $b_i^* = \frac{v_i}{2}$ guarantees non-negative utility,

$$u_i(b_i^*, \mathbf{b}_{-i}; v_i) \geq \left(\frac{1}{2}v_i - p(\mathbf{b})\right) \cdot x_i^*(\mathbf{v}).$$

2-approximation of First-Price Auction

For each agent i, one possible strategy is to bid $b_i^* = \frac{v_i}{2}$ regardless of the opponents' strategy.

$$u_i(b_i^*, \mathbf{b}_{-i}; v_i) \geq \frac{1}{2} v_i - p(\mathbf{b}).$$

since the bidder either wins and obtains utility $v_i - b_i^* = v_i - \frac{1}{2}v_i = \frac{1}{2}v_i \geq \frac{1}{2}v_i - p(\mathbf{b})$, or loses and obtains utility $0 \geq \frac{1}{2}v_i - p(\mathbf{b})$.

Let x_i^* be the welfare optimal allocation. Since the bid $b_i^* = \frac{v_i}{2}$ guarantees non-negative utility,

$$u_i(b_i^*, \mathbf{b}_{-i}; v_i) \geq \left(\frac{1}{2} v_i - p(\mathbf{b})\right) \cdot x_i^*(\mathbf{v}).$$

Summing this inequality over all bidders i, we obtain

$$\sum_{i=1}^{n} u_i(b_i^*, \mathbf{b}_{-i}; v_i) \geq \sum_{i=1}^{n} \left(\frac{1}{2} v_i - p(\mathbf{b})\right) \cdot x_i^*(\mathbf{v}) = \frac{1}{2}\mathsf{OPT}(\mathbf{v}) - p(\mathbf{b}),$$

for every valuation profile $\mathbf{v}$ and bid profile $\mathbf{b}$.

2-approximation of First-Price Auction

Let s be a Bayes-Nash equilibrium: for every player i with valuation v_i,

$$\mathbb{E}_{\mathbf{v}_{-i}}\left[u_i(s(\mathbf{v}); v_i)\right] \geq \mathbb{E}_{\mathbf{v}_{-i}}\left[u_i(b_i^*, s_{-i}(\mathbf{v}_{-i}); v_i)\right].$$

2-approximation of First-Price Auction

Let s be a Bayes-Nash equilibrium: for every player i with valuation v_i,

$$\mathbb{E}_{\mathbf{v}_{-i}}\left[u_i(s(\mathbf{v}); v_i)\right] \geq \mathbb{E}_{\mathbf{v}_{-i}}\left[u_i(b_i^*, s_{-i}(\mathbf{v}_{-i}); v_i)\right].$$

Taking expectations over v_i and summing it up for all n agents, we obtain

$$\sum_{i=1}^{n} \mathbb{E}_{\mathbf{v}}\left[u_i(s(\mathbf{v}); v_i)\right] \geq \sum_{i=1}^{n} \mathbb{E}_{\mathbf{v}}\left[u_i(b_i^*, s_{-i}(\mathbf{v}_{-i}); v_i)\right] \geq \mathbb{E}_{\mathbf{v}}\left[\frac{1}{2}\mathsf{OPT}(\mathbf{v}) - p(s(\mathbf{v}))\right].$$

2-approximation of First-Price Auction

Let $\mathbf{s}$ be a Bayes-Nash equilibrium: for every player i with valuation v_i,

$$\mathbb{E}_{\mathbf{v}_{-i}}\left[u_i(s(\mathbf{v});v_i)\right] \geq \mathbb{E}_{\mathbf{v}_{-i}}\left[u_i(b_i^*, s_{-i}(\mathbf{v}_{-i});v_i)\right].$$

Taking expectations over v_i and summing it up for all n agents, we obtain

$$\sum_{i=1}^{n}\mathbb{E}_{\mathbf{v}}\left[u_i(s(\mathbf{v});v_i)\right] \geq \sum_{i=1}^{n}\mathbb{E}_{\mathbf{v}}\left[u_i(b_i^*, s_{-i}(\mathbf{v}_{-i});v_i)\right] \geq \mathbb{E}_{\mathbf{v}}\left[\frac{1}{2}\mathsf{OPT}(\mathbf{v}) - p(s(\mathbf{v}))\right].$$

Note that for every bid profile $\mathbf{b}$ and valuation profile $\mathbf{v}$, we have

$$\sum_{i=1}^{n} u_i(\mathbf{b};v_i) = SW(\mathbf{b};\mathbf{v}) - p(\mathbf{b}).$$

2-approximation of First-Price Auction

Let $\mathbf{s}$ be a Bayes-Nash equilibrium: for every player i with valuation v_i,

$$\mathbb{E}_{\mathbf{v}_{-i}}\left[u_i(s(\mathbf{v}); v_i)\right] \geq \mathbb{E}_{\mathbf{v}_{-i}}\left[u_i(b_i^*, s_{-i}(\mathbf{v}_{-i}); v_i)\right].$$

Taking expectations over v_i and summing it up for all n agents, we obtain

$$\sum_{i=1}^{n} \mathbb{E}_{\mathbf{v}}\left[u_i(s(\mathbf{v}); v_i)\right] \geq \sum_{i=1}^{n} \mathbb{E}_{\mathbf{v}}\left[u_i(b_i^*, s_{-i}(\mathbf{v}_{-i}); v_i)\right] \geq \mathbb{E}_{\mathbf{v}}\left[\frac{1}{2}\mathsf{OPT}(\mathbf{v}) - p(s(\mathbf{v}))\right].$$

Note that for every bid profile $\mathbf{b}$ and valuation profile $\mathbf{v}$, we have

$$\sum_{i=1}^{n} u_i(\mathbf{b}; v_i) = SW(\mathbf{b}; \mathbf{v}) - p(\mathbf{b}).$$

Combining the inequalities yields

$$\mathbb{E}_{\mathbf{v}}\left[SW(s(\mathbf{v}); \mathbf{v})\right] = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{v}}\left[u_i(s(\mathbf{v}); v_i)\right] + \mathbb{E}_{\mathbf{v}}\left[p(s(\mathbf{v}))\right] \geq \frac{1}{2}\mathbb{E}_{\mathbf{v}}\left[\mathsf{OPT}(\mathbf{v})\right].$$

Mechanism Design

Mechanism Design

A mechanism design instance is denoted as $\Gamma_M = \left(N, \Omega, (u_i)_{i\in N}, (\Theta_i)_{i\in N}, \mu\right)$ where

- N is the set of players;
- Ω is the set of outcomes;
- Θ_i is the set of player i's "types" where $\theta_i \in \Theta_i$ is private information of i;
- $u_i : \Omega \times \Theta \to \mathbb{R}$ is player i's payoff function;
- $\mu(\theta)$ is the probability that a type profile $\theta \in \Theta$ occurs.

VCG Mechanisms

VCG mechanism: mechanism that implements efficient allocation in general environment.

- **allocation**: chooses outcome

$$\omega^* = \operatorname*{argmax}_{\omega \in \Omega} \sum_i v_i(\omega, \theta_i).$$

VCG Mechanisms

VCG mechanism: mechanism that implements efficient allocation in general environment.

- **allocation**: chooses outcome

$$\omega^* = \operatorname*{argmax}_{\omega \in \Omega} \sum_i v_i(\omega, \theta_i).$$

- **payment**: each agent i pays his externality on the welfare

$$p_i(\theta) = \max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) - \sum_{j \neq i} v_j(\omega^*, \theta_j) \geq 0.$$

VCG Mechanisms

Agent i's utility in VCG mechanism:

$$v_i(\omega^*, \theta_i) - \left(\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) - \sum_{j \neq i} v_j(\omega^*, \theta_j) \right)$$
$$= \sum_j v_j(\omega^*, \theta_j) - \max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) \geq 0.$$

VCG Mechanisms

Agent i's utility in VCG mechanism:

$$v_i(\omega^*, \theta_i) - \left(\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) - \sum_{j \neq i} v_j(\omega^*, \theta_j) \right)$$
$$= \sum_j v_j(\omega^*, \theta_j) - \max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) \geq 0.$$

Agent i's utility is maximized by truthfully reporting his type to choose the allocation ω^* that maximizes the welfare.

VCG Mechanisms

In the special case of single-item auction: item is allocated to the highest bidder

$$p_i(\theta) = \max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) - \sum_{j \neq i} v_j(\omega^*, \theta_j).$$

VCG Mechanisms

In the special case of single-item auction: item is allocated to the highest bidder

$$p_i(\theta) = \max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) - \sum_{j \neq i} v_j(\omega^*, \theta_j).$$

If i is the highest bidder:

- $\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j)$ is the second highest bid;

VCG Mechanisms

In the special case of single-item auction: item is allocated to the highest bidder

$$p_i(\theta) = \max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) - \sum_{j \neq i} v_j(\omega^*, \theta_j).$$

If i is the highest bidder:

- $\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j)$ is the second highest bid;
- $\sum_{j \neq i} v_j(\omega^*, \theta_j) = 0$.

VCG Mechanisms

In the special case of single-item auction: item is allocated to the highest bidder

$$p_i(\theta) = \max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) - \sum_{j \neq i} v_j(\omega^*, \theta_j).$$

If i is the highest bidder:

- $\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j)$ is the second highest bid;
- $\sum_{j \neq i} v_j(\omega^*, \theta_j) = 0.$

If i is not the highest bidder:

- $\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j)$ is the highest bid;

VCG Mechanisms

In the special case of single-item auction: item is allocated to the highest bidder

$$p_i(\theta) = \max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) - \sum_{j \neq i} v_j(\omega^*, \theta_j).$$

If i is the highest bidder:

- $\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j)$ is the second highest bid;
- $\sum_{j \neq i} v_j(\omega^*, \theta_j) = 0$.

If i is not the highest bidder:

- $\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j)$ is the highest bid;
- $\sum_{j \neq i} v_j(\omega^*, \theta_j)$ is also the highest bid.

VCG Mechanisms

In the special case of single-item auction: item is allocated to the highest bidder

$$p_i(\theta) = \max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j) - \sum_{j \neq i} v_j(\omega^*, \theta_j).$$

If i is the highest bidder:

- $\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j)$ is the second highest bid;
- $\sum_{j \neq i} v_j(\omega^*, \theta_j) = 0$.

If i is not the highest bidder:

- $\max_{\omega \in \Omega} \sum_{j \neq i} v_j(\omega, \theta_j)$ is the highest bid;
- $\sum_{j \neq i} v_j(\omega^*, \theta_j)$ is also the highest bid.

VCG mechanism reduces to the second-price auction.

Welfare Maximization

Implementing the VCG mechanism requires solving the optimal allocation problem:

$$\omega^* = \operatorname*{argmax}_{\omega \in \Omega} \sum_i v_i(\omega, \theta_i).$$

Welfare Maximization

Implementing the VCG mechanism requires solving the optimal allocation problem:

$$\omega^* = \operatorname*{argmax}_{\omega \in \Omega} \sum_i v_i(\omega, \theta_i).$$

Is this tractable in practice?

Welfare Maximization

Implementing the VCG mechanism requires solving the optimal allocation problem:

$$\omega^* = \operatorname*{argmax}_{\omega \in \Omega} \sum_i v_i(\omega, \theta_i).$$

Is this tractable in practice?

Example: (Knapsack problem) consider the allocation problem of servicing agents, where $\Omega \subseteq 2^N$.

- each agent has private value θ_i for being serviced;
- servicing each agent i requires a resource of r_i;
- there is a total budget of B on resource;
- allocation ω is feasible if and only if $\sum_{i \in \omega} r_i \leq B$.

Welfare Maximization

Implementing the VCG mechanism requires solving the optimal allocation problem:

$$\omega^* = \operatorname*{argmax}_{\omega \in \Omega} \sum_i v_i(\omega, \theta_i).$$

Is this tractable in practice?

Example: (Knapsack problem) consider the allocation problem of servicing agents, where $\Omega \subseteq 2^N$.

- each agent has private value θ_i for being serviced;
- servicing each agent i requires a resource of r_i;
- there is a total budget of B on resource;
- allocation ω is feasible if and only if $\sum_{i \in \omega} r_i \leq B$.

How to find the optimal allocation? Trying all combination requires time exponential in $|N|$. Not practical if $n = |N|$ is large!

Running Time

An algorithm is a polynomial-time algorithm if there exists $c \in (0, \infty)$ such that its running time $f(n)$ satisfies $f(n) = O(n^c)$.

Running Time

An algorithm is a polynomial-time algorithm if there exists $c \in (0, \infty)$ such that its running time $f(n)$ satisfies $f(n) = O(n^c)$.

Enumerating all subsets of N is not a polynomial-time algorithm: $2^n = \omega(n^c)$ for any $c < \infty$.

Running Time

An algorithm is a polynomial-time algorithm if there exists $c \in (0, \infty)$ such that its running time $f(n)$ satisfies $f(n) = O(n^c)$.

Enumerating all subsets of N is not a polynomial-time algorithm: $2^n = \omega(n^c)$ for any $c < \infty$.

Under the assumption that P≠NP, the knapsack problem does not have any polynomial-time algorithm.

- There exist polynomial-time algorithms for approximating the optimal solutions.

Running Time

An algorithm is a polynomial-time algorithm if there exists $c \in (0, \infty)$ such that its running time $f(n)$ satisfies $f(n) = O(n^c)$.

Enumerating all subsets of N is not a polynomial-time algorithm: $2^n = \omega(n^c)$ for any $c < \infty$.

Under the assumption that P$\neq$NP, the knapsack problem does not have any polynomial-time algorithm.

- There exist polynomial-time algorithms for approximating the optimal solutions.

Question: does there exist polynomial-time mechanism that guarantees good welfare approximations?