Algorithmic game theory

Martin Balko

12th lecture

January 6th 2026

Proof of Myerson's lemma

Myerson's lemma

Myerson's lemma (Theorem 3.8)

In a single-parameter environment, the following three claims hold.

(a) An allocation rule is implementable if and only if it is monotone.

(b) If an allocation rule x is monotone, then there exists a unique payment rule p such that the mechanism (x, p) is DSIC (assuming that $b_i = 0$ implies $p_i(b) = 0$).

(c) The payment rule p is given by the following explicit formula

$$p_i(b_i; b_{-i}) = \int_0^{b_i} z \cdot \frac{\mathrm{d}}{\mathrm{d}z} x_i(z; b_{-i}) \, \mathrm{d}z$$

for every $i \in \{1, \ldots, n\}$.

- We have applied this result many times, but we have not seen its proof yet. Let's fix that.

Proof of Myerson's lemma I

- Let x be an allocation rule and p be a payment rule such that (x, p) is DSIC. We prove all three claims at once use a clever swapping trick.
- The DSIC property says that, for every z,
  $u_i(z; b_{-i}) = v_i \cdot x_i(z; b_{-i}) - p_i(z; b_{-i}) \leq v_i \cdot x_i(v_i; b_{-i}) - p_i(v_i; b_{-i})$.
- For two possible bids y and z with $0 \leq y < z$, bidder i might as well have private valuation z and can submit the false bid y if he wants, thus the DSIC condition gives

$$u_i(y; b_{-i}) = z \cdot x_i(y; b_{-i}) - p_i(y; b_{-i}) \leq z \cdot x_i(z; b_{-i}) - p_i(z; b_{-i}) = u_i(z; b_{-i}).$$

- Analogously, we can have $v_i = y$ and $b_i = z$ and thus (x, p) satisfies

$$u_i(z; b_{-i}) = y \cdot x_i(z; b_{-i}) - p_i(z; b_{-i}) \leq y \cdot x_i(y; b_{-i}) - p_i(y; b_{-i}) = u_i(y; b_{-i}).$$

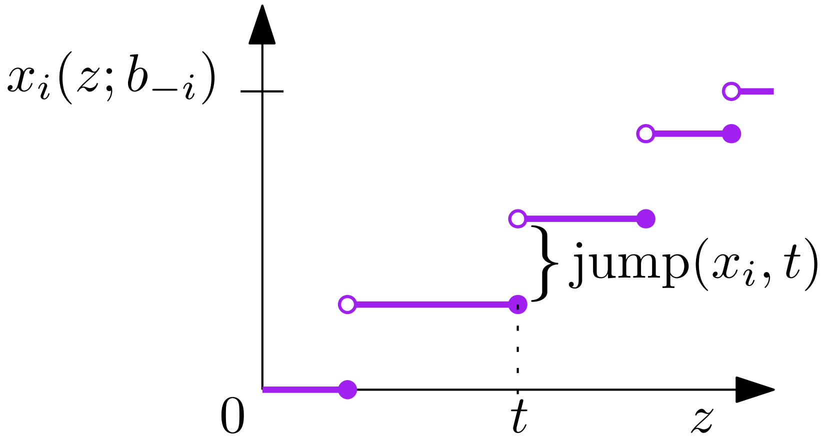
- By putting these inequalities together, we obtain the following payment difference sandwich:

$$z(x_i(y; b_{-i}) - x_i(z; b_{-i})) \leq p_i(y; b_{-i}) - p_i(z; b_{-i}) \leq y(x_i(y; b_{-i}) - x_i(z; b_{-i})).$$

- Since $0 \leq y < z$, we obtain $x_i(y; b_{-i}) \leq x_i(z; b_{-i})$. Thus, if (x, p) is DSIC, then x is monotone.

Proof of Myerson's lemma II

- In the rest of the proof, we assume that the allocation x is monotone.
- Let i and b_{-i} be fixed, so we consider x_i and p_i as functions of z.
- First, we also assume that the function x_i is piecewise constant. Thus, the graph of x_i consists of a finite number of intervals with "jumps" between consecutive intervals:



- For a piecewise constant function f, we use $\mathrm{jump}(f, t)$ to denote the magnitude of the jump of f at point t.
- If we fix z in the payment difference sandwich and let y approach z from below, then both sides become 0 if there is no jump of x_i at z. If $\mathrm{jump}(x_i, z) = h > 0$, then both sides tend to $z \cdot h$.

Proof of Myerson's lemma III

- Thus, if the mechanism (x, p) is supposed to be DSIC, then the following constraint on p must hold for every z:

$$\mathrm{jump}(p_i, z) = z \cdot \mathrm{jump}(x_i, z).$$

- If we combine this constraint with the initial condition $p_i(0; b_{-i}) = 0$, we obtain a formula for the payment function p for every bidder i and bids b_{-i} of other bidders,

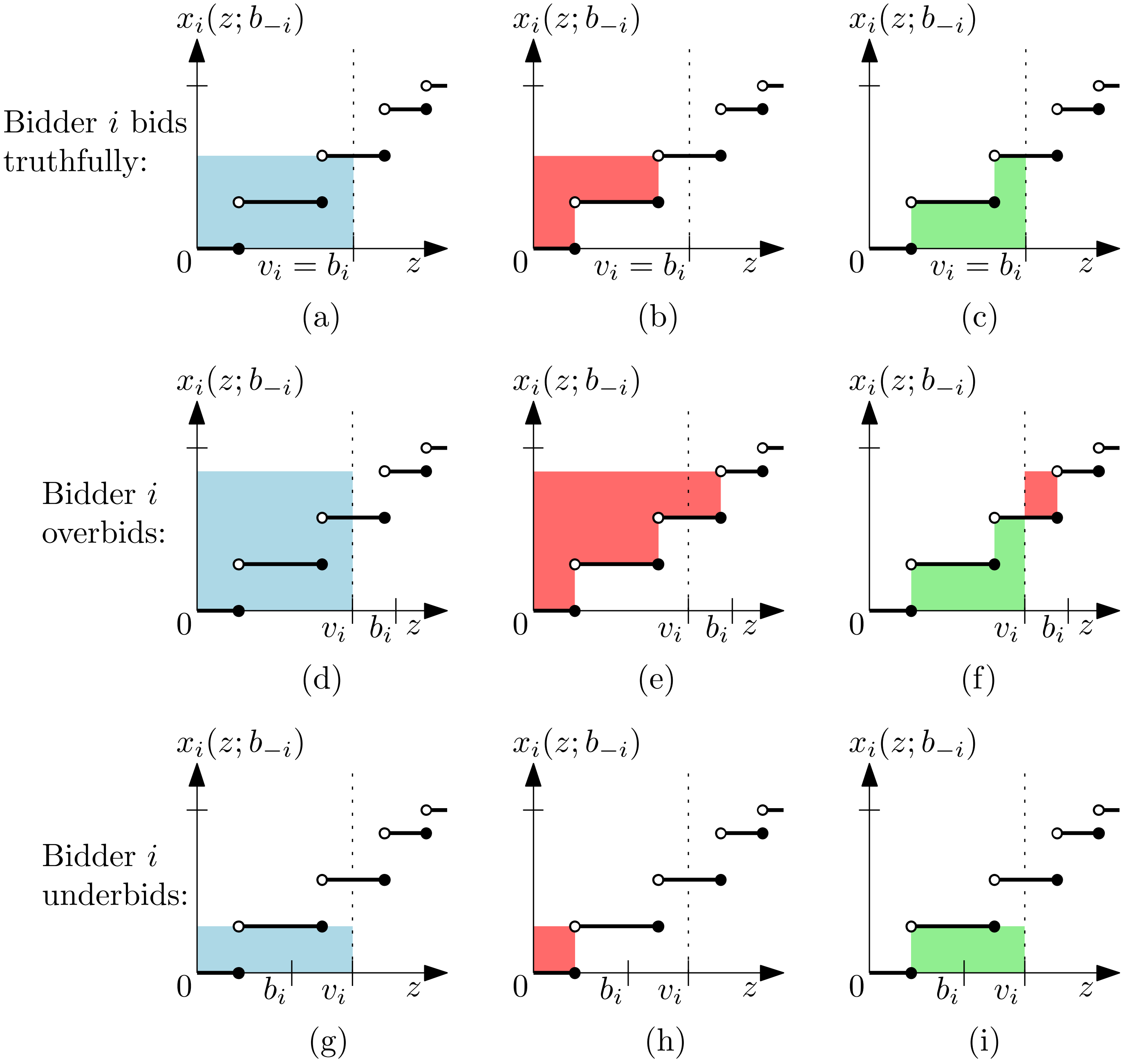
$$p_i(b_i; b_{-i}) = \sum_{j=1}^{\ell} z_j \cdot \mathrm{jump}(x_i(\,\cdot\,; b_{-i}), z_j),$$

  where $z_1, \ldots, z_\ell$ are the breakpoints of the allocation function $x_i(\,\cdot\,; b_{-i})$ in the interval $[0, b_i]$.
- With some additional facts from calculus, this argument can be generalized to general monotone functions x_i. We omit the details.

Proof of Myerson's lemma IV

- It remains to show that if x is monotone, then the mechanism (x, p) is indeed DSIC.
- This argument works in general, but we present it only for piecewise constant functions.
- We recall that the utility $u_i(b_i; b_{-i}) = v_i \cdot x_i(b_i; b_{-i}) - p_i(b_i; b_{-i})$.
- Using the expression of the payment, we see that the payment $p_i(b_i; b_{-i})$ of bidder i corresponds to the part of $[0, b_i] \times [0, x_i(b_i; b_{-i})]$ lying to the left of the curve $x_i(\,\cdot\,; b_{-i})$.
- It will follow from a picture that it is optimal for bidder i to bid $b_i = v_i$.

Proof of Myerson's lemma by picture



Knapsack auctions

Going back to social surplus

- Let us go back to mechanisms that maximize social surplus.
- We know what awesome mechanisms are in single-item auctions. We also have Myerson's lemma for designing DSIC mechanisms.
- Can we design an awesome mechanism for every single-parameter environment?

- An awesome mechanism satisfies the following three properties:
  - DSIC: everybody has dominant strategy "bid truthfully" which guarantees non-negative utility,
  - strong performance: we maximize social surplus if everybody bids truthfully,
  - computational efficiency: the mechanism runs in polynomial time.

Auction example: scheduling TV commercials

- Bidders are companies such that each company has its own TV commercial of length w_i and is willing to pay v_i in order to have the commercial presented during a commercial break. The seller is a television station with a commercial break of length W.

Sources: https://mountain.com/ and https://www.eq-international.com/

- Can we design an awesome mechanism that assigns the slots?

Knapsack auction

- We formalize this auction as follows.
- In a knapsack auction of n bidders $1, \ldots, n$, each bidder i has publicly known size $w_i \geq 0$ and a private valuation $v_i \geq 0$. There is a single seller who has a capacity $W \geq 0$. The feasible set X consists of $\{0, 1\}$-vectors $(x_1, \ldots, x_n)$ such that $\sum_{i=1}^{n} x_i w_i \leq W$, where $x_i = 1$ indicates that bidder i is a winning bidder.

- We now try to design an awesome mechanism for knapsack auctions.
  - For bids $b = (b_1, \ldots, b_n)$, we choose $x(b)$ from X such that $\sum_{i=1}^{n} b_i x_i$ is maximized. Then, when bidders bid truthfully, the social surplus is maximized.
  - The allocation rule x is monotone (one-step function with breakingpoint at some z) and thus Myerson's lemma gives us a payment rule p such that (x, p) is DSIC. If $b_i < z$, then bidder i pays nothing, otherwise he pays $z \cdot (1 - 0) = z$.
  - So we have the first two conditions satisfied. However, the third one will be problematic since x solves the Knapsack problem.

Knapsack problem

- given a capacity W and n items of values $v_1, \ldots, v_n$ and sizes $w_1, \ldots, w_n$, find a subset of the items having a maximum total value such that the total size is at most W.

Sources: https://en.wikipedia.org and https://twitter.com/

- This problem is NP-hard.
- There is a pseudo-polynomial time algorithm using dynamic programming and a fully polynomial-time approximation scheme.

We are not always awesome

- Assuming $\mathsf{P} \neq \mathsf{NP}$, we cannot satisfy the third condition (polynomial time), since the Knapsack problem is NP-hard, and choosing the allocation rule so that the social surplus is maximized solves it. Thus, knapsack auctions are not awesome.
- Relaxing the first condition (DSIC property) does not help, since it is the last two conditions that collide. We might relax the third condition, say to pseudopolynomial time. This is helpful if our instances are small or structured enough and we have enough time and computing power to implement optimal surplus-maximization.
- The dominant paradigm is to relax the second constraint (optimal surplus) as little as possible, subject to the first (DSIC) and the third (polynomial-time) constraints.
- Myerson's Lemma implies that the following goal is equivalent: design a polynomial-time and monotone allocation rule that comes as close as possible to maximizing the social surplus.

Approximation with monotonicity

- This resembles the primary goal in approximation: design algorithms for NP-hard problems that are as close to optimal as possible, subject to a polynomial-time constraint.
- For us, the algorithms must additionally obey a monotonicity constraint.
- The "holy grail" in algorithmic mechanism design: for as many NP-hard problems as possible, match the best-known approximation guarantee for (not necessarily monotone) approximate surplus maximization algorithms, subject to $P \neq NP$. That is, we would like the DSIC/monotone constraint to cause no additional surplus loss.

- We now illustrate this approach by designing an allocation rule that gives at least half of the optimum social surplus in knapsack auctions.

Greedy allocation for knapsack auctions

- We assume without loss of generality that no bidder i has $w_i > W$. We also assume that the bidders $1, \ldots, n$ are sorted in the order $<$ so that

$$\frac{b_1}{w_1} \geq \cdots \geq \frac{b_n}{w_n}.$$

- Consider the following greedy allocation rule $x^G = (x_1^G, \ldots, x_n^G) \in X$, which for given bids $b = (b_1, \ldots, b_n)$ selects a subset of bidders so that $\sum_{i=1}^n x_i^G w_i \leq W$ using the following procedure.
  - Pick winners in the order $<$ until one does not fit and then halt.
  - Return either the solution from the first step or the highest bidder, whichever creates more social surplus.
- The reason for the second step is that the solution in the first step might be highly suboptimal if there is a very valuable and very large bidder. Consider, for example, $n = 2$ with $b_1 = 2$, $w_1 = 1$, $b_2 = W$, and $w_2 = W$ for a very large W.
- The rule x_G is monotone (Exercise).

2-approximation for knapsack auctions I

Theorem 3.10

Assuming truthful bids, the social surplus of the greedy allocation rule x^G is at least one half of the maximum possible social surplus.

- Proof (sketch): Let $w_1, \ldots, w_n$ be the given sizes, $v_1, \ldots, v_n$ the valuations (and also the bids), and W be the capacity.
- First, we consider a relaxation of the problem, where we can choose each bidder i with fraction $\alpha_i \in [0, 1]$ so that i contributes with $\alpha_i \cdot v_i$ to the solution. The greedy algorithm to solve this fractional version: pick winners in the order $<$ until the capacity W is fully used with the possibility to pick the last winner fractionally, if needed.
- We show that this algorithm maximizes the surplus over all feasible solutions to the fractional knapsack problem.
    - Let $1, \ldots, k$ be the winners selected by the greedy algorithm and suppose for contradiction that there is another feasible solution that gives higher social surplus.

2-approximation for knapsack auctions II

- Then our solution can be improved by changing one of the constraints α_i to some larger β_i. Since $\alpha_1 = \cdots = \alpha_{k-1} = 1$, we have $i \geq k$. Since $\sum_{l=1}^{k} \alpha_l w_l = W$, there is a $j \in \{1, \ldots, k\}$ with $j < i$ and $\beta_j < \alpha_j$. We can assume that these are the only changed coefficients (Exercise).
- Then $(\beta_i - \alpha_i)w_i \leq (\alpha_j - \beta_j)w_j$, as $\sum_{l=1}^{k} \alpha_l w_l = W$ and we add $(\beta_i - \alpha_i)w_i$ to the size, while removing $(\alpha_j - \beta_j)w_j$.
- On the other hand, since the social surplus is now larger, we have $(\beta_i - \alpha_i)v_i > (\alpha_j - \beta_j)v_j$. By dividing the left side of the second inequality with the left side of the first inequality and doing the same for the right sides, we obtain $v_i/w_i > v_j/w_j$, which, since $j < i$, contradicts our choice of the order $<$.

- Now, assume that in the fractional setting the first $k-1$ winners i have $\alpha_i = 1$ while $\alpha_k < 1$. Then the social surplus achieved by step 1 is exactly $\sum_{i=1}^{k-1} \alpha_i v_i = \sum_{i=1}^{k-1} v_i$. The social surplus in step 2 is at least v_k.
- Thus, we have social surplus at least $\max\{v_k, \sum_{i=1}^{k-1} v_i\}$, at least half of the optimal fractional solution, which is at least the non-fractional optimum. $\square$

Multi-parameter mechanism design

Multi-parameter mechanism design

- In multi-parameter mechanism design, we have the following setting:
  - n strategic bidders,
  - a finite set Ω of outcomes,
  - each bidder i has a private valuation $v_i(\omega) \geq 0$ for every outcome $\omega \in \Omega$.
- Each bidder i submits his bids $b_i(\omega) \geq 0$ for each $\omega \in \Omega$ and our goal is to design a mechanism that selects an outcome $\omega \in \Omega$ so that it maximizes the social surplus $\sum_{i=1}^{n} v_i(\omega)$.
- The valuations now depend on possible outcomes, so, for example, if bidders compete for a single item, each bidder can have an opinion about each other bidder winning the item as well.
- Example (single-item auction): we set $\Omega = \{\omega_1, \ldots, \omega_n, \omega_\emptyset\}$ has size $n+1$ and each outcome ω_i with $i \in \mathbb{N}$ corresponds to the winner i of the item. The last outcome $\omega_\emptyset$ corresponds to nobody getting the item. The valuations are $v_i(\omega_j) = 0$ for every $j \neq i$ and $v_i(\omega_i) = v_i$ otherwise.

The Vickrey–Clarke–Groves (VCG) mechanism

VCG mechanism (Theorem 3.18)

In every multi-parameter mechanism design environment, there is a DSIC social-surplus-maximizing mechanism.

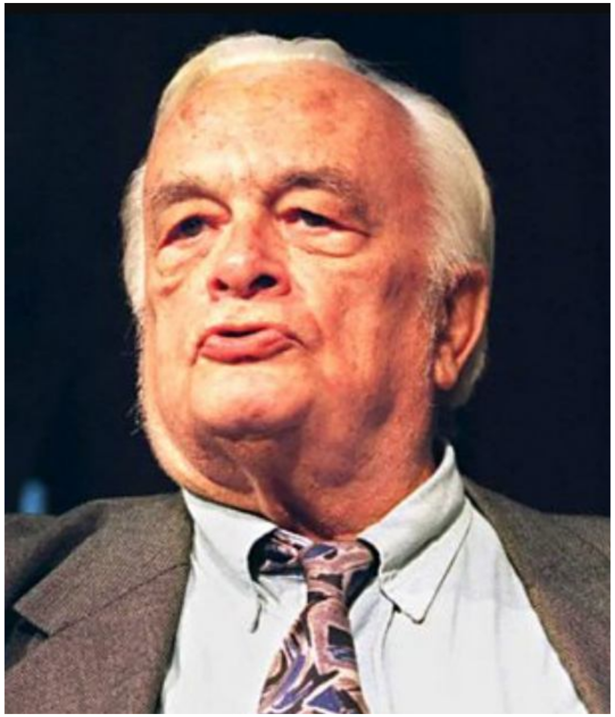


Figure: William Vickrey, Edward H. Clarke, and Theodore Groves.

Sources: : https://en.wikipedia.org, https://www.demandrevelation.com/, and https://www.researchate.net/

- We now present the proof.

VCG mechanism: proof idea

- The key idea is to consider the the loss of social surplus inflicted on the other $n-1$ bidders by the presence of bidder i. For example, in single-item auctions, the winning bidder inflicts a social surplus loss of the second-highest bid to the others.
- We define the payments to force each bidder to care about the others.
- We will see that the following allocation rule works

$$x(b) = \mathrm{argmax}_{\omega \in \Omega} \sum_{i=1}^{n} b_i(\omega)$$

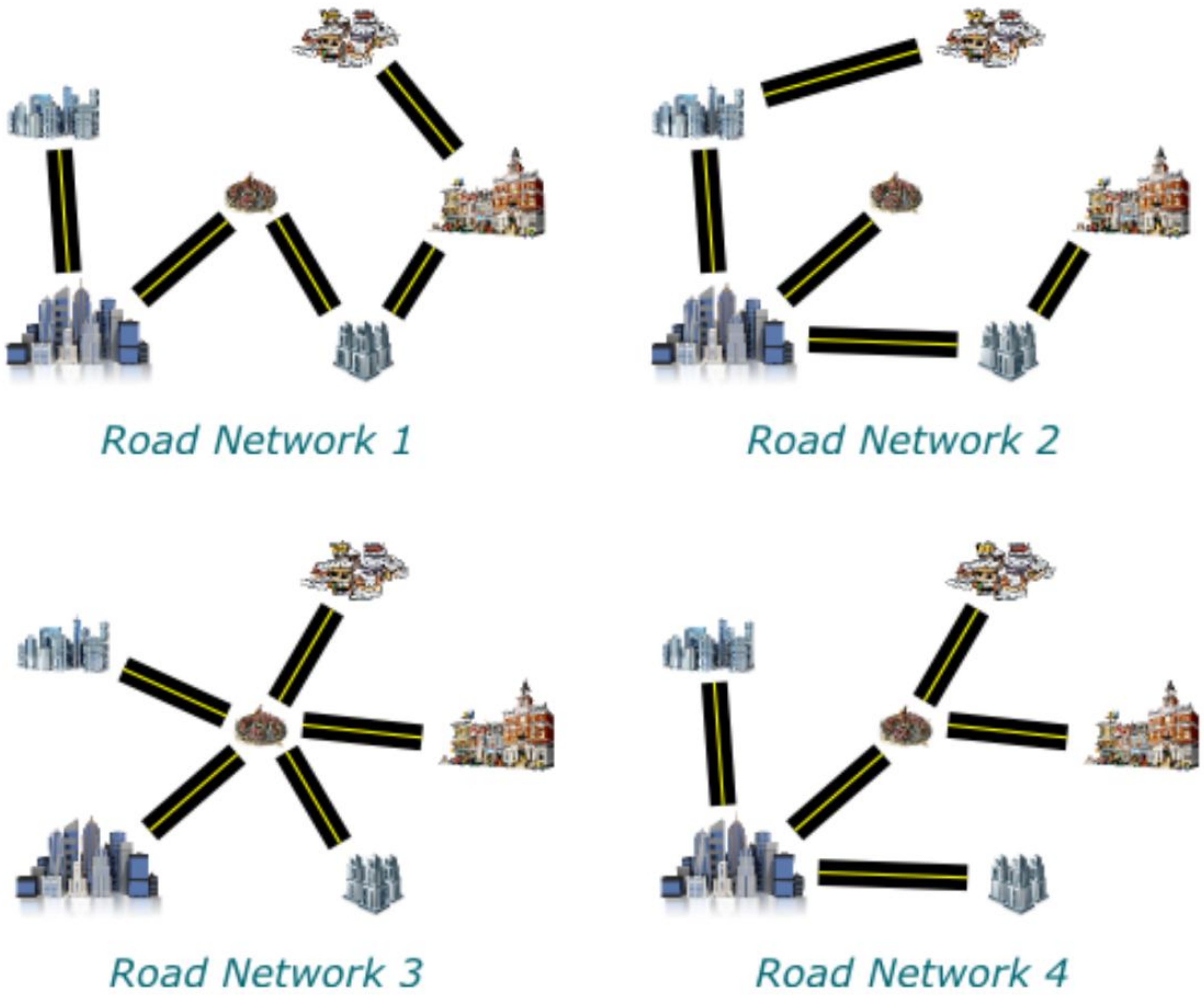
together with this payment formula

$$p_i(b) = \max_{\omega \in \Omega} \left\{ \sum_{\substack{j=1 \\ j \neq i}}^{n} b_j(\omega) \right\} - \sum_{\substack{j=1 \\ j \neq i}}^{n} b_j(\omega^*),$$

where $\omega^* = x(b)$ is the outcome chosen by our allocation rule x for given bids b.

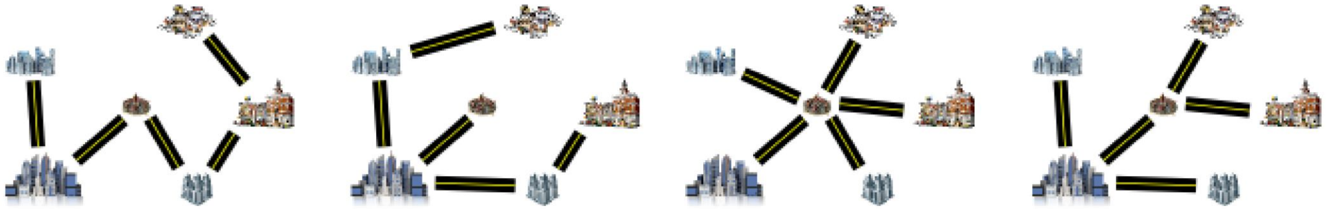
VCG auction example

- The government wants to construct roads connecting diverse cities, and he wants cities to pay for the roads.



Sources: https://www.science4all.org/article/auction-design/

VCG auction example

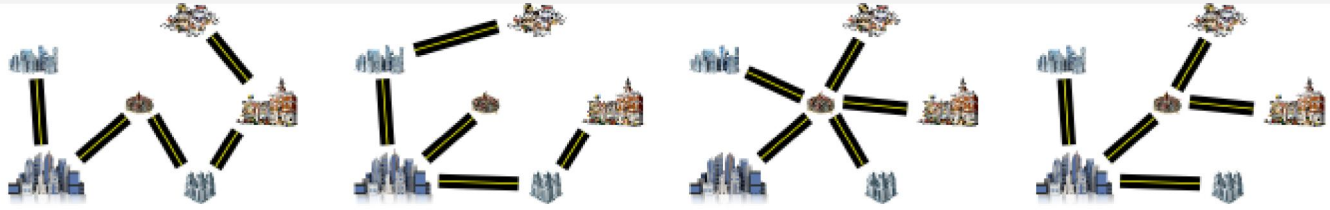


	Road Network 1	Road Network 2	Road Network 3	Road Network 4
	6 M$	14 M$	2 M$	**16 M$**
	5 M$	8 M$	4 M$	**12 M$**
	2 M$	1 M$	**20 M$**	4 M$
	4 M$	**6 M$**	3 M$	5 M$
	1 M$	1 M$	**6 M$**	2 M$
	1 M$	2 M$	2 M$	**3 M$**
Total (social welfare)	19 M$	32 M$	37 M$	**42 M$**

Sources: https://www.science4all.org/article/auction-design/

- Cities pay their negative externalities on the collectivity. Other cities would be happier without the biggest city (NYC, say). How much happier they would be is exactly what NYC must pay.

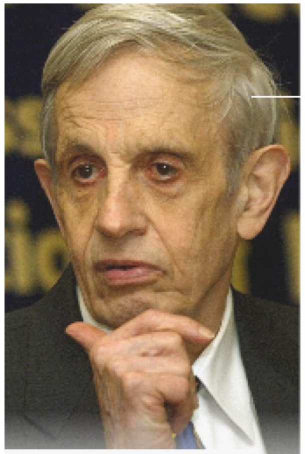
VCG auction example



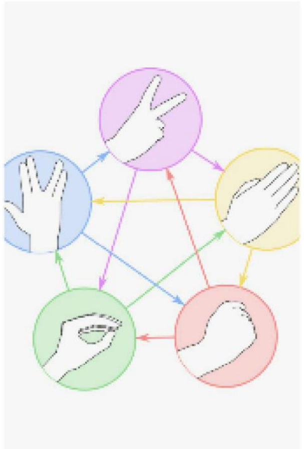
	Road Network 1	Road Network 2	Road Network 3	Road Network 4
	6 M$	14 M$	2 M$	**16 M$**
	5 M$	8 M$	4 M$	**12 M$**
	2 M$	1 M$	**20 M$**	4 M$
	4 M$	**6 M$**	3 M$	5 M$
	1 M$	1 M$	**6 M$**	2 M$
	1 M$	2 M$	2 M$	**3 M$**
Total (social welfare)	19 M$	32 M$	37 M$	**42 M$**

Sources: https://www.science4all.org/article/auction-design/

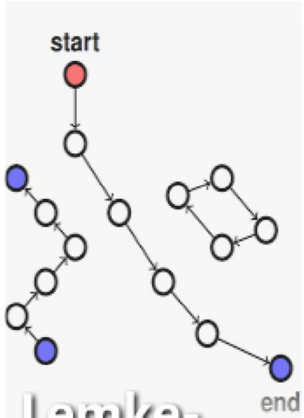
- If NYC was not there, then road network number 3 (RN3) would have been chosen, as opposed to RN4. The value of RN3 for the other cities would be 35 M$, as opposed to the 26 M$ of RN4. Therefore, the negative externality of NYC is $35 - 26 = 9$ M$.



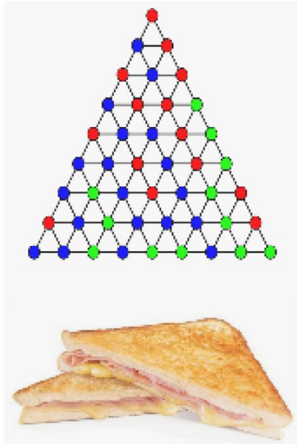
Nash equilibria



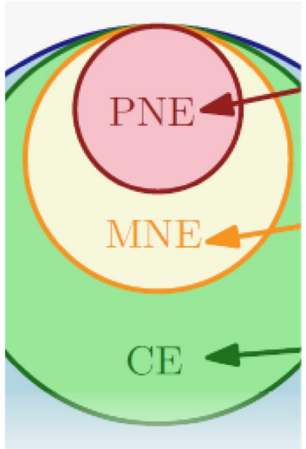
Minimax theorem



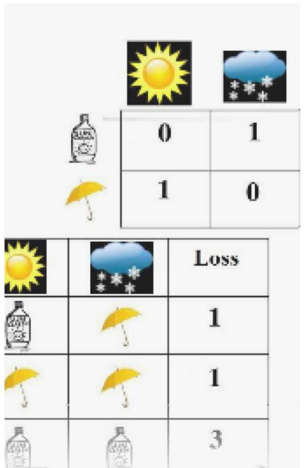
Lemke-Howson algorithm



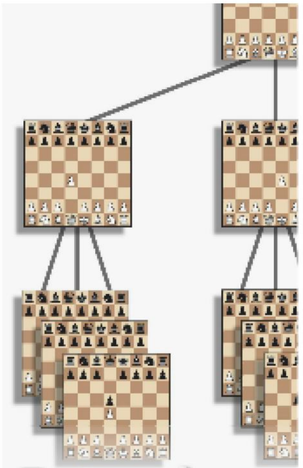
Complexity of NASH



Variants of NE

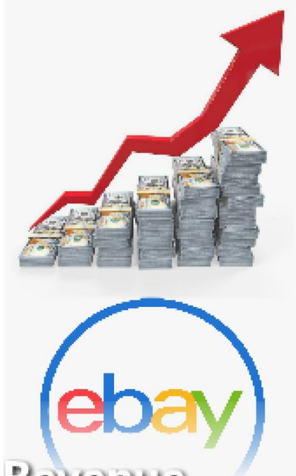


Regret minimization

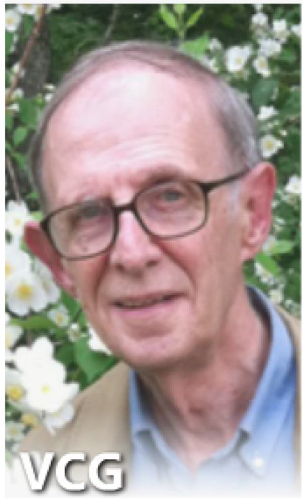


Extensive games

Mechanism design



Revenue maximization



VCG mechanism

Thank you for your attention.