New Frontiers of Automated Mechanism Design for Pricing and Auctions

Tuomas Sandholm and Ellen Vitercik
Carnegie Mellon University

Outline

1. Mechanism design basics
2. Introduction to automated mechanism design (AMD)
3. Sample complexity guarantees for AMD
4. Additional AMD algorithms
5. Other learning models

Mechanism design

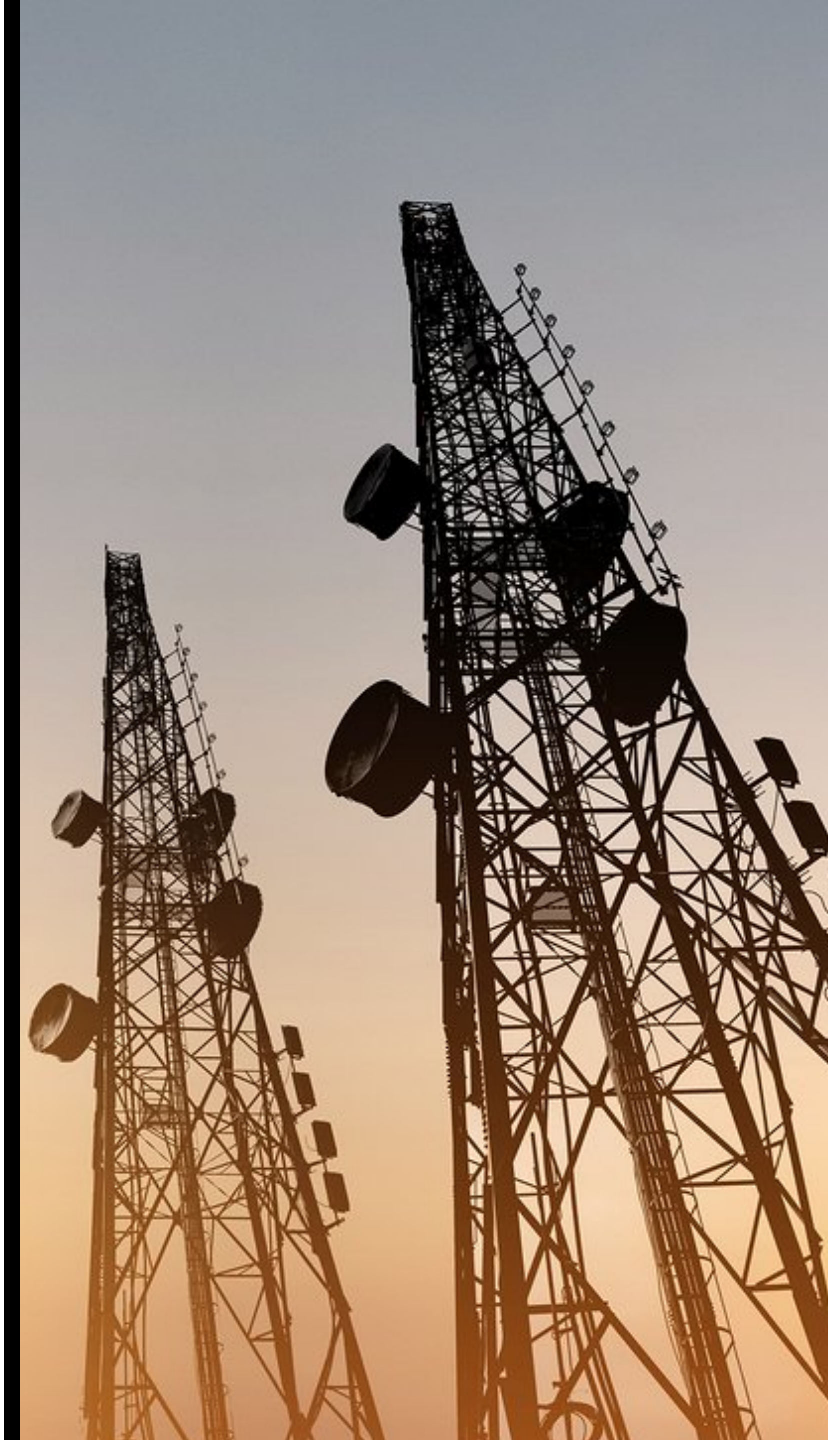
Field of game theory with significant real-world impact.

Encompasses areas such as pricing and auction design.

Very-large-scale generalized combinatorial multi-attribute auctions: Lessons from conducting $60B of sourcing

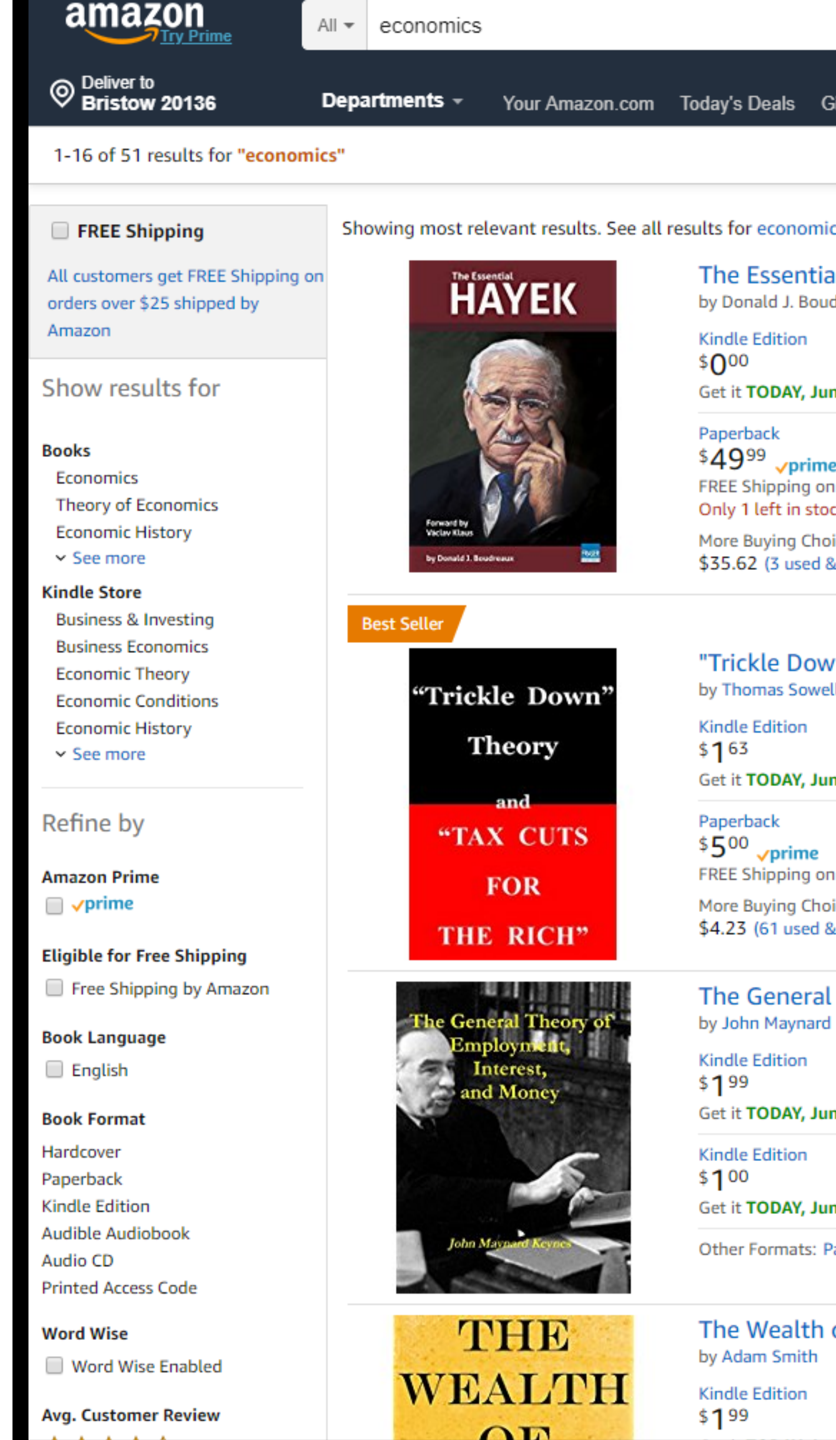
[Sandholm, chapter in Handbook of Market Design, 2013]

Bidding in government auction of airwaves reaches $34B
[NYTimes ‘14]

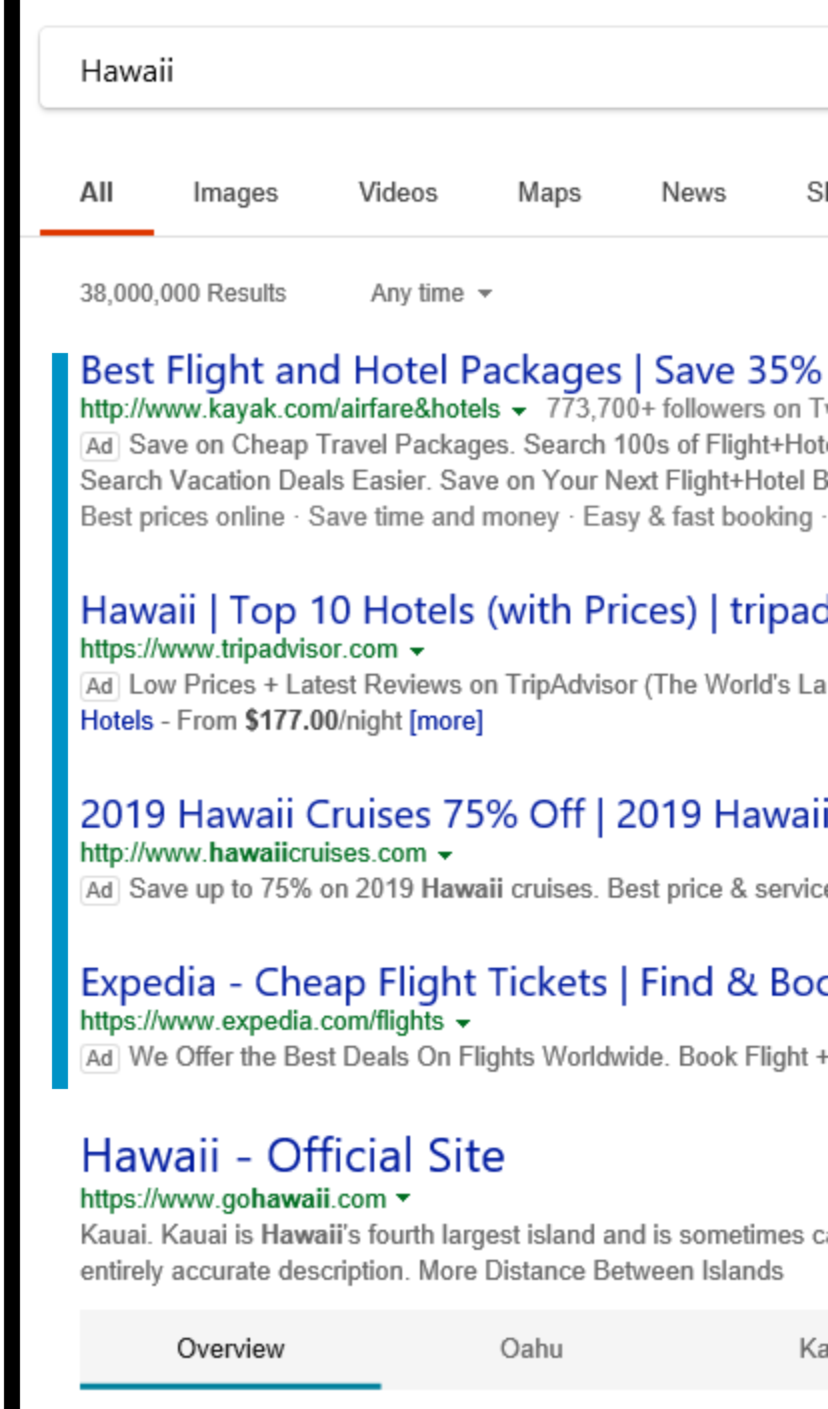


Amazon's profit swells to $1.6B

[NY Times '18]



	Ad rev. in 2016	Total rev. in 2016
Google	$79 B	$89.46 B
Facebook	$27 B	$27.64 B



Automated mechanism design

[Conitzer and Sandholm, UAI'02; Sandholm CP'03]

Use optimization, ML, & data to design mechanisms

– Helps overcome challenges faced by manual approaches:

2 items for sale: Revenue-maximizing mechanism unknown

Automated mechanism design

[Conitzer and Sandholm, UAI'02; Sandholm CP'03]

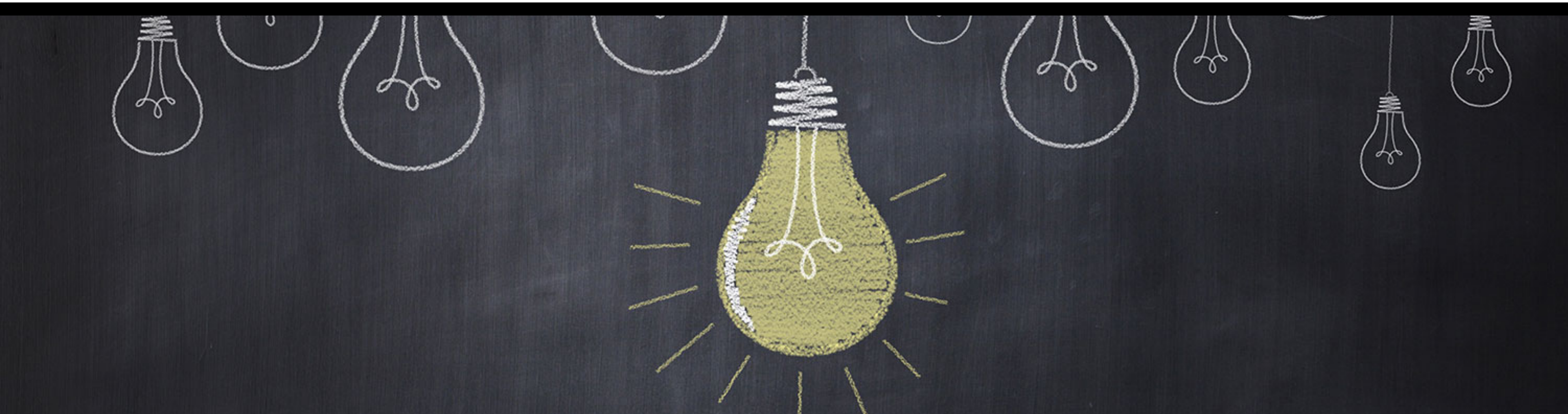
Use optimization, ML, & data to design mechanisms

- Helps overcome challenges faced by manual approaches:

  2 items for sale: Revenue-maximizing mechanism unknown

In this tutorial, we:

- Cover optimization algorithms
- Provide statistical guarantees
  - Techniques of independent interest (we believe) to ML theory



Outline

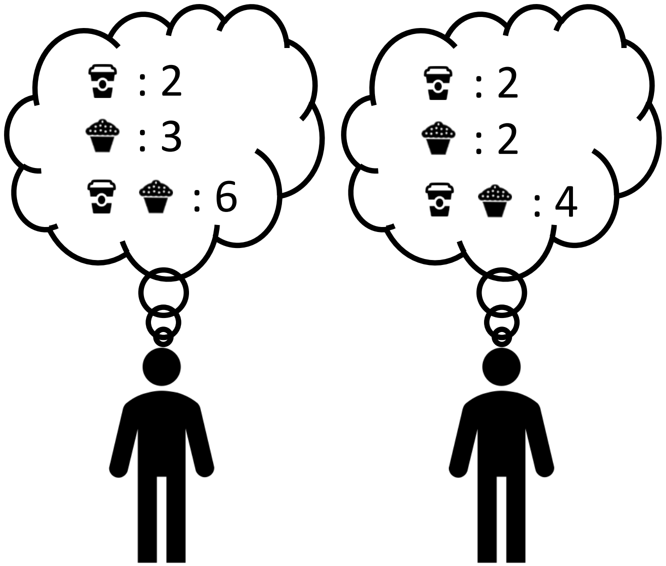
1. Mechanism design basics
2. Introduction to automated mechanism design (AMD)
3. Sample complexity guarantees for AMD
4. Additional AMD algorithms
5. Other learning models

Mechanism design for sales settings

There's a set of **items** for sale and a set of **buyers**

At a high level, a mechanism determines:

1. Which buyers receive which items
2. What they pay

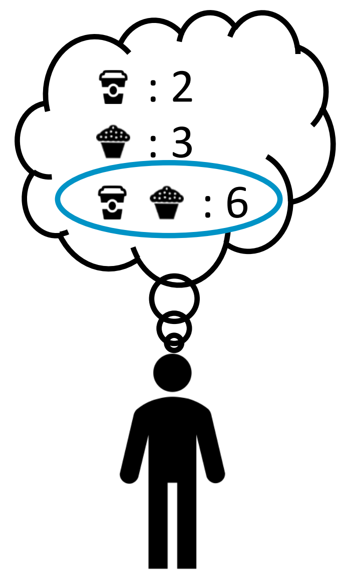


Mechanism design example: Posted price mechanisms

Set a price per item

Buyers buy the items maximizing their utility

Value for items minus price



Mechanism design example:
First-price auction

Highest bidder wins. Pays his bid.

Mechanism design example: Second-price auction

Highest bidder wins. Pays second highest bid.

Mechanism design example:
Second-price auction with a reserve

Auctioneer sets reserve price $\boldsymbol{r}$

Highest bidder wins if bid $\geq \boldsymbol{r}$

Pays maximum of second highest bid and $\boldsymbol{r}$

Reserve price: \$8 ⇨ Revenue = \$8

Reserve price: \$6 ⇨ Revenue = \$7

Second-price auction

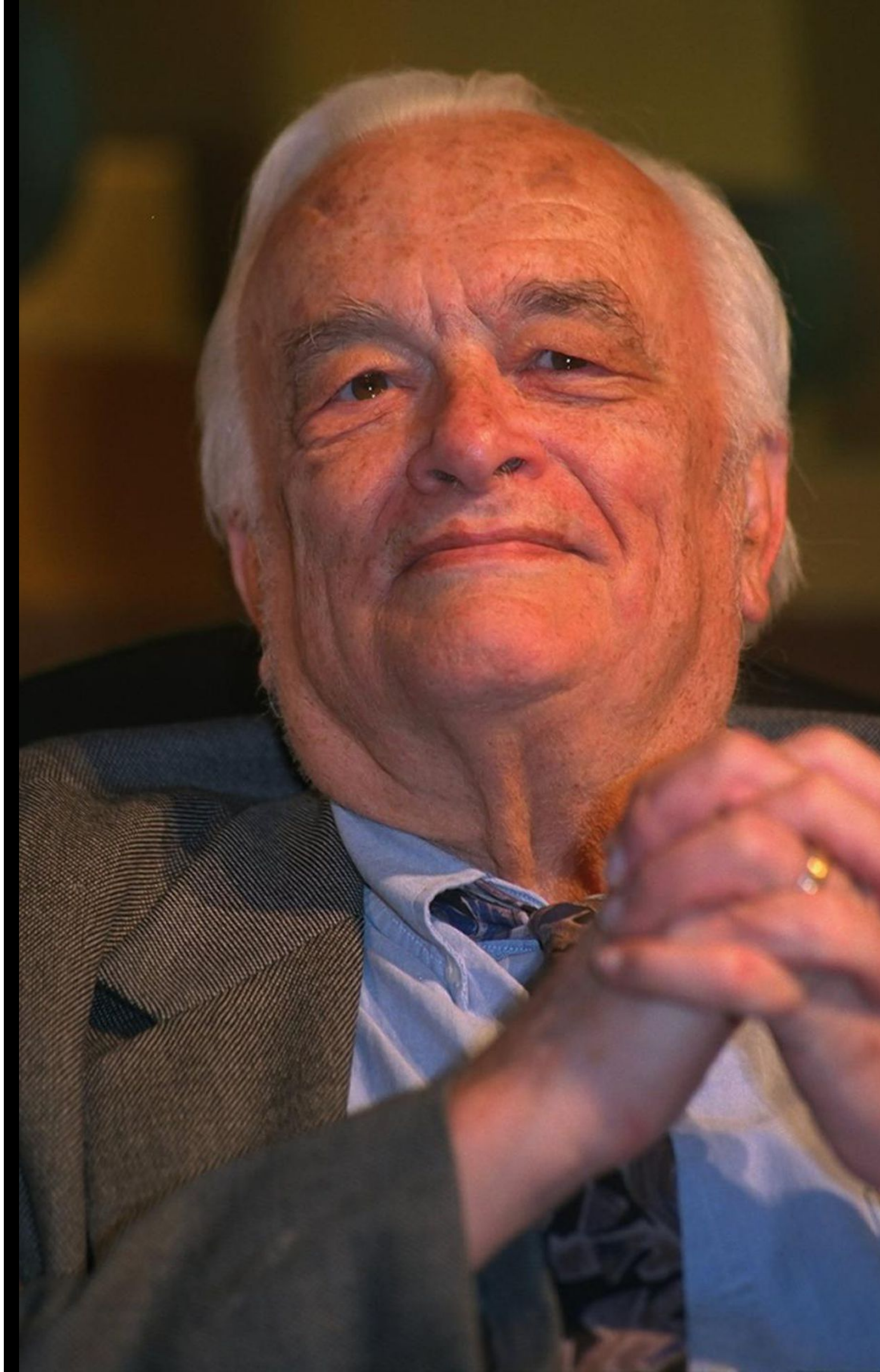
1961: Introduced by Vickrey

Vickrey, William. "Counterspeculation, auctions, and competitive sealed tenders." The Journal of finance 16.1 (1961): 8-37.

1996: He won Nobel Prize

Studied extensively in CS

[E.g., Sandholm, Intl. J. Electronic Commerce ’00; Cesa-Bianchi, Gentile, and Mansour, IEEE Transactions on Information Theory, ‘15; Daskalakis and Syrgkanis, FOCS’16].



Notation

There are m items and n buyers

Each buyer i has value $v_i(b) \in \mathbb{R}$ for each bundle $b \subseteq [m]$

Let $\boldsymbol{v}_i = \left(v_i(b_1), \ldots, v_i(b_{2^m})\right)$ for all $b_1, \ldots, b_{2^m} \subseteq [m]$

Buyer i's "type"

Notation

There are m items and n buyers

Each buyer i has value $v_i(b) \in \mathbb{R}$ for each bundle $b \subseteq [m]$

Let $\boldsymbol{v}_i = \left(v_i(b_1), \dots, v_i(b_{2^m})\right)$ for all $b_1, \dots, b_{2^m} \subseteq [m]$

Buyer i's "type"

Example

Items = { , }

$v_i(\emptyset) = 0$ $v_i(\ ) = 2$ $v_i(\ ) = 3$ $v_i(\ ,\ ) = 6$

$\boldsymbol{v}_i = (\ 0\ ,\ 2\ ,\ 3\ ,\ 6\ )$

What exactly is a **mechanism**?
(In sale settings)

Mechanism M is defined by an **allocation** and **payment** function.

1. **Allocation** function defines which buyers receive which items
2. **Payment** function defines how much each buyer pays

Revenue of M given values $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ is sum of payments:

$$\mathrm{revenue}_M(\boldsymbol{v}_1, \dots, \boldsymbol{v}_n)$$

Sometimes, each buyer i might need to submit a set of bids:

$$\widetilde{\boldsymbol{v}}_i = \left(\tilde{v}_i(b_1), \dots, \tilde{v}_i(b_{2^m})\right)$$

$\widetilde{\boldsymbol{v}}_i$ may not equal buyer i's true values $\boldsymbol{v}_i$

Mechanism desiderata

We want to design mechanisms that are:

Incentive compatible

Agents' bids equal their true values

They're incentivized to bid truthfully

Individually rational

Agents have nothing to lose by participating

Incentive compatibility

The second-price auction is **incentive compatible**.

*Every bidder will maximize their **utility** by bidding truthfully.*

$$\boxed{\left(\textbf{value}(\;) - \textbf{payment}\right) \bullet \mathbf{1}\left(\textbf{wins item}\right)}$$

Why not bid **above** value( )?

Incentive compatibility

The second-price auction is **incentive compatible**.

*Every bidder will maximize their **utility** by bidding truthfully.*

(value() - payment) • 1(wins item)

Why not bid **above** value() ?

– If winner, will stay winner and price won't change

Incentive compatibility

The second-price auction is **incentive compatible**.

*Every bidder will maximize their **utility** by bidding truthfully.*

(value( ) - payment) • 1 (wins item)

Why not bid **above** value( ) ?

– If winner, will stay winner and price won't change

– If loser, might become winner, but will pay more than value( )

Incentive compatibility

The second-price auction is **incentive compatible**.

*Every bidder will maximize their **utility** by bidding truthfully.*

(value( ) - payment) • 1 (wins item)

Why not bid **below** value( )?

Incentive compatibility

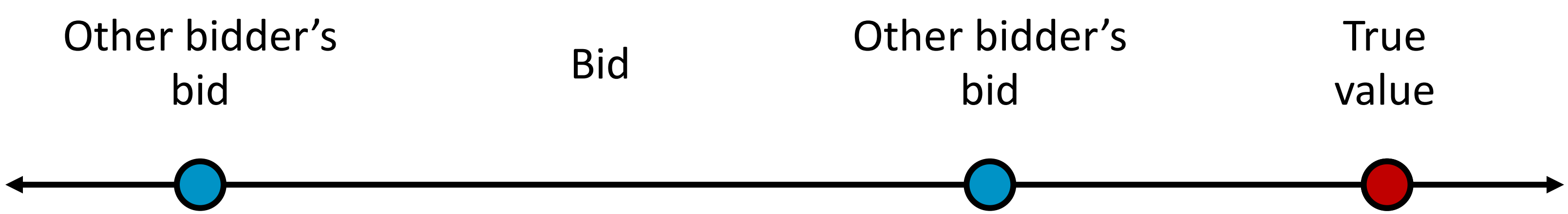
The second-price auction is **incentive compatible**.

Every bidder will maximize their ***utility*** *by bidding truthfully.*

$$\boxed{\left(\text{value}(\;) - \text{payment}\right) \bullet 1\left(\text{wins item}\right)}$$

Why not bid **below** value( )?

– If winner, might become loser; shift from non-negative to zero utility



Incentive compatibility

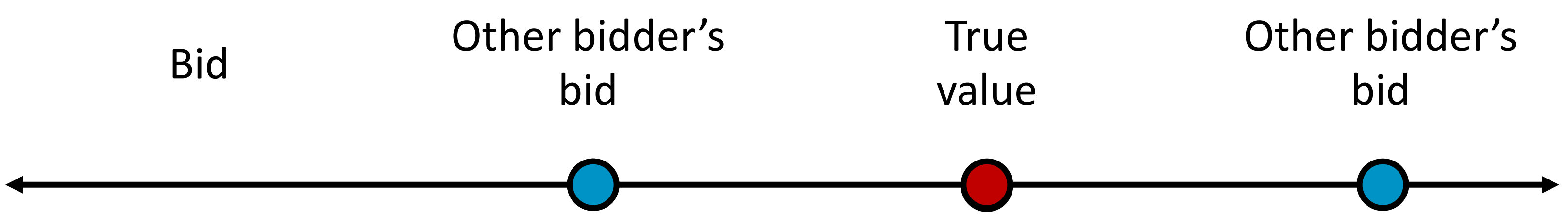
The second-price auction is **incentive compatible**.

*Every bidder will maximize their **utility** by bidding truthfully.*

$\big(\textbf{value}(\;) - \textbf{payment}\big) \bullet \mathbf{1}\big(\textbf{wins item}\big)$

Why not bid **below** value( )?

- If winner, might become loser; shift from non-negative to zero utility
- If loser, will still be loser, so utility will still be zero



Individual rationality

The second-price auction is individually rational.

Each bidder is no worse off participating than not, when truthful

Bidders pay nothing or their payment is smaller than their value.

A bit more formally…

Standard assumption

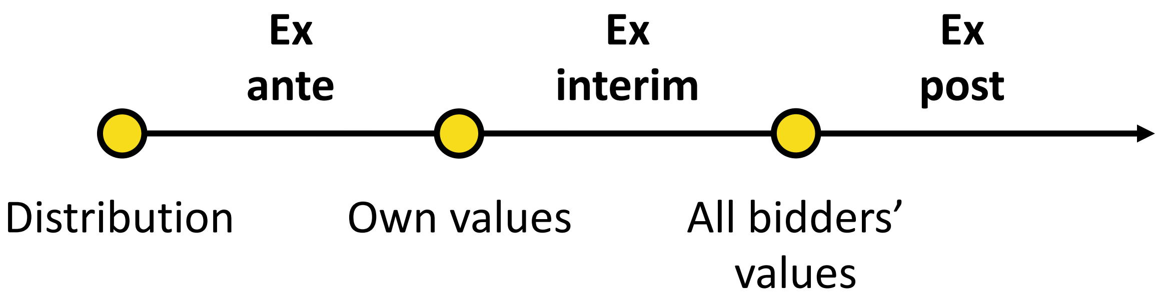
Buyers' values are drawn from a probability **distribution**.

Example

$(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) \sim \mathcal{D}$, where $\boldsymbol{v}_i = \big[\, v_i(\emptyset),\ v_i($  $),\ v_i($  $),\ v_i($  ,  $)\,\big]$

Different types of incentive compatibility

Each bidder knows:

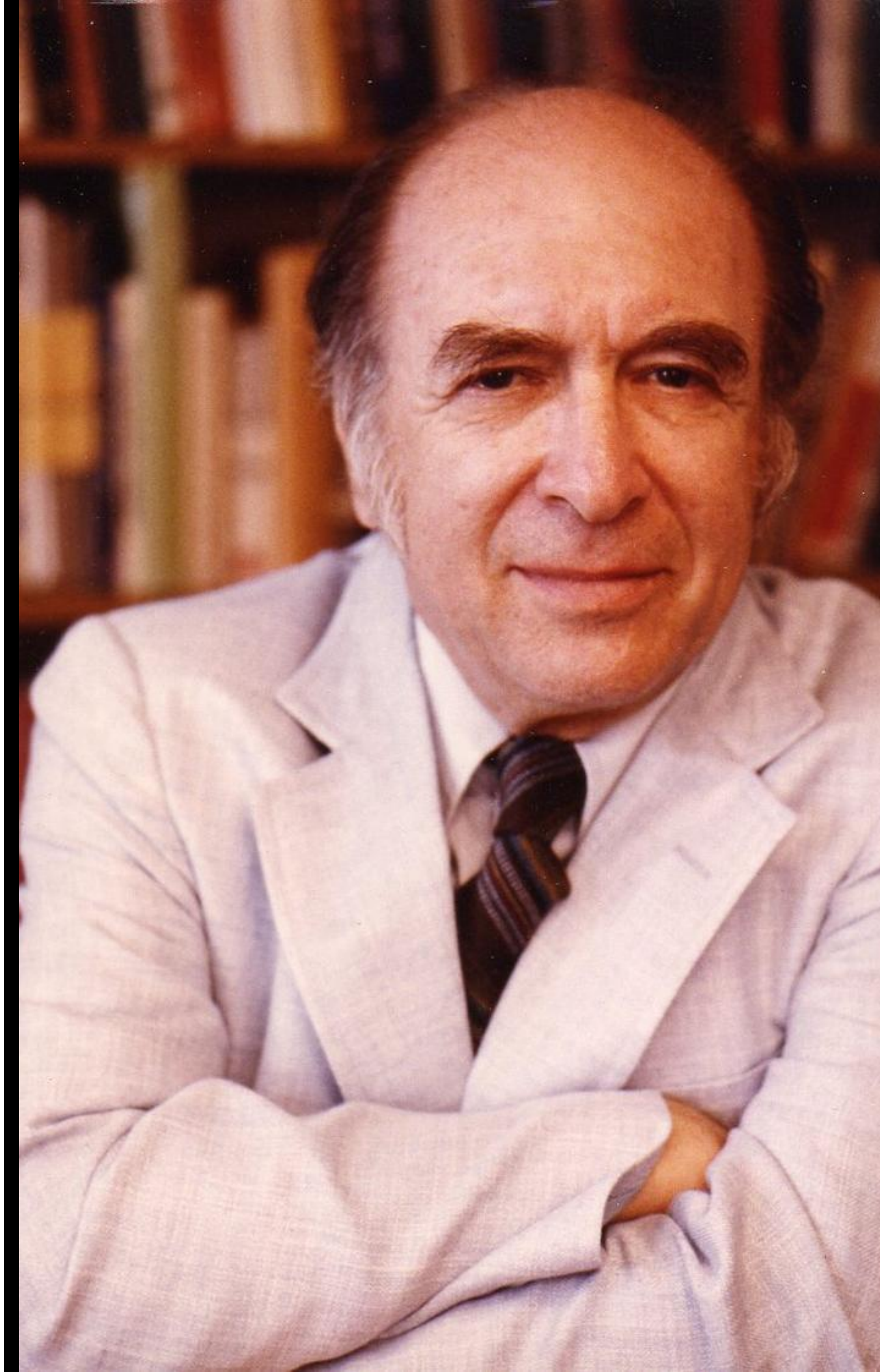


Incentive compatibility

1972: Hurwicz introduced IC

2007: He won Nobel prize

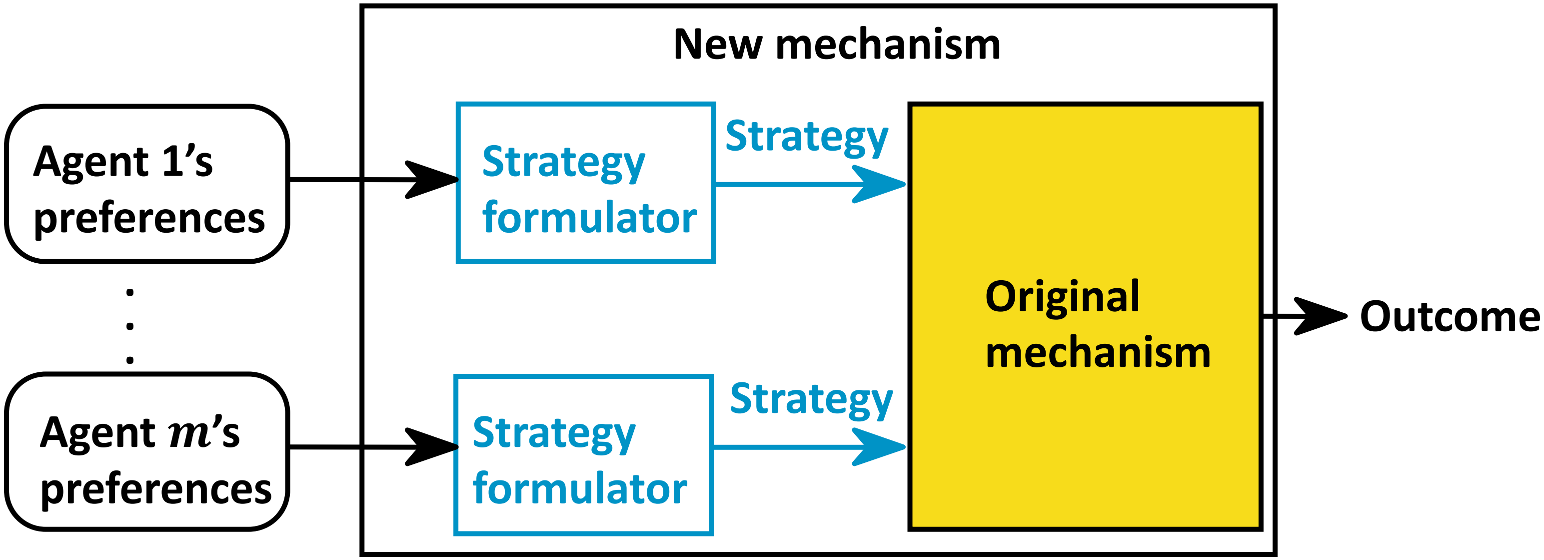
L. Hurwicz. On Informationally Decentralized Systems. Decision and Organization, edited by C.B. McGuire and R. Radner. 1972.



Revelation principle (informal)

If agents always incentivized to follow fixed strategy…

there's an IC mechanism w/ same payments and allocations



Mechanism lies for the agents!

Optimal single-item sales mechanism

1981: Myerson discovered "optimal" 1-item auction

$\boxed{\text{Revenue-maximizing}}$

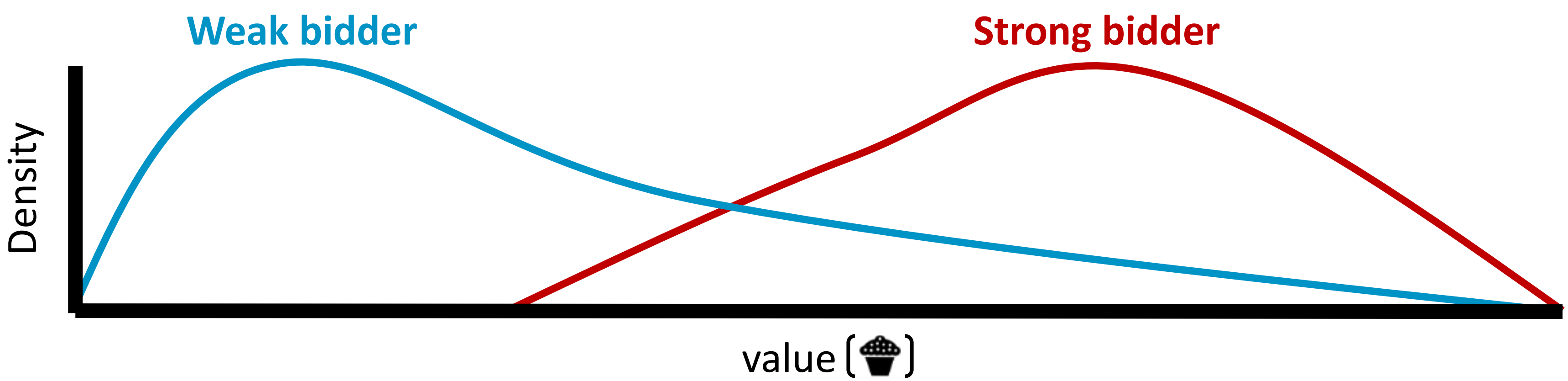
2007: Won Nobel prize

R. Myerson. Optimal auction design. Mathematics of Operations Research, 6(1):58–73, 1981.

Optimal single-item auctions

What's the problem with second-price auction?

- Strong bidder typically wins and pays weak bidder's bid
- Leaves revenue on the table!



Myerson's optimal auction boosts weak bidders' bids

Creates extra competition while maintaining IC

Optimal single-item auctions

Bidder i's value distribution has PDF f_i, CDF F_i, support in [0, 1]

Myerson's optimal auction

Let $\phi_i(t) = t - \frac{1-F_i(t)}{f_i(t)}$. Solicit bids $\tilde{v}_1, \dots, \tilde{v}_n$ from buyers

If all *virtual values* $\phi_1(\tilde{v}_1), \dots, \phi_n(\tilde{v}_n) < 0$, don't allocate item

Else allocate item to buyer i^* with highest virtual value $\phi_i(\tilde{v}_i)$

Charge bidder i^* her ***threshold bid*** (min she could bid and win):

$$\phi_{i^*}^{-1}\left(\max\left(0, \{\phi_{i^*}(\tilde{v}_j)\}_{j \neq i^*}\right)\right)$$

Optimal single-item auctions

When buyers' values are i.i.d.:

Equivalent to 2nd-price auction with reserve of $\phi_i^{-1}(0)$

Extended to selling multiple units of an item [Maskin & Riley, '89]

Major challenge:
Optimal multi-item auctions

Don’t know how to sell two items optimally! Tons of work, e.g.:

Economics

E.g., Rochet, Journal of Mathematical Economics, ‘87; Avery and Hendershott, Review of Economic Studies, ’00; Armstrong, Review of Economic Studies, ’00; Thanassoulis, Journal of Economic Theory, ’04; Manelli and Vincent, Journal of Economic Theory ’06

Computer science

E.g., Conitzer and Sandholm, UAI’02, ICEC’03, EC’04; Likhodedov and Sandholm, AAAI‘04, AAAI’05; Cai and Daskalakis, FOCS’11; Cai, Daskalakis, and Weinberg, STOC’12, FOCS’12; Sandholm and Likhodedov, Operations Research ’15; Yao, SODA’15; Hart and Nisan, Journal of Economic Theory, ‘17

Outline

1. Mechanism design basics
2. Introduction to automated mechanism design (AMD)
3. Sample complexity guarantees for AMD
4. Additional AMD algorithms
5. Other learning models

Automated mechanism design (AMD) [Conitzer and Sandholm, UAI'02; Sandholm CP'03]

Solve mechanism design as a search/optimization problem automatically

- Built a system for doing that
- Create a mechanism for the **specific setting at hand** rather than a class of settings

---

- Can lead to greater value of designer's objective than known mechanisms
- Sometimes circumvents economic impossibility results
  - Always minimizes the pain implied by them
- Can be used in new settings & for unusual objectives
- Can yield stronger incentive compatibility & participation properties
- Shifts the burden of design from human to machine

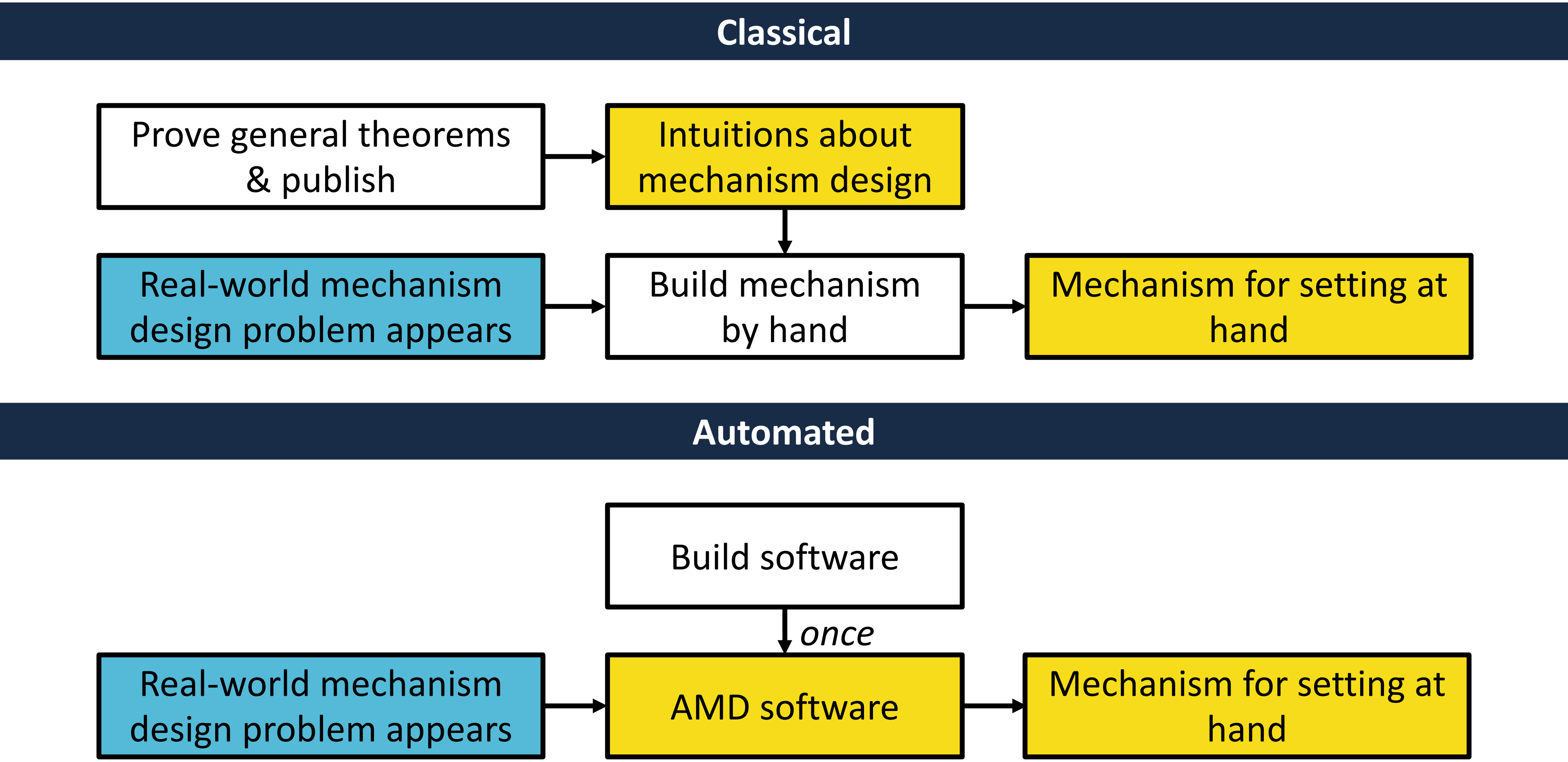
- Often designer has info about agents – silly to ignore

Automated mechanism design ≠ Algorithmic mechanism design

[Conitzer and Sandholm, UAI-02]

[Nisan and Ronen`01]

Classical vs. automated mechanism design

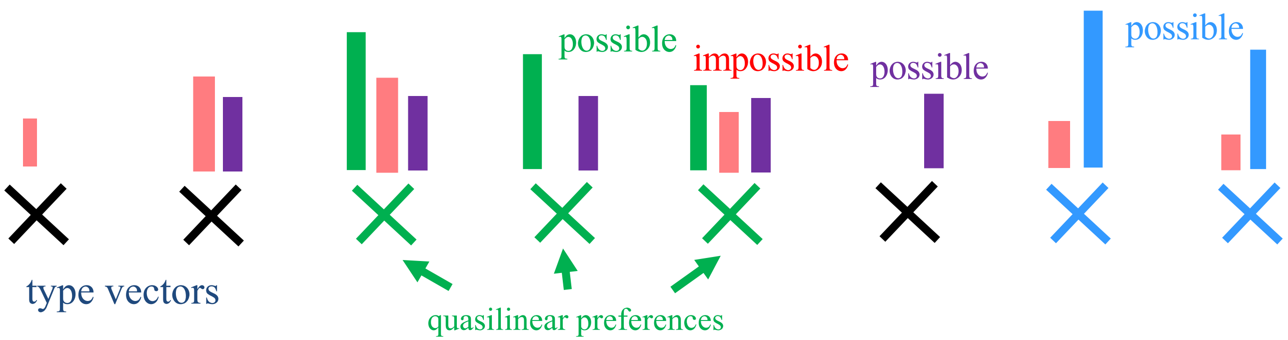


Input

- Instance is given by
  - Set of possible outcomes
  - Set of agents
    - For each agent
      - set of possible types
      - probability distribution over these types
      - utility function converting type/outcome pairs to utilities
  - Objective function
    - Gives a value for each outcome for each combination of agents' types
    - E.g. payment maximization
  - Restrictions on the mechanism
    - Are side payments allowed?
    - Is randomization over outcomes allowed?
    - What concept of nonmanipulability is used?
    - What participation constraint notion (if any) is used?

Output

- Mechanism
  - A mechanism maps combinations of agents' revealed types to outcomes
    - Randomized mechanism maps to probability distributions over outcomes
    - Also specifies payments by agents (if payments allowed)
- …which
  - is nonmanipulable (according to the given concept)
  - satisfies the given participation constraint
  - maximizes the expectation of the objective function


possible
impossible
possible
possible
type vectors
quasilinear preferences

Complexity of AMD

Theorem [Conitzer and Sandhom, UAI'02, ICEC'03, EC'04]

The following are NP-complete (even for 1 buyer) for designing a deterministic mechanism:

1. Maximizing social welfare (sum of agents' values for their allocations) (no payments)
2. Maximizing designer's utility over outcomes (no payments)
3. Maximizing a general (linear) objective that doesn't regard payments
4. Expected revenue

Polynomial time for designing a randomized mechanism for constant #agents (LP)

But also there is a blowup in *input*

- **Exponential** allocation space: $(\#agents + 1)^{\#items}$
- The support of the distribution over values might be **doubly exponential**: $k\text{^}(2^{\#items})$
  - k is the number of possible values a buyer might have for a bundle

Two key ideas to get scalability and avoid the need to discretize type space

[Likhodedov & Sandholm AAAI-04, AAAI-05]

- Don't assume valuation distribution is given, only samples from it

- AMD as search in a parametric mechanism class

There's an **unknown** distribution over valuations.

Use a set of samples to **learn** a mechanism that has high expected revenue.

Multi-item

E.g., Likhodedov and Sandholm, AAAI'04, AAAI'05; Morgenstern and Roughgarden, COLT'16; Syrgkanis, NIPS'17; Cai and Daskalakis, FOCS'17; Gonczarowski and Weinberg, FOCS'18

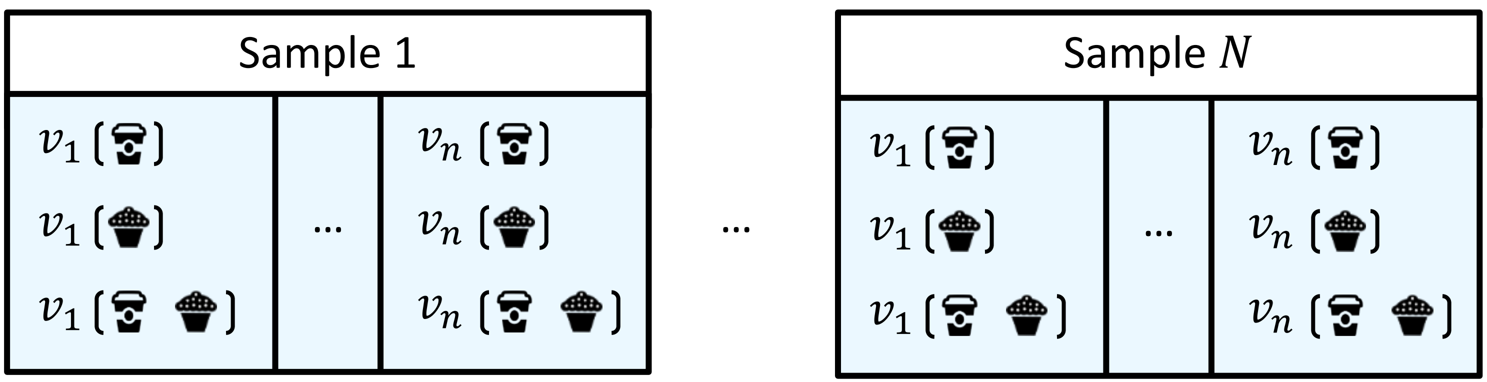
Single-item

E.g., Balcan, Blum, Hartline, and Mansour, FOCS'05; Elkind, SODA'07; Dhangwatnotai, Roughgarden, and Yan, EC'10; Mohri and Medina, ICML'14; Cole and Roughgarden STOC'14

Mechanism design as a learning problem

Goal: Given large family of mechanisms and set of buyers' values sampled from unknown distribution $\mathcal{D}$, find mechanism with high expected revenue.

Approach: Find mechanism that's (nearly) optimal over the set of samples.

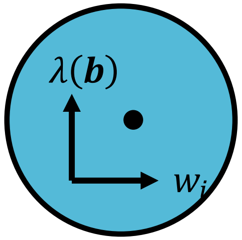


Two key ideas to get scalability and avoid the need to discretize type space

[Likhodedov & Sandholm AAAI-04, AAAI-05]

1. Don't assume valuation distribution is given, only samples from it

2. AMD as search in a parametric mechanism class



Vickrey-Clarke-Groves mechanism (VCG)

A **fundamental building block** for multi-item, multi-bidder automated mechanism design of deterministic mechanisms

Based on a series of papers by Vickrey [Journal of Finance '61], Clarke [Public Choice '71], and Groves [Econometrica '73]

The multi-item, multi-bidder incentive compatible auction that maximizes **social welfare**

Sum of the buyers' values for the items they're allocated

Generalization of the Vickrey auction

Vickrey-Clarke-Groves mechanism (VCG)

Each buyer i submits a bid $v_i(b)$ for each bundle b of items.

The auction is **incentive compatible**, so we assume the bidders' bids equal their true values [Clarke, Public Choice '71; Groves, Econometrica '73; Vickrey, Journal of Finance '61]

Vickrey-Clarke-Groves mechanism (VCG)

Let $(b_1, \dots, b_n)$ be an allocation of the m goods.

This means $b_1, \dots, b_n \subseteq [m]$ *and* $b_i \cap b_j = \emptyset$.

$SW(b_1, \dots, b_n) = \sum_{i \in Bidders} v_i(b_i)$

$\boldsymbol{b}^* = (b_1^*, \dots, b_n^*)$ maximizes social welfare $SW(\cdot)$

$SW_{-i}(b_1, \dots, b_n) = \sum_{j \in Bidders - \{i\}} v_j(b_j)$

Social welfare of the allocation, not including bidder i's value

Vickrey-Clarke-Groves mechanism (VCG)

Let $(b_1, \dots, b_n)$ be an allocation of the m goods.

This means $b_1, \dots, b_n \subseteq [m]$ *and* $b_i \cap b_j = \emptyset$.

$SW(b_1, \dots, b_n) = \sum_{i \in Bidders} v_i(b_i)$

$\boldsymbol{b}^* = (b_1^*, \dots, b_n^*)$ maximizes social welfare $SW(\cdot)$

$SW_{-i}(b_1, \dots, b_n) = \sum_{j \in Bidders - \{i\}} v_j(b_j)$

$\boldsymbol{b}^{-i} = \left(b_1^{-i}, \dots, b_n^{-i}\right)$ maximizes $SW_{-i}(b_1, \dots, b_n)$

The social-welfare-maximizing allocation if bidder i hadn't participated.

Bidder	1	2
	1	0
	2	1
 	2.5	1

$(b_1^*, b_2^*) =$ (( ) ($\emptyset$))

$(b_1^{-1}, b_2^{-1}) =$ (($\emptyset$) ( ))

Vickrey-Clarke-Groves mechanism (VCG)

Let $(b_1, \ldots, b_n)$ be an allocation of the m goods.

This means $b_1, \ldots, b_n \subseteq [m]$ *and* $b_i \cap b_j = \emptyset$.

$SW(b_1, \ldots, b_n) = \sum_{i \in Bidders} v_i(b_i)$

$\boldsymbol{b}^* = (b_1^*, \ldots, b_n^*)$ maximizes social welfare $SW(\cdot)$

$SW_{-i}(b_1, \ldots, b_n) = \sum_{j \in Bidders - \{i\}} v_j(b_j)$

$\boldsymbol{b}^{-i} = \left(b_1^{-i}, \ldots, b_n^{-i}\right)$ maximizes $SW_{-i}(b_1, \ldots, b_n)$

Allocation: $\boldsymbol{b}^*$

The social-welfare-maximizing allocation.

Bidder i pays $SW_{-i}\left(\boldsymbol{b}^{-i}\right) - SW_{-i}(\boldsymbol{b}^*)$

Bidder	1	2
	1	0
	2	1
	2.5	1

$(b_1^*, b_2^*) = $ (() ($\emptyset$))

$(b_1^{-1}, b_2^{-1}) = $ (($\emptyset$) ())

Vickrey-Clarke-Groves mechanism (VCG)

Let $(b_1, \dots, b_n)$ be an allocation of the m goods.

This means $b_1, \dots, b_n \subseteq [m]$ *and* $b_i \cap b_j = \emptyset$.

$SW(b_1, \dots, b_n) = \sum_{i \in Bidders} v_i(b_i)$

$\boldsymbol{b}^* = (b_1^*, \dots, b_n^*)$ maximizes social welfare $SW(\cdot)$

$SW_{-i}(b_1, \dots, b_n) = \sum_{j \in Bidders - \{i\}} v_j(b_j)$

$\boldsymbol{b}^{-i} = \left(b_1^{-i}, \dots, b_n^{-i}\right)$ maximizes $SW_{-i}(b_1, \dots, b_n)$

Allocation: $\boldsymbol{b}^*$

How much happier everyone would be if buyer i hadn't participated.

Payment:

Bidder i pays $SW_{-i}\left(\boldsymbol{b}^{-i}\right) - SW_{-i}(\boldsymbol{b}^*)$

Bidder	1	2
	1	0
	2	1
	2.5	1

$(b_1^*, b_2^*) = ((\quad) \; (\emptyset))$

$(b_1^{-1}, b_2^{-1}) = ((\emptyset) \; (\quad))$

Vickrey-Clarke-Groves mechanism (VCG)

What if we add an **additive boost** to the social welfare of the allocation (b_1^{-1}, b_2^{-1})?

Allocation: $\boldsymbol{b}^*$

How much happier everyone would be if buyer i hadn't participated.

Payment:

Bidder i pays $SW_{-i}(\boldsymbol{b}^{-i}) - SW_{-i}(\boldsymbol{b}^*)$

Bidder	1	2
	1	0
	2	1
 	2.5	1

$(b_1^*, b_2^*) =$ [( ) ($\emptyset$)]

$(b_1^{-1}, b_2^{-1}) =$ [($\emptyset$) ( )]

$SW_{-1}(b_1^{-1}, b_2^{-1}) = 1$

$SW_{-1}(b_1^*, b_2^*) = 0$

Bidder 1 pays

$1 - 0 = 1$

Bidder 1 values her allocation for $2.5, but only payed $1. **How can we get her to pay more?**

Vickrey-Clarke-Groves mechanism (VCG)

What if we add an **additive boost** to the social welfare of the allocation (b_1^{-1}, b_2^{-1})?

Allocation: $\boldsymbol{b}^*$

Payment:

Bidder i pays $SW_{-i}(\boldsymbol{b}^{-i}) - SW_{-i}(\boldsymbol{b}^*)$

Bidder	1	2
	1	0
	2	1
	2.5	1

$(b_1^*, b_2^*) = [\,(\;\;)\;(\emptyset)\,]$

$(b_1^{-1}, b_2^{-1}) = [\,(\emptyset)\;(\;\;)\,]$

$SW_{-1}(b_1^{-1}, b_2^{-1}) = 1 + \mathbf{1.49}$

$SW_{-1}(b_1^*, b_2^*) = 0$

Bidder 1 pays

$1 + \mathbf{1.49} - 0 = \mathbf{2.49}$

Affine maximizer auctions

Affine maximizer auction [Roberts 1979]

1. Compute the social-welfare-maximizing allocation:

$$\boldsymbol{b}^* = (b_1^*, \dots, b_n^*) = \operatorname{argmax}\left\{\sum_{j\in\text{Bidders}} \boldsymbol{w_j}\, v_j(b_j) + \boldsymbol{\lambda}(b_1, \dots, b_n)\right\}$$

2. For each bidder i, find social-welfare-maximizing allocation w/o his participation:

$$\boldsymbol{b}^{-i} = (b_1^{-i}, \dots, b_n^{-i}) = \operatorname{argmax}\left\{\sum_{j\in\text{Bidders}-\{i\}} \boldsymbol{w_j}\, v_j(b_j) + \boldsymbol{\lambda}(b_1, \dots, b_n)\right\}$$

3. Compute bidder i's payment, for all i

(*How much happier everyone would be if bidder i hadn't participated*):

$$\left[\left(\sum_{j\in\text{Bidders}-\{i\}} \boldsymbol{w_j}\, v_j(b_j^{-i}) + \boldsymbol{\lambda}(\boldsymbol{b}^{-i})\right) - \left(\sum_{j\in\text{Bidders}-\{i\}} \boldsymbol{w_j}\, v_j(b_j^*) + \boldsymbol{\lambda}(\boldsymbol{b}^*)\right)\right]$$

- AMAs are ex-post IC and IR [Roberts 1979]
- Every IC multi-item, multi-bidder auction (where each bidder only cares about what she gets and pays) is almost an affine maximizer auction (with some qualifications) [Lavi, Mu'Alem, and Nisan, FOCS'03].

Virtual valuation combinatorial auctions (VVCAs)

Boost per bidder-bundle pair (j, b): $\boldsymbol{\lambda_j}(b)$; Weight per bidder i: $\boldsymbol{w_i}$

$\boldsymbol{\lambda}(b_1, \dots, b_n)$ replaced with $\sum_{j \in \text{Bidders}} \boldsymbol{\lambda_j}(b_j)$

Virtual valuation combinatorial auctions [Likhodedov and Sandholm, AAAI'04, '05; OR'15]

1. Compute the social-welfare-maximizing allocation:

$$\boldsymbol{b}^* = (b_1^*, \dots, b_n^*) = \operatorname{argmax}\{\textstyle\sum_{j \in \text{Bidders}}[\, \boldsymbol{w_j}\, v_j(b_j) + \boldsymbol{\lambda_j}(b_j)]\}$$

2. For each bidder i, compute the social-welfare-maximizing allocation without his participation:

$$\boldsymbol{b}^{-i} = (b_1^{-i}, \dots, b_n^{-i}) = \operatorname{argmax}\{\textstyle\sum_{j \in \text{Bidders}-\{i\}}[\, \boldsymbol{w_j}\, v_j(b_j) + \boldsymbol{\lambda_j}(b_j)]\}$$

3. Compute bidder i's payment, for all i

(*How much happier everyone would be if bidder i hadn't participated*):

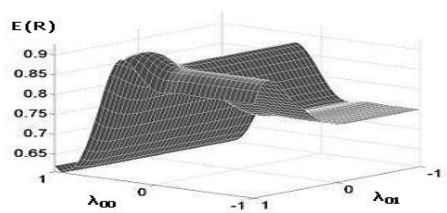
$$\frac{\mathbf{1}}{\boldsymbol{w_i}}\left[\textstyle\sum_{j \in \text{Bidders}-\{i\}}[\boldsymbol{w_j}\, v_j(b_j^{-i}) + \boldsymbol{\lambda}_j(b_j^{-i})] - \sum_{j \in \text{Bidders}-\{i\}}[\boldsymbol{w_j}\, v_j(b_j^{*}) + \boldsymbol{\lambda}_j(b_j^{*})]\right]$$

Computational considerations

[Sandholm & Likhodedov, OR'15]

Fact

Expected revenue is not convex in the VVCA or AMA parameters.



Projection of expected revenue surface on a 3D subspace

Theorem

There is no polynomial-time algorithm capable of determining (for every given set of valuations) whether one parameter vector is better than another (unless *P=NP*).

Theorem

For any given valuation vector, revenue has only one maximum in any parameter.

Theorem

Expected revenue is continuous and almost everywhere differentiable in parameters.

Algorithm possibilities:

1. Grid search
2. Hill climbing in parameter space – starting, e.g., from VCG

(In either method, evaluate each step using simulation.)

Simple search algorithms in parameter space
[Sandholm and Likhodedov, OR'15]

Algorithm AMA*

Iterated grid search of AMA parameter space, with grid tightened and re-centered around best solution from previous iteration.

Algorithm VVCA*

Ditto for VVCA parameter space.

- Grid search not scalable to large problems
- Overfitting already on 3rd iteration (when using 1,000 samples in the training set)
  => practical motivation for our learning theory

Algorithm BLAMA (Basic Local AMA search)

1. Start at VCG ($w_i = 1$ for every bidder i and $\lambda(b_1, \ldots, b_n) = 0$ for all allocations $(b_1, \ldots, b_n)$).
2. Run (Fletcher-Reeves conjugate) gradient ascent in AMA parameter space.

Reduce complexity by selecting gradient ascent direction using economic insights
[Sandholm and Likhodedov, OR'15]

High-level idea: If bidder i pays in allocation $\boldsymbol{b}^* = (b_1^*, \ldots, b_n^*)$ much less than her value for b_i^*, she should pay more.

Allocation boosting of AMA (ABAMA) [Sandholm and Likhodedov, OR'15]

1. Sample the valuations from the prior distributions
2. Start at VCG
3. For every sample point, compute the *revenue loss* on the winning allocation (ABAMAa) or the second-best allocation (ABAMAb)
   - The revenue loss from a bidder is the difference between the bidder's valuation and her payment
   - The revenue loss is the sum of the bidders' revenue losses
   - The revenue loss of an allocation is the sum of the revenue losses of the samples associated with the allocation
4. Make a list of allocations in decreasing order of revenue loss
5. Choose the first allocation, a, from the list. If the list is empty, exit.
6. Run (Fletcher-Reeves conjugate) gradient ascent in the {w, λ(a)} subspace of the AMA parameter space.
   – If the values of {w, λ(a)} did not change (i.e., we cannot further improve revenue by modifying {w, λ(a)}), remove a from the list and go to 5.
   – Otherwise go to 3.

Bidder-Bundle Boosting VVCA (BBBVVCA) algorithm [Sandholm and Likhodedov, OR'15]

- Similar idea, but optimized for VVCAs

Experiments: 2 items, 2 bidders

Experimental setup	
	• $v_i(\{1\})$ and $v_i(\{2\})$ are drawn from a prior distribution with PDF f_i • $v_i(\{1,2\}) = v_i(\{1\}) + v_i(\{2\}) + c_i$ • Each c_i is drawn from a distribution with PDF f_c

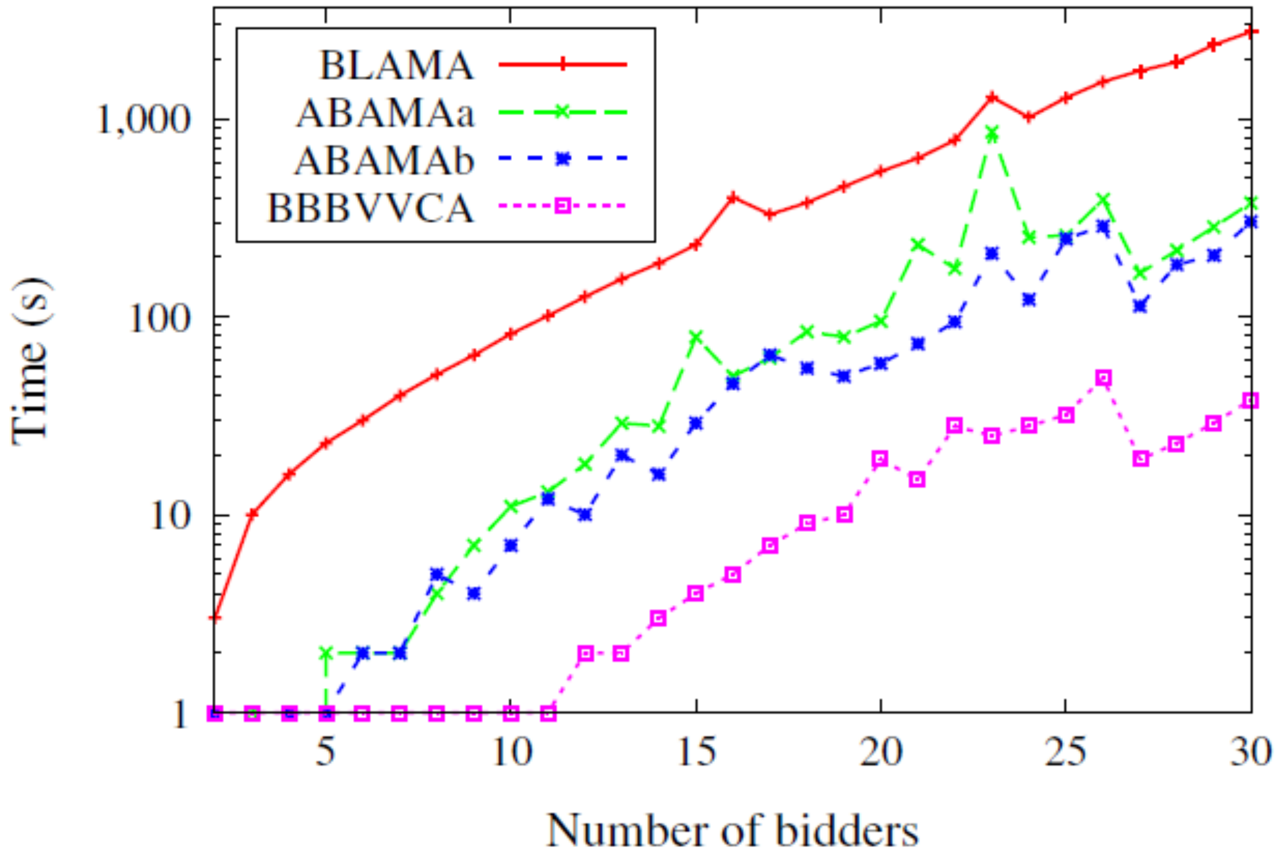
	Setting 1	**Setting 2**	**Setting 3**
f_1	U[0, 1]	U[1, 2]	U[1, 2]
f_2	U[0, 1]	U[1, 2]	U[1, 5]
f_c	0	U[-1, 1]	U[-1, 1]
VCG	2/3	2.45	2.85
AMA*	+32%	+14%	+48%
VVCA*	+31%	+13%	+47%
BLAMA	+17%	+13%	+31%
ABAMA	+17%	+13%	+32%
BBBVVCA	+18%	+14%	+30%

Table shows the revenue lift of various mechanisms over VCG

[Sandholm and Likhodedov, OR'15]

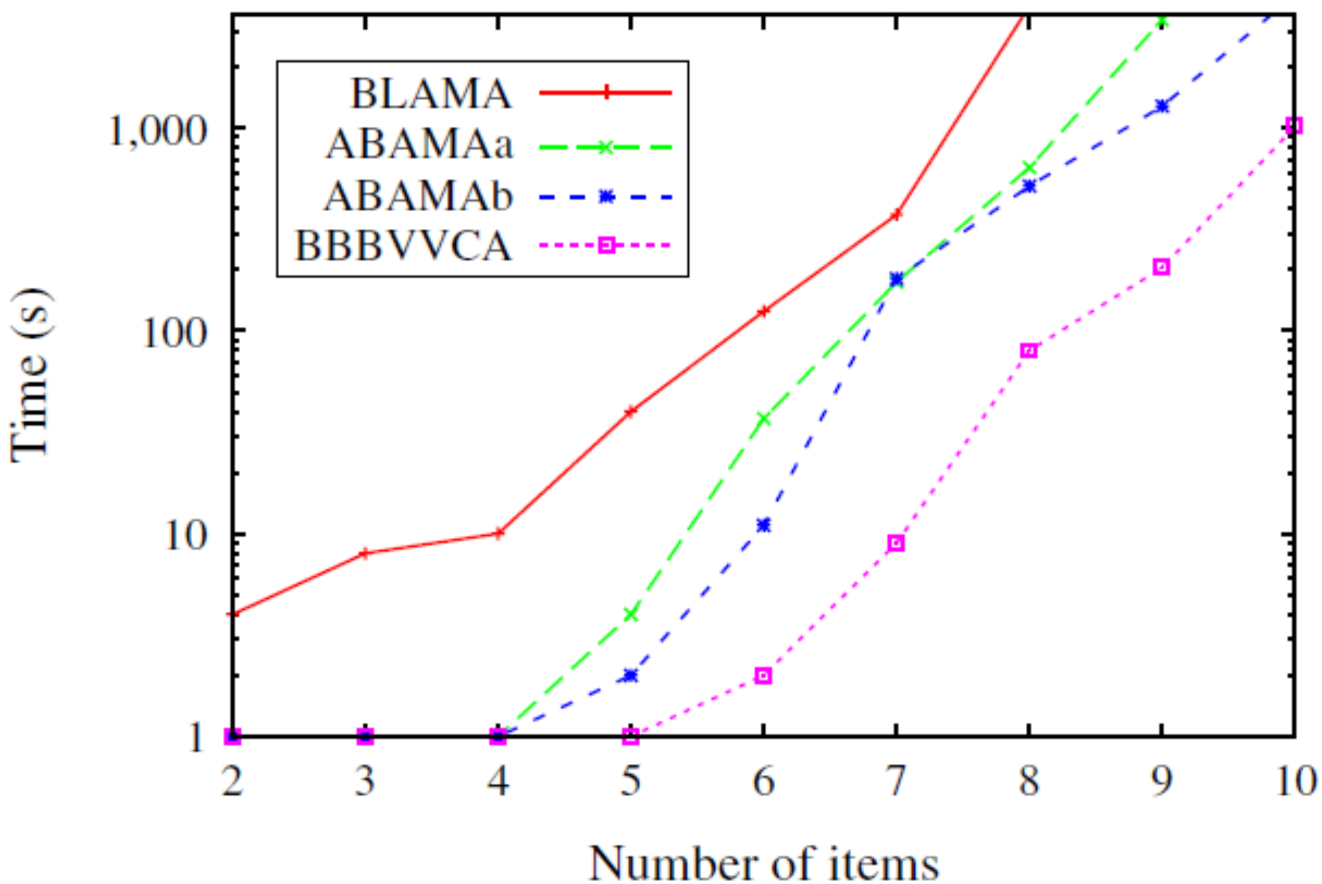
In Setting I, generalizing the mechanism design from MBARPs to VVCAs doesn't yield additional revenue, but generalizing further to AMAs does.

Scalability experiments (3 items, symmetric distribution)



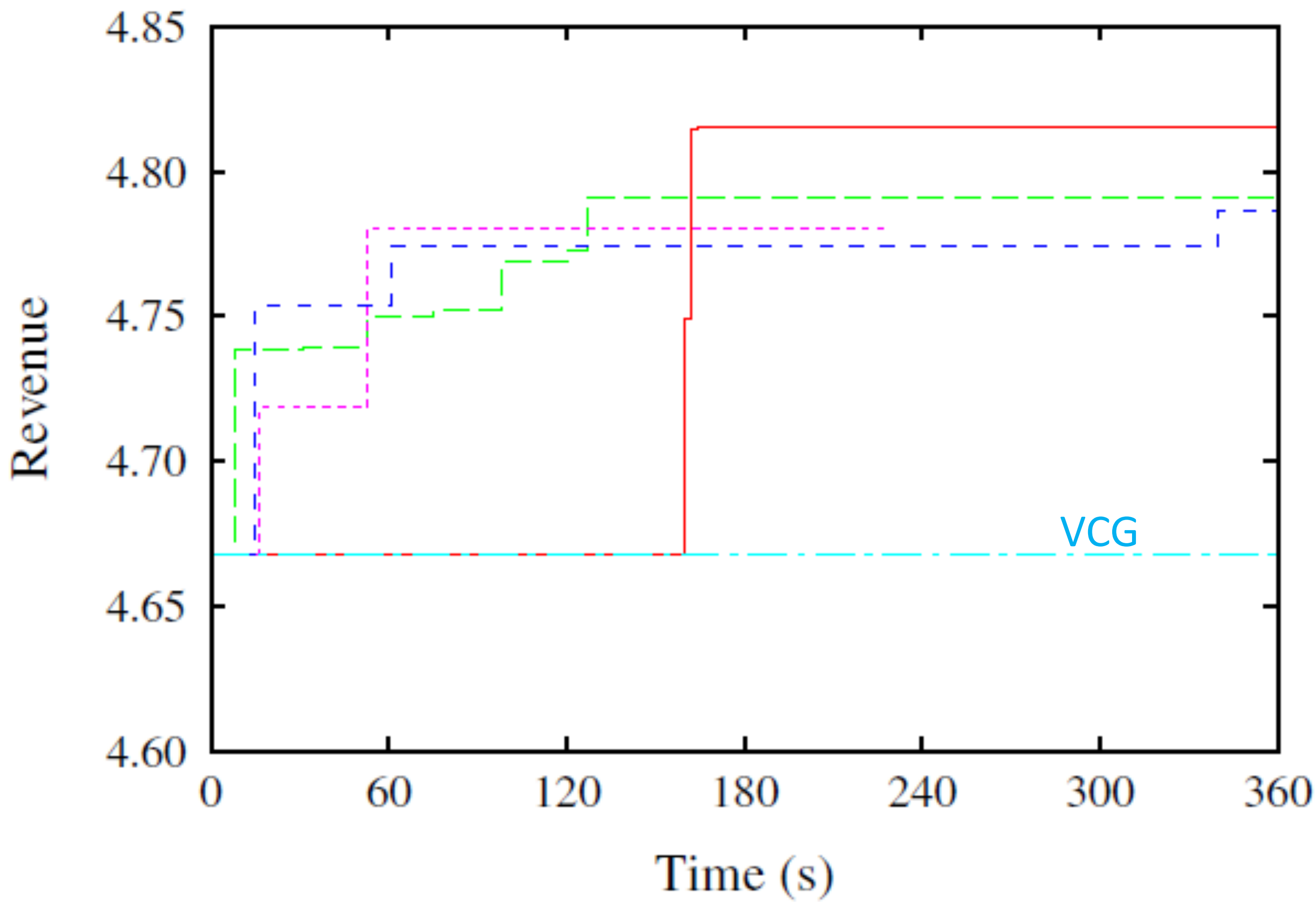
[Sandholm and Likhodedov, OR'15]

Scalability experiments (3 bidders, symmetric distribution)



[Sandholm and Likhodedov, OR‘15]

Anytime performance
(7 items, 7 bidders, symmetric distribution)



[Sandholm and Likhodedov, OR'15]

Incremental automated mechanism design [Conitzer and Sandholm IJCAI`07]

1. Start with some (manipulable) mechanism *M*
2. Find some set *F* of manipulations
   – Here a manipulation is given by an agent *i*, a type vector $\langle\theta_1,\ldots,\theta_n\rangle$, and a better type report θ'_i for agent *i*
3. If possible, change the mechanism *M* to prevent (many of) these manipulations from being beneficial
   a) make the outcome that M selects for θ more desirable for agent i (when he has type θ_i), or
   b) make the outcome that M selects for θ' less desirable for agent i (when he has type θ_i), or
   c) a combination of (a) and (b)
4. Repeat from step 2 until termination

An application of incremental automated mechanism design to a setting with payments [Conitzer and Sandholm IJCAI`07]

- Our objective *g* is to maximize some (say, linear) combination of allocative social welfare (i.e., social welfare not taking payments into account) and revenue
  - Doesn't matter what the combination is
- The set F of manipulations that we consider is that of all possible misreports (by any single agent at a time)
- We try to prevent manipulations according to (a) above (for a type vector from which there is a beneficial manipulation, make its outcome desirable enough to the manipulating agents to prevent the manipulation)
  - Among outcomes that achieve this, we choose one maximizing the objective *g*
- *Designs the **VCG** mechanism in a single iteration*

An application of incremental automated mechanism design to a setting with ordinal preferences

[Conitzer and Sandholm IJCAI`07]

- The set *F* consists of all manipulations in which a voter changes which candidate he ranks first
- We try to prevent manipulations as follows:
  For a type (vote) vector from which there is a beneficial manipulation, consider all the outcomes that may result from such a manipulation (in addition to the current outcome), and choose as the new outcome the one that minimizes #agents that still have an incentive to manipulate from this vote vector
- We'll change the outcome for each vote vector at most once
  - but we'll have multiple iterations for vote vectors
- *Designs **plurality-with-runoff** voting rule*

Incremental AMD via deep learning

[Dütting, Feng, Narasimhan, Parkes, and Ravindranath, ICML'19]

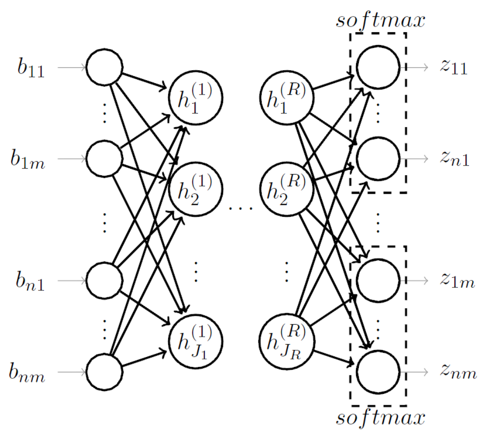
m items, n additive bidders

Parameters w

Bid of bidder i for item j: b_{ij}

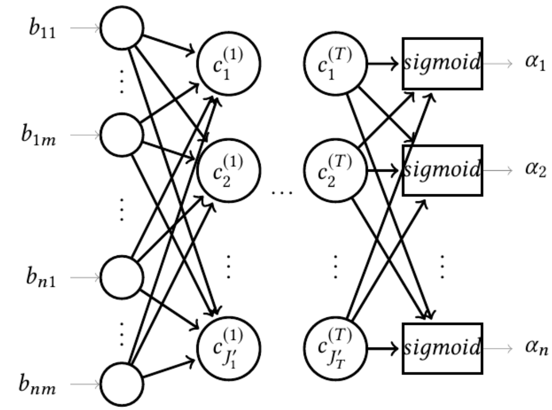
Feedback: Revenue and bidders' regret

Allocation Net



Allocation: $g^w\colon\ \mathbb{R}^{nm} \to \Delta_1 \times \cdots \times \Delta_m$

Payment Net



Payment: $p^w\colon\ \mathbb{R}^{nm} \to \mathbb{R}^n_{\geq 0}$

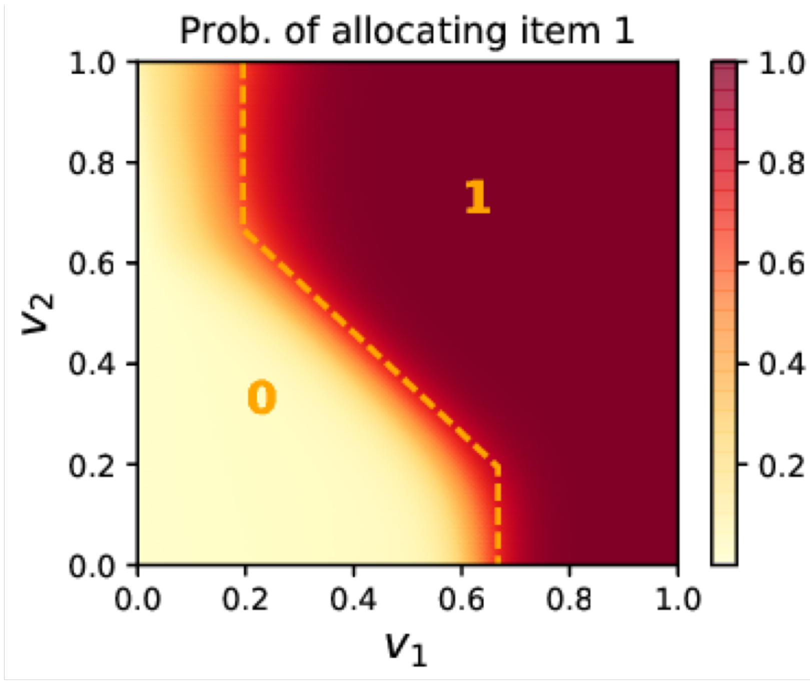
Fractional payment:

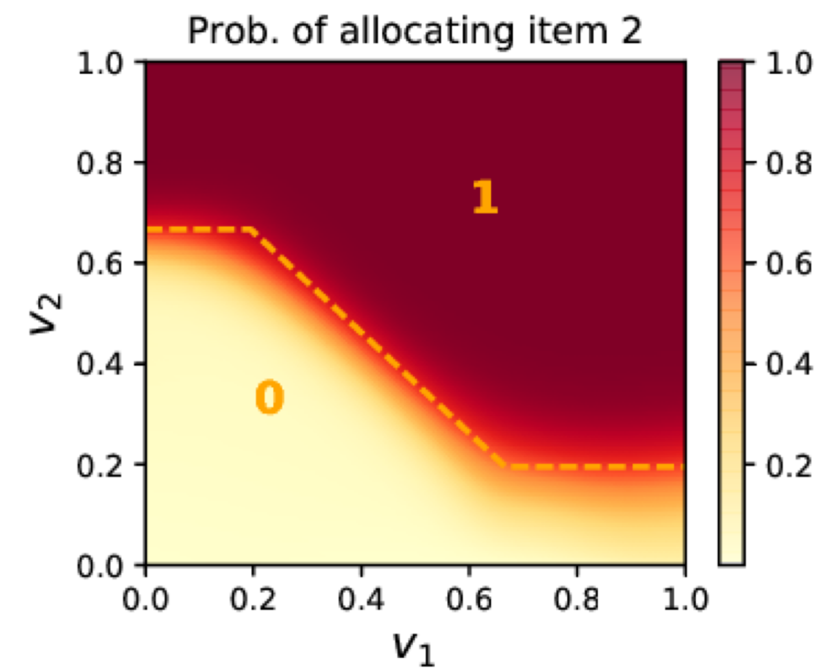
$$p_i^w = \alpha_i \cdot (g_i^w \cdot v_i), \alpha_i \in [0,1]$$

(Guarantees IR)

Incremental AMD via deep learning

[Dütting, Feng, Narasimhan, Parkes, and Ravindranath, ICML'19]





Solid regions: Learned allocation probability when single bidder with $v_1, v_2 \sim U[0,1]$

Optimal mechanism [Manelli and Vincent, JET'06] represented by regions separated by dashed orange lines

Revenue optimization using interim variables

Setting: **Single** item, known value distribution with finite support T^n

Can write single-item revenue maximization problem as LP: Find

1. Allocation function $\boldsymbol{X}: T^n \longrightarrow [0,1]^n$
2. Payment function $\boldsymbol{P}: T^n \longrightarrow \mathbb{R}^n$

with maximum expected revenue $\sum_{\boldsymbol{v} \in T^n} \mathbb{P}[\boldsymbol{v}] \sum_{i=1}^{n} P_i(\boldsymbol{v})$ s.t.

a. Allocation is always feasible

b. Mechanism is Bayes-Nash (i.e., *ex interim*) incentive compatible: $\forall i, v_i, \tilde{v}_i$,
$\mathbb{E}_{\boldsymbol{v}_{-i}}[v_i \cdot X_i(v_i, \boldsymbol{v}_{-i}) - P_i(v_i, \boldsymbol{v}_{-i})] \geq \mathbb{E}_{\boldsymbol{v}_{-i}}[v_i \cdot X_i(\tilde{v}_i, \boldsymbol{v}_{-i}) - P_i(\tilde{v}_i, \boldsymbol{v}_{-i})]$

There are $|T|^n$ variables $X_i(\boldsymbol{v})$!

[Cai, Daskalakis, and Weinberg, '12]

Revenue optimization using interim variables…

Instead, optimize over *interim* variables (single-item case):

- $x_i(v_i) = \mathbb{E}_{\boldsymbol{v}_{-i}}[X_i(v_i, \boldsymbol{v}_{-i})]$

  Expected probability bidder i receives item given bid v_i

- $p_i(v_i) = \mathbb{E}_{\boldsymbol{v}_{-i}}[P_i(v_i, \boldsymbol{v}_{-i})]$

  Bidder i's expected payment, given bid v_i

1-item thm: $n|T|$ interim variables & $n|T|$ constraints suffice

Can be generalized to multi-item for additive bidders

– Runtime remains polynomial in #bidders

– Polynomial in distribution's support size: Exponential in #items

[Cai, Daskalakis, and Weinberg, '12]

Revenue optimization and optimal transport

Setting: **Single, additive bidder** with independent values

Value distribution known

Main result [Daskalakis, Deckelbaum, and Tzamos, EC'13]:

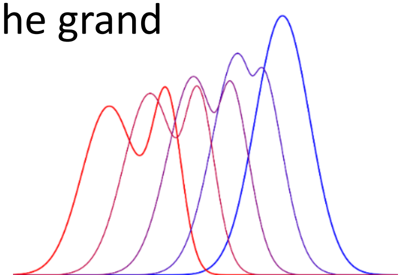
Rev. max. has dual that takes the form of optimal transport problem

OTP: Move one mass to another, minimizing cost

Dual is tight

Useful consequences:

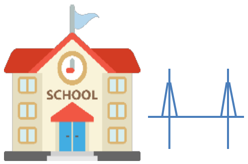
- In that setting, every optimal auction has certificate in form of transportation flow
  - Help verify whether candidate auction is optimal
- Can be a tool for characterizing optimal multi-item auctions in restricted settings
  - They studied conditions under which a take-it-or-leave-it offer for the grand bundle is optimal



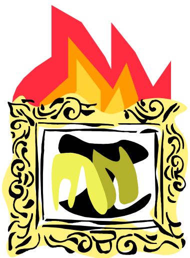
Automated mechanism design in sponsored search auctions

- Generalized second price auction is the basic mechanism used by most companies for sponsored search
  - But it has many knobs one can tweak
  - Essentially all sponsored search companies nowadays do some forms of automated mechanism design
- Optimizing mechanisms with different expressiveness – "the premium mechanism" [Benisch, Sadeh & Sandholm, Ad Auctions Workshop 2008, IJCAI-09]
  - First to use computational learning theory tools to characterize expressiveness of a mechanism [Benisch, Sandholm & Sadeh AAAI-08]
- Redoing Baidu's sponsored search auction [Sandholm 2009-13]
- Optimizing reserve prices in Yahoo!'s sponsored search auction [Ostrovsky & Schwartz EC-11]
  - See also reserve price optimization for overstock liquidation (aka "asset recovery") [Walsh, Parkes, Sandholm & Boutilier AAAI-08]
- Reinforcement learning for ad auctions: "reinforcement mechanism design" [Tang IJCAI-17, …]
- Boosted second price auction for Google's display ads [Golrezaei, Lin, Mirrokni, and Nazerzadeh, Management Science R&R]
- …

Automated mechanism design beyond sales mechanisms

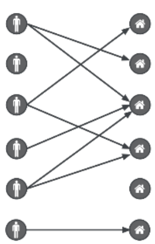


- Combinatorial public goods problems [Conitzer and Sandholm, UAI'03 Bayesian Modeling Applications Workshop]



- Real-world industrial sourcing mechanisms
- Divorce settlement mechanisms [Conitzer and Sandholm, UAI'03 Bayesian Modeling Applications Workshop]

- Reputation/recommendation systems [Jurca and Faltings, EC'06, EC'07]
- Facility location problems [Sui, Boutilier, and Sandholm, IJCAI'13]



- Assignment mechanisms [Narasimhan and Parkes, UAI'16]
- Mechanism design without money [Narasimhan, Agarwal and Parkes, IJCAI'16]
- Redistribution mechanisms [Guo and Conitzer, EC'07, AAMAS'08, EC'08, EC'09, AI'10, AIJ'14; Nath and Sandholm, WINE'16, GEB'19…]
- …

Outline

1. Mechanism design basics
2. Introduction to automated mechanism design (AMD)
3. Sample complexity guarantees for AMD
4. Additional AMD algorithms
5. Other learning models

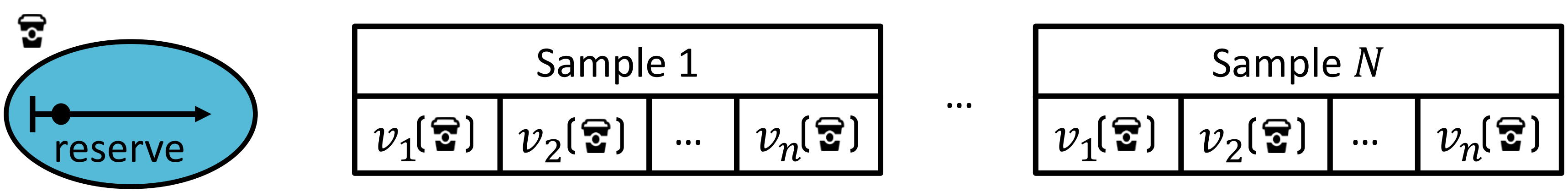
Mechanism design as a learning problem

Goal: Given mechanism family $\mathcal{M}$ and set of buyers' values sampled from unknown distribution $\mathcal{D}$, find mechanism with high expected revenue

- **Large family $\mathcal{M}$ of parametrized mechanisms**

  (E.g., 2nd-price auctions w/ reserves or posted price mechanisms)
- **Set of buyers' values sampled from unknown distribution $\mathcal{D}$**

2nd price auctions with reserves:



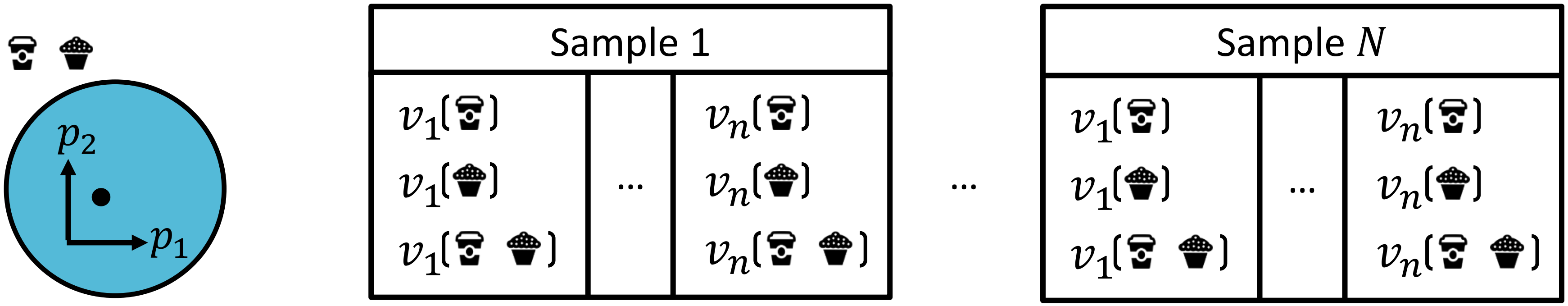
Mechanism design as a learning problem

Goal: Given mechanism family $\mathcal{M}$ and set of buyers' values sampled from unknown distribution $\mathcal{D}$, find mechanism with high expected revenue

- **Large family $\mathcal{M}$ of parametrized mechanisms**

  (E.g., 2nd-price auctions w/ reserves or posted price mechanisms)
- **Set of buyers' values sampled from unknown distribution $\mathcal{D}$**

Posted price mechanisms:



Mechanism design as a learning problem

Goal: Given mechanism family $\mathcal{M}$ and set of buyers' values sampled from unknown distribution $\mathcal{D}$, find mechanism with high expected revenue

Approach: Find $\widehat{M}$ (nearly) optimal mechanism over the set of samples.

Mechanism design as a learning problem

Goal: Given mechanism family $\mathcal{M}$ and set of buyers' values sampled from unknown distribution $\mathcal{D}$, find mechanism with high expected revenue

Approach: Find $\widehat{M}$ (nearly) optimal mechanism over the set of samples

Key question: Will $\widehat{M}$ have high expected revenue?

Technical tool: uniform convergence

For any mechanism in class $\mathcal{M}$, average revenue over samples close to its expected revenue

Implies $\widehat{M}$ has high expected revenue

Mechanism design as a learning problem

Goal: Given mechanism family $\mathcal{M}$ and set of buyers' values sampled from unknown distribution $\mathcal{D}$, find mechanism with high expected revenue

Approach: Find $\widehat{M}$ (nearly) optimal mechanism over the set of samples

Key question: Will $\widehat{M}$ have high expected revenue?

Technical tool: uniform convergence

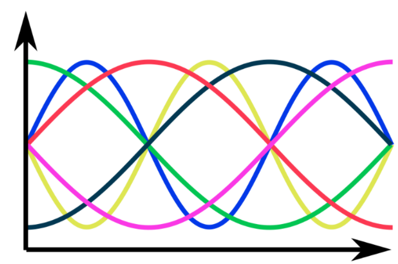
Learning theory: $\tilde{O}(\dim(\mathcal{M})\ /\epsilon^2)$ samples suffice for ϵ-close

Mechanism design as a learning problem

Goal: Given mechanism family $\mathcal{M}$ and set of buyers' values sampled from unknown distribution $\mathcal{D}$, find mechanism with high expected revenue

Learning theory: $\tilde{O}(\mathrm{dim}(\mathcal{M})\ /\epsilon^2)$ samples suffice for ϵ-close

$\mathrm{dim}(\mathcal{M})$ (e.g. pseudo-dim): ability of revenue fns to fit complex patterns



Challenge: analyze dim($\mathcal{M}$) for complex combinatorial, modular mechanisms

Mechanism design as a learning problem

Goal: Given mechanism family $\mathcal{M}$ and set of buyers' values sampled from unknown distribution $\mathcal{D}$, find mechanism with high expected revenue

Learning theory: $\tilde{O}(\dim(\mathcal{M})\ /\epsilon^2)$ samples suffice for ϵ-close

Our results [Balcan, Sandholm, Vitercik, EC'18]:
General way to bound $\dim(\mathcal{M})$ for any mechanism class satisfying **key structural property**: revenue is piecewise linear function of class's parameters

Many applications to multi-item, multi-buyer scenarios

Second-price auctions with reserves, posted price mechanisms, two-part tariffs, parameterized VCG mechanisms, etc.

Uniform Convergence of Auctions

- Digital goods (unrestricted supply): Balcan, Blum, Hartline, and Mansour [FOCS'05] were first to use **learning-theoretic tools** to design and analyze auctions.
- Mohri and Medina [ICML'14] use a combination of **pseudo-dimension** and **Rademacher complexity** to analyze second-price auctions with reserves.
- Morgenstern and Roughgarden provide **pseudo-dimension** bounds for t-level auctions [NIPS'15] and "simple" (by design) multi-item mechanisms [COLT'16].
- Balcan, Sandholm, and Vitercik [NIPS'16, EC'18] give **general theorem** for bounding pseudo-dimension of multi-item mechanism classes.
- Syrgkanis [NIPS'17] provides a **new complexity measure** (the "split-sample growth rate" based on Rademacher complexity) to analyze auction classes.
- Cai and Daskalakis [FOCS'17] give a new complexity measure implying uniform convergence bounds when the underlying distribution is a **product distribution**.
- Devanur, Huang, and Psomas [STOC'16] and Gonczarowski and Nisan [STOC'17] give **covering-style** analyses for single-item settings.

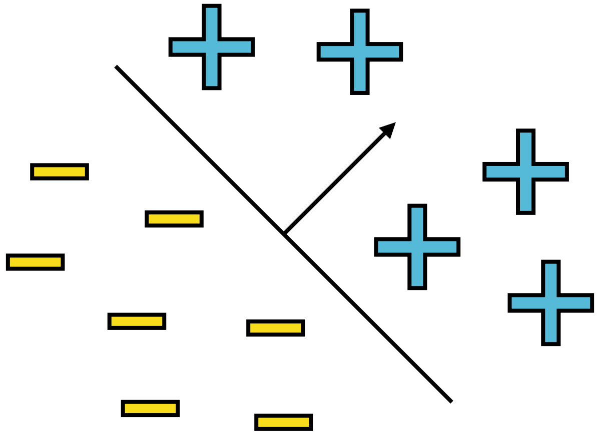
VC-theory brief tour

VC dimension

Complexity measure characterizing the sample complexity of **binary-valued** function classes

(Classes of functions $h : \mathcal{X} \to \{-1,1\}$)

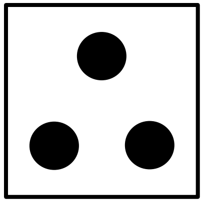
E.g., linear separators



VC dimension

VC-dimension of a function class $\mathcal{H}$ is the size of the largest set $\mathcal{S} \subseteq \mathcal{X}$ that can be labeled in all $2^{|\mathcal{S}|}$ ways by functions in $\mathcal{H}$.

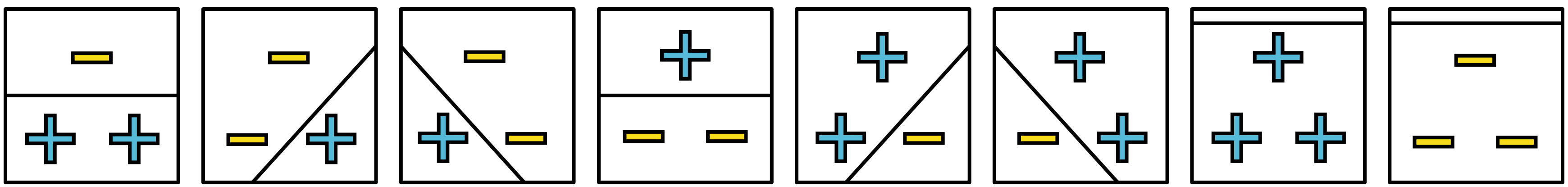
Example: $\mathcal{H}$ = Linear separators in $\mathbb{R}^2$. VCdim($\mathcal{H}$) ≥ 3.



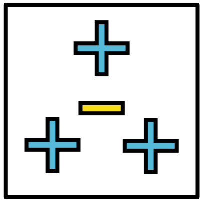
VC dimension

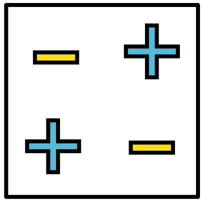
VC-dimension of a function class $\mathcal{H}$ is the size of the largest set $\mathcal{S} \subseteq \mathcal{X}$ that can be labeled in all $2^{|\mathcal{S}|}$ ways by functions in $\mathcal{H}$.

Example: $\mathcal{H}$ = Linear separators in $\mathbb{R}^2$. VCdim($\mathcal{H}$) ≥ 3.



VCdim($\mathcal{H}$) ≤ 3.





VCdim({Linear separators in $\mathbb{R}^d$}) $= d + 1$.

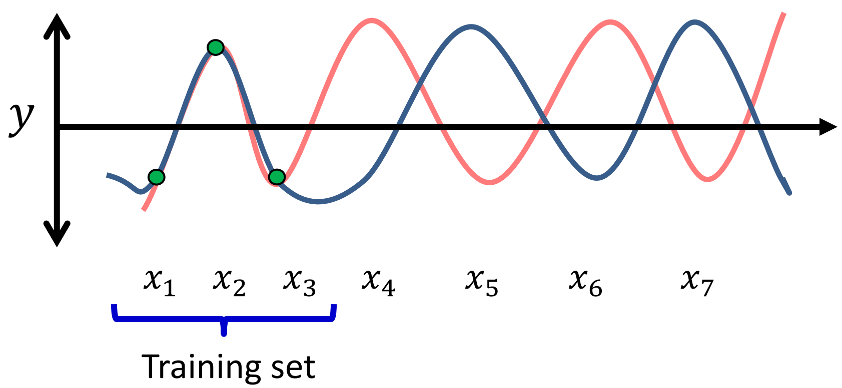
Why VC dimension matters

Why does it matter:

"how many points we can label in all ways with functions from the class"?

Example: H={all binary functions over some domain}, then $\mathrm{VCdim(H)} = \infty$

Given training set (pts & labels), there exist fns in H that label training set correctly, but provide complete opposite answers everywhere else



Why VC dimension matters

Why does it matter:
“how many points we can label in all ways with functions from the class”?

Sauer’s Lemma: If d = VCdim(H), then any set of points size m>d, can be labelled only in $\mathrm{O}(\mathrm{m}^{d})$ ways with functions from the class.

Not all 2^{m} labelings are achievable!

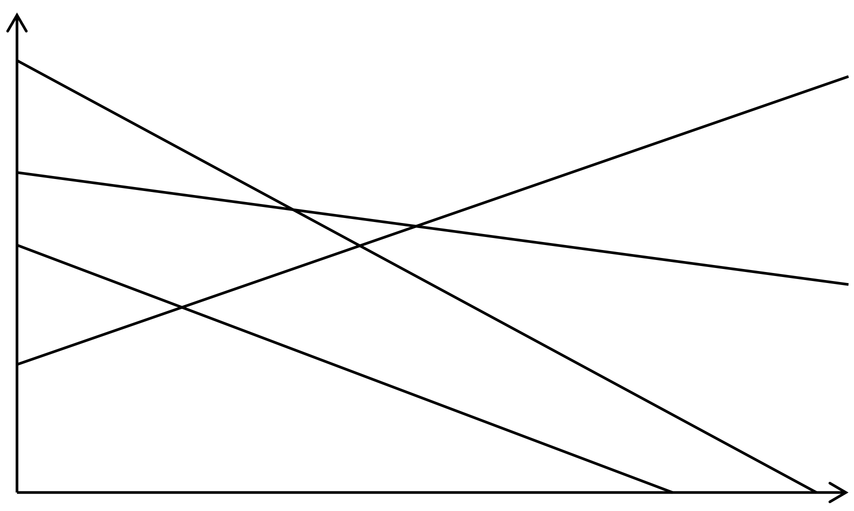
Sample complexity:
$\mathrm{O}(\mathrm{VCdim}(H)\ /\epsilon^{2})$ training instances suffice for generalizability

Pseudo-dimension

Complexity measure characterizing the sample complexity of **real-valued** function classes

(Classes of functions $f : \mathcal{X} \to [0,1]$*)*

E.g., affine functions

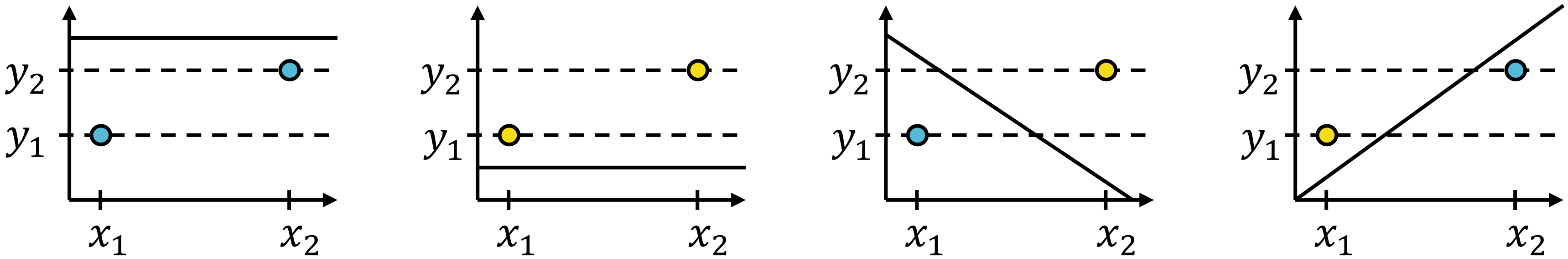


Pseudo-dimension

The **pseudo-dimension** of a function class $\mathcal{F}$ is the size of the largest set $\mathcal{S} = \{x_1, \ldots, x_N\} \subseteq \mathcal{X}$ s.t. for some thresholds $y_1, \ldots, y_N \in \mathbb{R}$, all 2^N above/below binary patterns can be achieved by functions in $\mathcal{F}$.

Example: $\mathcal{F}$ = Affine functions in $\mathbb{R}$.

$\mathrm{Pdim}(\mathcal{F}) \geq 2$.



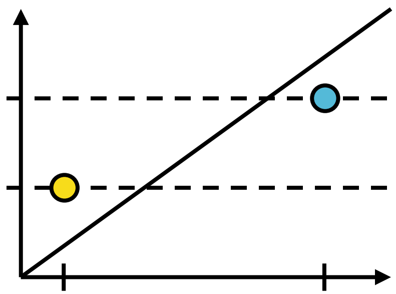
Pseudo-dimension

The **pseudo-dimension** of a function class $\mathcal{F}$ is the size of the largest set $\mathcal{S} = \{x_1, \dots, x_N\} \subseteq \mathcal{X}$ s.t. for some thresholds $y_1, \dots, y_N \in \mathbb{R}$, all 2^N above/below binary patterns can be achieved by functions in $\mathcal{F}$.

Theorem [Pollard, 1984]

For any $\epsilon \in (0,1)$ and any distribution $\mathcal{D}$ over $\mathcal{X}$, with high probability over the draw of $N = \widetilde{\Theta}\left(\frac{\mathbf{Pdim}(\mathcal{F})}{\epsilon^2}\right)$ samples $\{x_1, \dots, x_N\} \sim \mathcal{D}^N$, for all $f \in \mathcal{F}$,

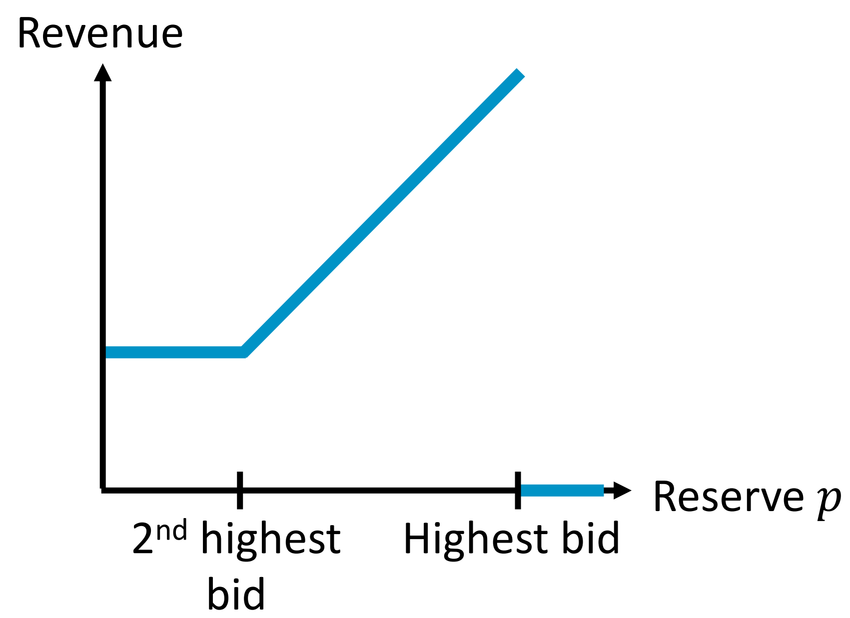
$$\left| \mathbb{E}_{x \sim \mathcal{D}}[f(x)] - \frac{1}{N} \sum_{i=1}^{N} f(x_i) \right| \leq \epsilon.$$



Example:
P-dim of 2^{nd}-price auctions with reserves

2^{nd}-price auction with a reserve

- Auctioneer sets reserve price p
- Highest bidder wins if bid $\geq p$. Pays maximum of second highest bid and p



Claim

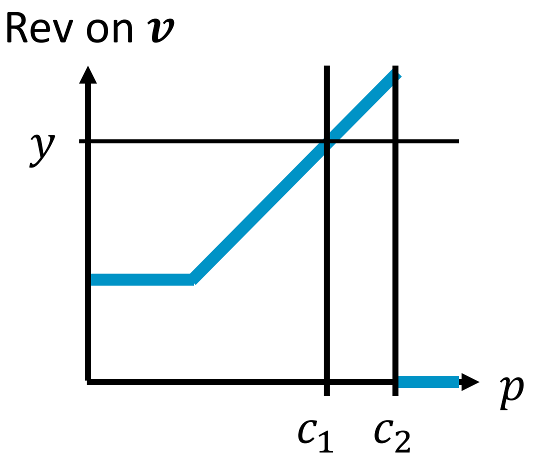
For a fixed set of bids, revenue is a piecewise linear function of the reserve.

Example:
P-dim of 2nd-price auctions with reserves

Theorem [Mohri-Medina'14; Morgenstern-Roughgarden'16; Balcan-Sandholm-V.'18]
$\mathcal{M} = \{\mathrm{rev}_p :=$ revenue of 2nd-price auction with reserve $p\}$. $\mathrm{Pdim}(\mathcal{M}) \leq 2$.

Key idea: Consider some valuation vector $\boldsymbol{v}$ and revenue-threshold y.

- Ranging p from 0 to ∞, will be (at most) two cutoff values c_1,c_2 where revenue goes from "below" to "above" to "below"
- With N examples, look at all $2N$ cutoff values
- All p in same interval between consecutive cutoff values will give same binary pattern
- So, at most $2N + 1$ binary patterns
- Pseudo-dimension is max N s.t. all 2^N binary above/below patterns are achievable
  - Need $2^N \leq 2N + 1$, so $N \leq 2$



Bounding pseudo-dim of mechanism classes

Theorem

Suppose:

1. The mechanism class $\mathcal{M}$ is parameterized by vectors $\boldsymbol{p} \in \mathbb{R}^{\boldsymbol{d}}$

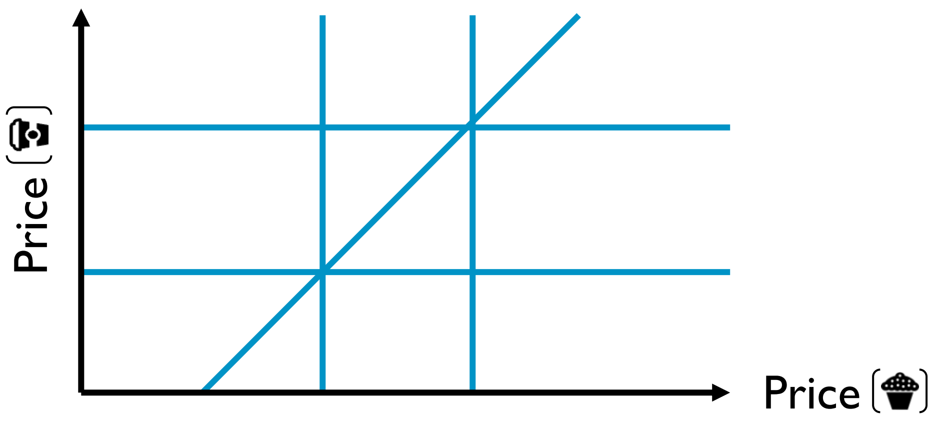
For example, $\boldsymbol{p} = \Big($ price $\Big($  $\Big)$, price $\Big($  $\Big)\Big)$

Bounding pseudo-dim of mechanism classes

Theorem

Suppose:

1. The mechanism class $\mathcal{M}$ is parameterized by vectors $\boldsymbol{p} \in \mathbb{R}^{\boldsymbol{d}}$
2. For every set $\boldsymbol{v}$ of buyers' values, a set of $\leq \boldsymbol{t}$ hyperplanes partition $\mathbb{R}^{\boldsymbol{d}}$ such that in every cell of this partition, $\text{revenue}_{\boldsymbol{v}}(\boldsymbol{p})$ is linear



In this example, $d = 2$ and $t = 5$.

Bounding pseudo-dim of mechanism classes

Theorem

Suppose:

1. The mechanism class $\mathcal{M}$ is parameterized by vectors $\boldsymbol{p} \in \mathbb{R}^{\boldsymbol{d}}$
2. For every set $\boldsymbol{v}$ of buyers' values, a set of $\leq \boldsymbol{t}$ hyperplanes partition $\mathbb{R}^{\boldsymbol{d}}$ such that in every cell of this partition, $\mathrm{revenue}_{\boldsymbol{v}}(\boldsymbol{p})$ is linear

Then $\mathrm{Pdim}(\mathcal{M}) = O(\boldsymbol{d}\log(\boldsymbol{dt}))$.

Bounding pseudo-dim of mechanism classes

Corollary

Suppose:

1. The mechanism class $\mathcal{M}$ is parameterized by vectors $\boldsymbol{p} \in \mathbb{R}^{\boldsymbol{d}}$
2. For every set $\boldsymbol{v}$ of buyers' values, a set of $\leq \boldsymbol{t}$ hyperplanes partition $\mathbb{R}^{\boldsymbol{d}}$ such that in every cell of this partition, $\text{revenue}_{\boldsymbol{v}}(\boldsymbol{p})$ is linear

For any $\epsilon \in (0,1)$, with high probability over the draw of $N = \tilde{\Theta}\left(\frac{\boldsymbol{d} \log(\boldsymbol{dt})}{\epsilon^2}\right)$ samples $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(N)}\right\} \sim \mathcal{D}^N$, for all mechanisms in $\mathcal{M}$:

$$|\text{average revenue over } \mathcal{S} - \text{expected revenue}| \leq \epsilon.$$

High-level learning theory bit

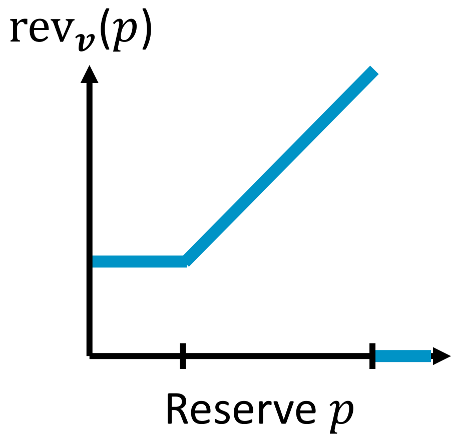
Theorem (Informal)
$\boldsymbol{d}$-dim. parameter space, $\boldsymbol{t}$ hyperplanes splitting parameters into linear pieces
$\Rightarrow \text{Pdim}(\mathcal{M}) = O(\boldsymbol{d} \log(\boldsymbol{dt}))$

Want to prove that for any mechanism parameters $\boldsymbol{p}$:

$$\frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{v}\in\mathcal{S}} \text{rev}_{\boldsymbol{p}}(\boldsymbol{v}) \text{ close to } \mathbb{E}[\text{rev}_{\boldsymbol{p}}(\boldsymbol{v})]$$

Function class we analyze pseudo-dimension of:
$\{\text{rev}_{\boldsymbol{p}}: \text{parameters } \boldsymbol{p} \in \mathbb{R}^{\boldsymbol{d}}\}$

Proof takes advantage of structure exhibited by **dual class** $\{\text{rev}_{\boldsymbol{v}}: \text{buyer values } \boldsymbol{v}\}$



$$\text{rev}_{\boldsymbol{v}}(\boldsymbol{p}) = \text{rev}_{\boldsymbol{p}}(\boldsymbol{v})$$

Our main applications of our general theorem

- Match or improve over the best-known guarantees for many of the classes previously studied.
- Prove bounds for classes not yet studied from a learning perspective.

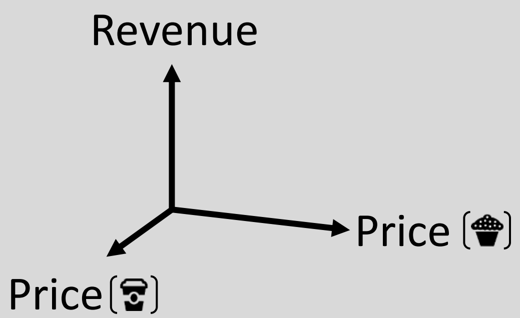
Mechanism class	**Sample complexity studied before?**
Randomized mechanisms (lotteries)	NA
Multi-part tariffs and other non-linear pricing mechanisms	NA
Posted price mechanisms	E.g., Morgenstern-Roughgarden COLT'16; Syrgkanis NIPS'17
Affine maximizer auctions	Balcan-Sandholm-Vitercik NIPS '16
Second price auctions with reserves	E.g., Morgenstern-Roughgarden COLT'16; Devanur et al. STOC'16

Application: Posted price mechanisms

$\mathcal{M}$ = multi-item, multi-buyer posted price mechanisms

Mechanism designer sets price per item

1. Buyer 1 arrives. Buys bundle maximizing his utility
2. Buyer 2 arrives. Buys remaining bundle maximizing his utility…



Studied extensively in econ-CS
[e.g., Feldman, Gravin, and Lucier, SODA’15; Babaioff, Immorlica, Lucier, and Weinberg, FOCS’14; Cai, Devanur, and Weinberg, STOC’16]

Pseudo-dimension of posted price mechanisms

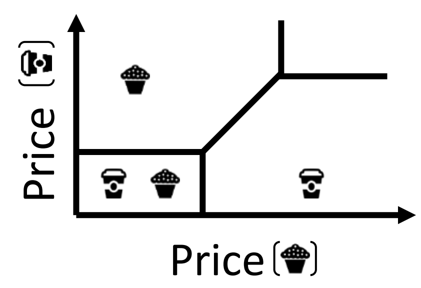
Theorem

Pdim$(\mathcal{M}) = O(d \log(dt))$ w/ $d = (\text{\#dimensions}) = (\text{\#items})$ and $t = (\text{\# hyperplanes}) = (\text{\# buyers}) \cdot \binom{2^{(\text{\# items})}}{2}$.

Proof sketch. For **every buyer** and **every pair of bundles**:

Hyperplane defines where buyer prefers each bundle

- t hyperplanes define where buyers' preference orders fixed
- When preference ordering fixed, bundles they buy are fixed
  - So revenue is linear function of prices of items they buy



Pseudo-dimension of posted price mechanisms

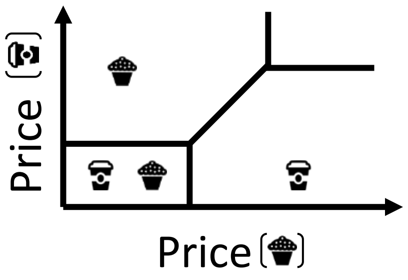
Theorem

$\text{Pdim}(\mathcal{M}) = O(d \log(dt))$ w/ $d = (\#\text{dimensions}) = (\#\text{items})$ and $t = (\#\text{ hyperplanes}) = (\#\text{ buyers}) \cdot \binom{2^{(\#\text{ items})}}{2}$.

Corollary

$\text{Pdim}(\mathcal{M}) = \tilde{O}\left((\#\text{items})^{2}\right)$

Also shown by Morgenstern and Roughgarden [COLT '16]

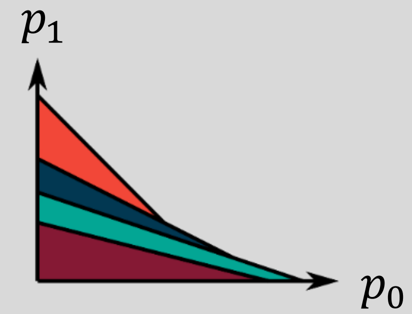


Two-part tariffs

Application: Single-item, multi-buyer two-part tariffs

- Multiple units of item for sale.
- Seller sets upfront fee p_0, fee per unit p_1.
- If buyer buys k units, pays $p_0 + k \cdot p_1$.
- Each buyer buys number of units maximizing utility.
- Seller offers "menu" of L tariffs.
  - Buyer chooses tariff and number of units to buy maximizing utility

Studied for decades in economics
[e.g., Oi, Quarterly Journal of Economics '71;
Feldstein, Quarterly Journal of Economics '72]



Pseudo-dimension of two-part tariff menus

Theorem

$\text{Pdim}(\mathcal{M}) = O(d \log(dt))$ with $d = (\#\text{dimensions}) = 2L$ and $t = (\#\text{hyperplanes}) = (\#\text{buyers})\binom{L(\#\text{units})}{2}$.

Proof sketch.

For every **buyer** & every **pair of (tariff, #units bought) tuples**:

Hyperplane defines where buyer prefers one tuple over other

- t hyperplanes define where buyers' preference orders fixed
- When preference ordering fixed, tariff and #units bought fixed
  - So revenue is linear function of upfront fee and price per unit

Pseudo-dimension of two-part tariff menus

Theorem

Pdim$(\mathcal{M}) = O(d \log(dt))$ with $d = (\#\text{dimensions}) = 2L$ and $t = (\#\text{hyperplanes}) = (\#\text{buyers})\binom{L(\#\text{units})}{2}$.

Corollary

Pdim$(\mathcal{M}) = \tilde{O}(L)$

Randomized mechanisms (lotteries)

Application: Multi-item lotteries for one additive buyer (generalizes easily to multiple unit-demand or additive buyers)

– Lottery represented by vector $(\phi_1, \ldots, \phi_{(\#\mathrm{items})})$ and price p

– If buyer buys lottery, pays p and receives each item i w.p. ϕ_i

  - Expected utility is $\sum_{i=1}^{(\#\mathrm{items})} v(\{i\}) \cdot \phi_i - p$

– Seller offers “menu” of L lotteries for buyer to choose from

  - Buyer chooses expected-utility-maximizing lottery (or buys nothing)

Studied extensively in econ-CS

[e.g., Briest, Chawla, Kleinberg, and Weinberg, SODA’10; Chawla, Malec, and Sivan, EC’10; Babioff, Gonczarowski, and Nisan, STOC’17]

Pseudo-dimension of lotteries

Theorem

$\mathrm{Pdim}(\mathcal{M}) = O(d \log(dt))$ with $t = (\#\text{ hyperplanes}) = L^2$

$d = (\#\text{dimensions}) = O\big((\#\text{items}) \cdot L\big)$

Proof sketch. Proof similar to previous.

Pseudo-dimension of lotteries

Theorem

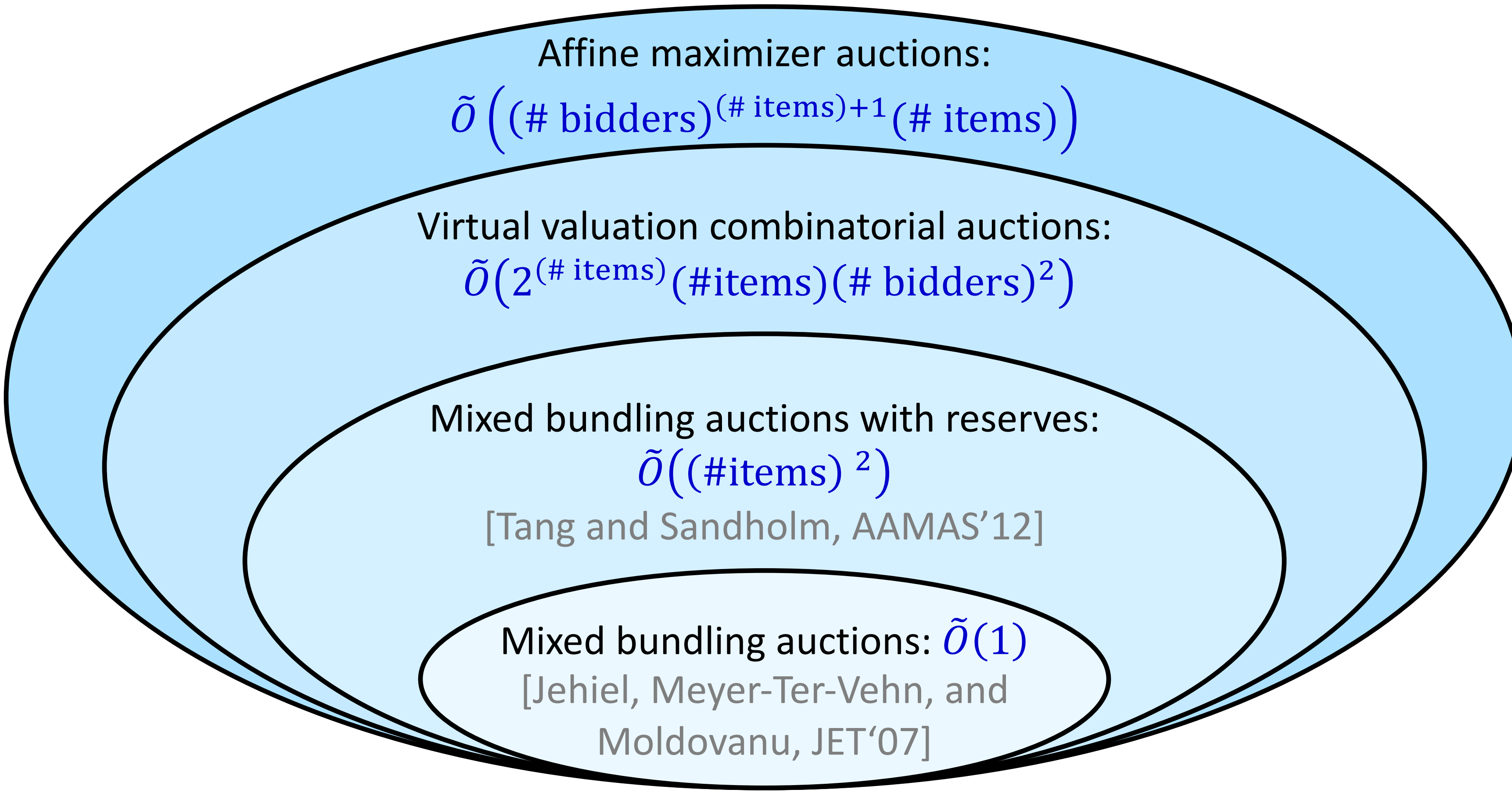
$\text{Pdim}(\mathcal{M}) = O(d\log(dt))$ with $t = (\#\text{ hyperplanes}) = L^2$

$d = (\#\text{dimensions}) = O\big((\#\text{items}) \cdot L\big)$

Corollary

$\text{Pdim}(\mathcal{M}) = \tilde{O}\big(L(\#\text{items})\big)$

Affine maximizer auction pseudo-dimension



[Balcan, Sandholm, and Vitercik, EC'18]

Additional applications of our general theorem

Multi-item, multi-unit non-linear pricing mechanisms

[E.g., Wilson, Oxford Press '93]

λ-auctions

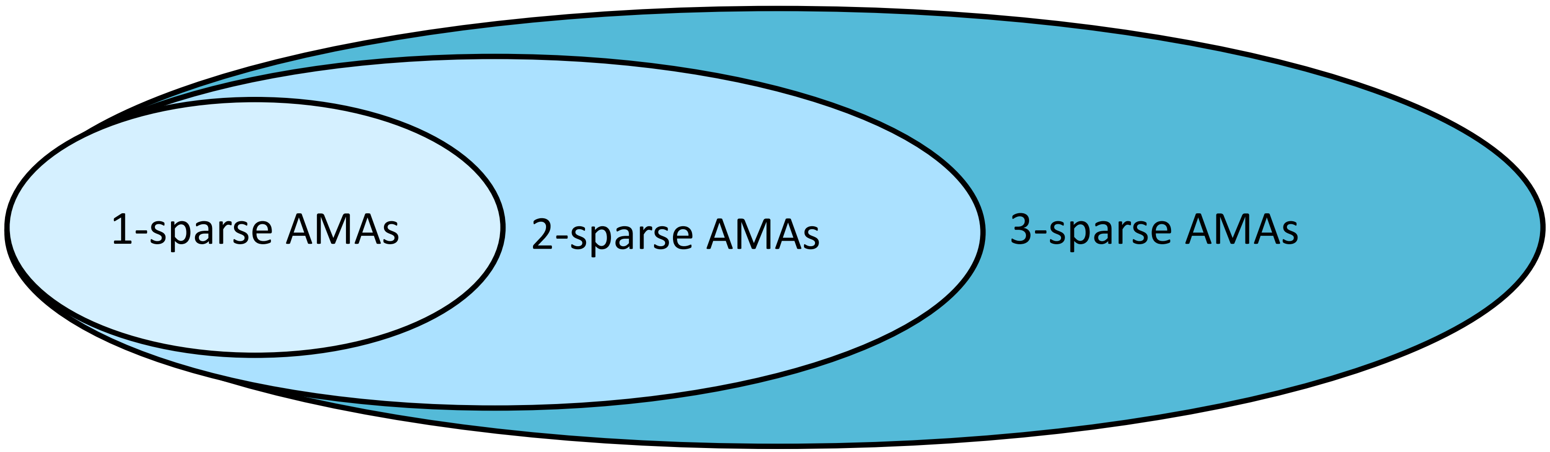
[Jehiel, Meyer-Ter-Vehn, and Moldovanu, J. of Econ. Theory '07]

Fine-grained auction hierarchies

Fine-grained **hierarchies** of AMAs:

– k-sparse AMAs: $\leq k$ allocation boosts

$$|\text{empirical revenue - expected revenue}| \leq \tilde{O}\left(\sqrt{\frac{\#\text{bidders}+k}{|S|}}\right)$$



Fine-grained auction hierarchies

Fine-grained **hierarchies** of AMAs:

- k-sparse AMAs: $\leq k$ allocation boosts

  $|\text{empirical revenue - expected revenue}| \leq \tilde{O}\left(\sqrt{\frac{\#\text{bidders}+k}{|S|}}\right)$

- A-boosted AMAs: only allocations in A boosted

  $|\text{empirical revenue - expected revenue}| \leq \tilde{O}\left(\sqrt{\frac{\#\text{bidders}+|A|}{|S|}}\right)$

Fine-grained auction hierarchies

Fine-grained **hierarchies** of AMAs:

– k-sparse AMAs: $\leq k$ allocation boosts

$$|\text{empirical revenue - expected revenue}| \leq \tilde{O}\left(\sqrt{\frac{\#\text{bidders}+k}{|S|}}\right)$$

– A-boosted AMAs: only allocations in A boosted

$$|\text{empirical revenue - expected revenue}| \leq \tilde{O}\left(\sqrt{\frac{\#\text{bidders}+|A|}{|S|}}\right)$$

Increasing k and $|A|$ means looser bounds,
but greater chance class contains high-revenue auction

Inevitably, there's a **revenue-generalization tradeoff**

Optimizing the revenue-generalization tradeoff

We provide guarantees for **optimizing this tradeoff**

E.g., k-sparse AMAs $\mathcal{M}_k$:

Theorem

Let $\widehat{M} = \mathbf{argmax}_{k, M \in \mathcal{M}_k} \left\{ \text{Empirical revenue of } M - \tilde{O}\left(\sqrt{\frac{\#\text{bidders} + k}{|S|}}\right) \right\}$

Increases with k

Decreases with k

Let k^* be optimal AMA's sparsity level.

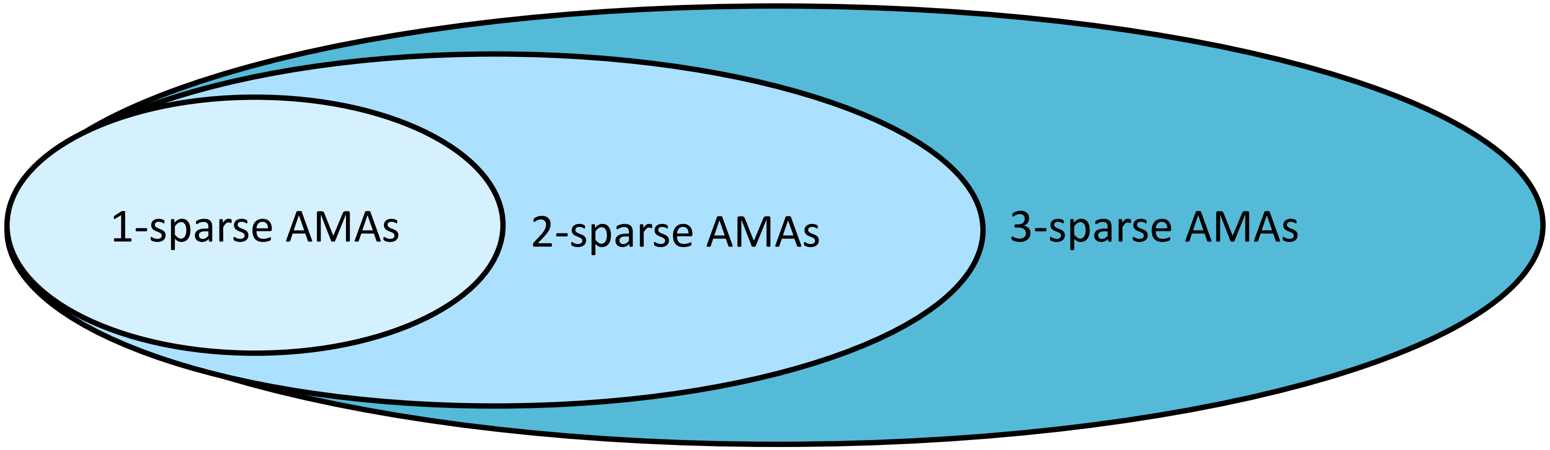
$\widehat{M}$'s revenue is within $\tilde{O}\left(\sqrt{\frac{\#\text{bidders} + k^*}{|S|}}\right)$ of optimal AMA's revenue.

Structural revenue maximization

Structural revenue maximization:

Optimize tradeoff between increasing empirical revenue…

and keeping mechanism class **simple**



Structural revenue maximization

Structural revenue maximization:

Optimize tradeoff between increasing empirical revenue…

and keeping mechanism class **simple**

Extensive literature on **structural risk minimization** research [e.g., Vapnik and Chervonenkis, Theory of Pattern Recognition, ’74; Blumer, Ehrenfeucht, Haussler, and Warmuth, Information Processing Letters ’87; Vapnik, Springer ’95]

Outline

1. Mechanism design basics
2. Introduction to automated mechanism design (AMD)
3. Sample complexity guarantees for AMD
4. Additional AMD algorithms
   a) Empirical Myerson
   b) Ad auctions
5. Other learning models

Empirical Myerson (single-item setting)

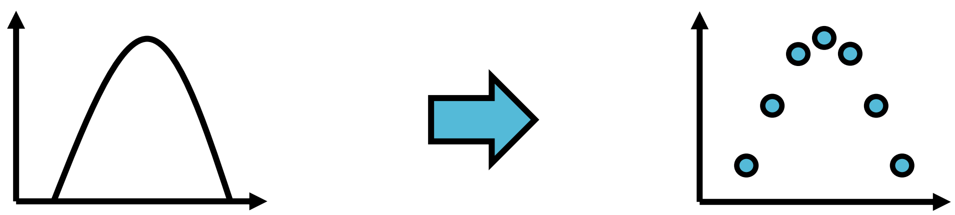
Single-item setting: **empirical variants** of Myerson's optimal auction, beginning with work by Elkind [SODA'07].

- Cole and Roughgarden [STOC'14] first to study **distributions with infinite support**.
- Morgenstern and Roughgarden [NIPS'15] provide **pseudo-dimension** bounds for t-level auctions (approximate Myerson's auction).
- Devanur, Huang, and Psomas [STOC'16] analyze model where the auction designer receives **side information** about buyers (e.g., income).
- Devanur, Huang, and Psomas [STOC'16] and Gonczarowski and Nisan [STOC'17] give algorithms when buyer values drawn from **non-i.i.d., irregular distributions**.
- Bubeck, Devanur, Huang, and Niazadeh [EC'17] provide upper and lower sample complexity bounds via **online learning** regret analysis.

Empirical Myerson (single-item setting)

Empirical Myerson (informal)

1. Draw samples
2. Formulate an empirical distribution $\widehat{\mathcal{D}}_i$ using the samples
3. Return Myerson's auction defined over $\widehat{\mathcal{D}}_1 \times \cdots \times \widehat{\mathcal{D}}_n$



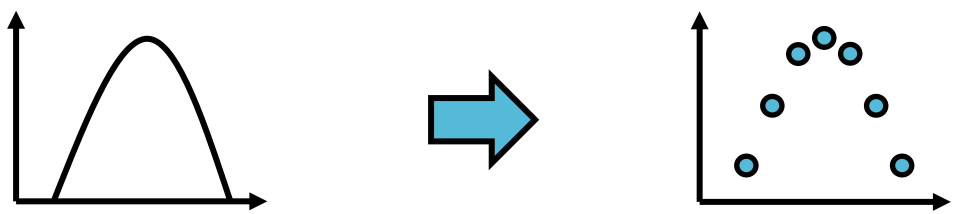
Cole and Roughgarden, STOC'14
Roughgarden and Schrijvers, EC'16

Devanur, Huang, and Psomas, STOC'16
Gonczarowski and Nisan, STOC'17

Empirical Myerson (single-item setting)

Empirical Myerson (informal)

1. Draw samples
2. Formulate an empirical distribution $\widehat{\mathcal{D}}_i$ using the samples
3. Return Myerson's auction defined over $\widehat{\mathcal{D}}_1 \times \cdots \times \widehat{\mathcal{D}}_n$



Settling the Sample Complexity
of Single-Parameter Revenue Maximization

Guo, Huang, and Zhang, STOC'19

Tight sample complexity upper and lower bounds

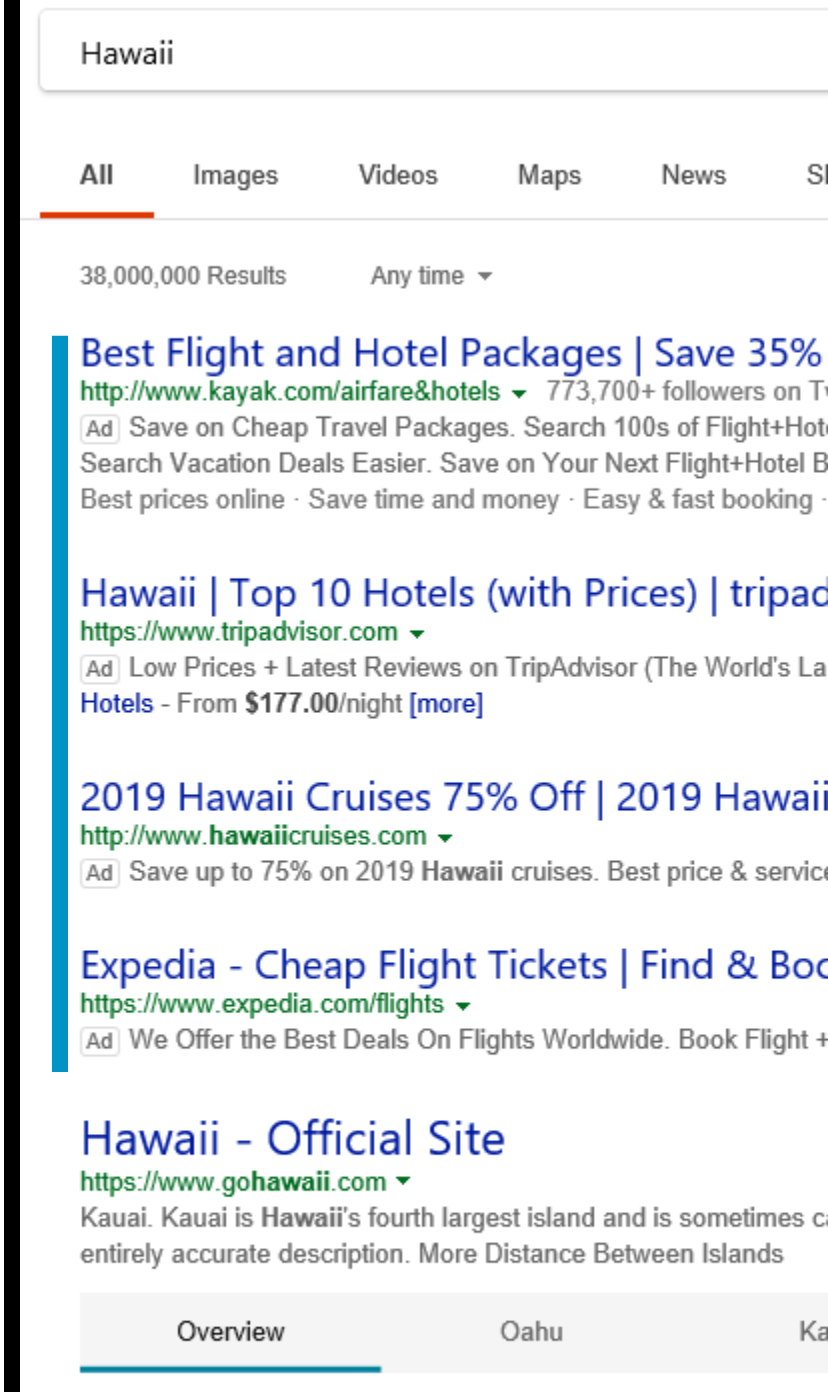
Single-item auctions, continued

- **Non-truthful auctions:** Chawla, Hartline, and Nekipelov [EC'14] study non-truthful auctions, where bidders' valuations inferred from equilibrium bids. They design non-truthful auctions that have equilibrium revenue within a constant of optimal.
- **Lower bounds:** Huang, Mansour, and Roughgarden [EC'15] provide sample complexity bounds for posted-price, single-item mechanisms. They give lower bounds using tools from differential privacy and information theory.
- **Submultiplicative Glivenko-Cantelli:** Alon, Babaioff, Mansour, Moran, and Yehudayoff [NIPS'17] derive a variant of the Glivenko-Cantelli Theorem which they apply to learning second-price auctions with reserves.
- **Learning from constant number of samples:** Dhangwatnotai, Roughgarden, and Yan [EC'10], Fu, Immorlica, Lucier, and Strack [EC '15], Goldner and Karlin [WINE'16], Babaioff, Gonczarowski, Mansour, and Moran [EC'18].
- **Ad auction design:** Mohri and Medina [UAI'15], Medina and Vassilvitskii [NIPS'17].

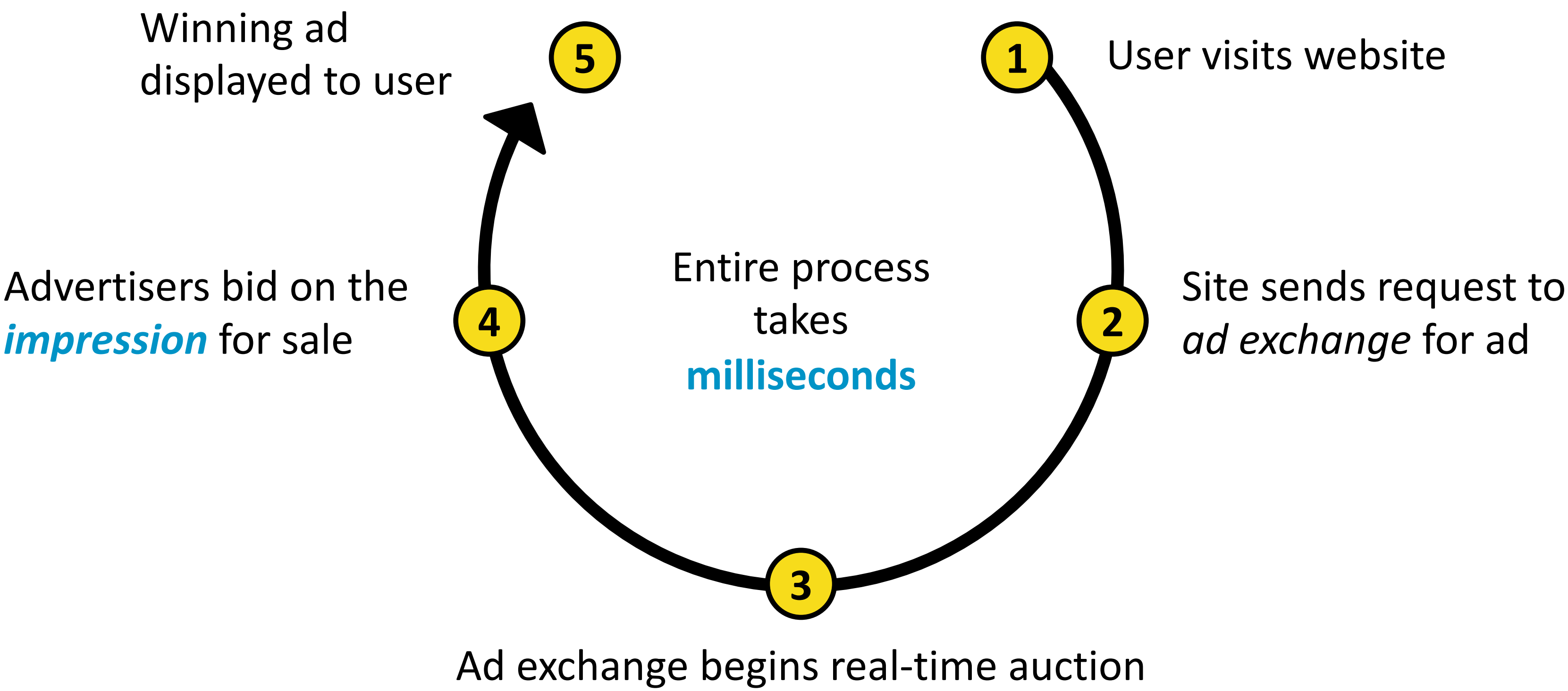
Outline

1. Mechanism design basics
2. Introduction to automated mechanism design (AMD)
3. Sample complexity guarantees for AMD
4. Additional AMD algorithms
    a) Empirical Myerson
    b) Ad auctions
5. Other learning models

	Ad rev. in 2016	Total rev. in 2016
Google	$79 B	$89.46 B
Facebook	$27 B	$27.64 B



Ad auction overview



Graphic based on: Bryant, Kathryn, et al. "Predicting Clicks in Mobile Advertising: An Experiment." *NYC Data Science Academy Blog*, 21 Dec. 2017.

Selection of ad auction prior work

- **Generalized second price auction** developed at Google.
  - Later studied by, e.g., Varian [International Journal of Industrial Organization '07], Edelman, Ostrovsky, and Schwarz [American Economic Review '07], and Aggarwal, Goel, and Motwani [EC'06]
- How to estimate **click through rates**?
  - [e.g., Pandey and Olston, NIPS'06]
- If **distribution over bidders' values known**, how to design high-revenue ad auctions
  - [e.g., Lahaie and Pennock, EC'07]
- **Reinforcement learning** for ad auctions
  - [E.g., Tang, IJCAI'17]

Batch learning for ad auctions

Key Challenge

There are **trillions** of different types of items (*impressions*)

Likely that a specific type of item has **never** been sold previously

Therefore, **no information** about its value is known

Practitioners represent each impression with a feature vector

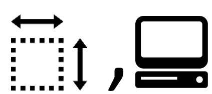
- E.g., ad size, whether it's mobile or desktop, etc.
- Assumption: bids from auctions with similar features will be similar

[Medina and Vassilvitskii, NIPS'17]

Batch learning for ad auctions: Setup

Distribution $\mathcal{D}$ over feature vector-value pairs $(\boldsymbol{z}, v) \in \mathcal{X} \times [0,1]$

– Auctioneer and buyer observe feature vector $\boldsymbol{z}$ describing impression

$\boldsymbol{z} = ($  $, \ldots)$

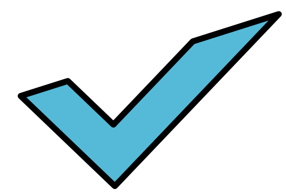
[Medina and Vassilvitskii, NIPS‘17]

Batch learning for ad auctions: Setup

Distribution $\mathcal{D}$ over feature vector-value pairs $(\boldsymbol{z}, v) \in \mathcal{X} \times [0,1]$

- Auctioneer and buyer observe feature vector $\boldsymbol{z}$ describing impression
- Auctioneer sets a price $p(\boldsymbol{z})$ for the impression
- Buyer has a value v for that impression
- Buyer buys the impression and pays $p(\boldsymbol{z})$ if and only if $v \geq p(\boldsymbol{z})$

$$p(\boldsymbol{z}) = p\left(\;, \;, \ldots \right) = \frac{1}{4}$$

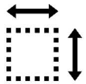


[Medina and Vassilvitskii, NIPS'17]

Batch learning for ad auctions: Setup

Distribution $\mathcal{D}$ over feature vector-value pairs $(\boldsymbol{z}, v) \in \mathcal{X} \times [0,1]$

- Auctioneer and buyer observe feature vector $\boldsymbol{z}$ describing impression
- Auctioneer sets a price $p(\boldsymbol{z})$ for the impression
- Buyer has a value v for that impression
- Buyer buys the impression and pays $p(\boldsymbol{z})$ if and only if $v \geq p(\boldsymbol{z})$

$p(\boldsymbol{z}) = p\big($  ,  $, \ldots \big) = \frac{3}{4}$ $v = \frac{1}{2}$ 

[Medina and Vassilvitskii, NIPS'17]

Batch learning for ad auctions: Setup

Distribution $\mathcal{D}$ over feature vector-value pairs $(\boldsymbol{z}, v) \in \mathcal{X} \times [0,1]$

- Auctioneer and buyer observe feature vector $\boldsymbol{z}$ describing impression
- Auctioneer sets a price $p(\boldsymbol{z})$ for the impression
- Buyer has a value v for that impression
- Buyer buys the impression and pays $p(\boldsymbol{z})$ if and only if $v \geq p(\boldsymbol{z})$

Goal: Learn good price function p, i.e., maximize **expected rev**:

$$\mathbb{E}_{(\boldsymbol{z},v)\sim\mathcal{D}}\left[p(\boldsymbol{z}) \cdot \mathbf{1}_{\{p(\boldsymbol{z})\leq v\}}\right].$$

[Medina and Vassilvitskii, NIPS'17]

Ad auctions: Key algorithmic ideas

Assumption

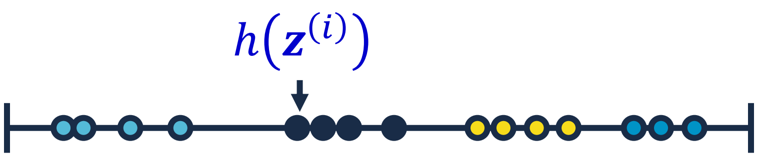
Bid prediction function $h(\boldsymbol{z})$ s.t. $\mathbb{E}_{(\boldsymbol{z},v)\sim\mathcal{D}}[(h(\boldsymbol{z})-v)^2]=\eta^2$

Algorithm RIC-$\boldsymbol{h}$ (Reserve Inference from Clusters) [Informal]

1. Input: $\left\{\left(\boldsymbol{z}^{(1)},v^{(1)}\right),\dots,\left(\boldsymbol{z}^{(N)},v^{(N)}\right)\right\}\sim\mathcal{D}^N$ and a parameter k
2. Cluster $\left\{h\left(\boldsymbol{z}^{(1)}\right),\dots,h\left(\boldsymbol{z}^{(N)}\right)\right\}$ into k clusters

   Corresponds to partition of $[0,1]$ into k intervals.
3. Set price per interval I:

   Best price on average over $\left\{v^{(j)}: h\left(\boldsymbol{z}^{(j)}\right)\in I\right\}$
4. Given vector $\boldsymbol{z}$, charge price assigned to interval $h(\boldsymbol{z})$ is in



[Medina and Vassilvitskii, NIPS'17]

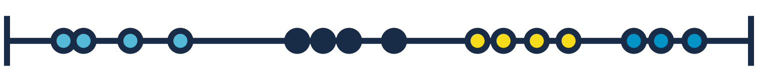
Batch learning for ad auctions: Guarantees

Theorem

Suppose we run RIC-h with N samples and set $k = N^{3/7}$.
Let M be the mechanism it returns. W.h.p.,

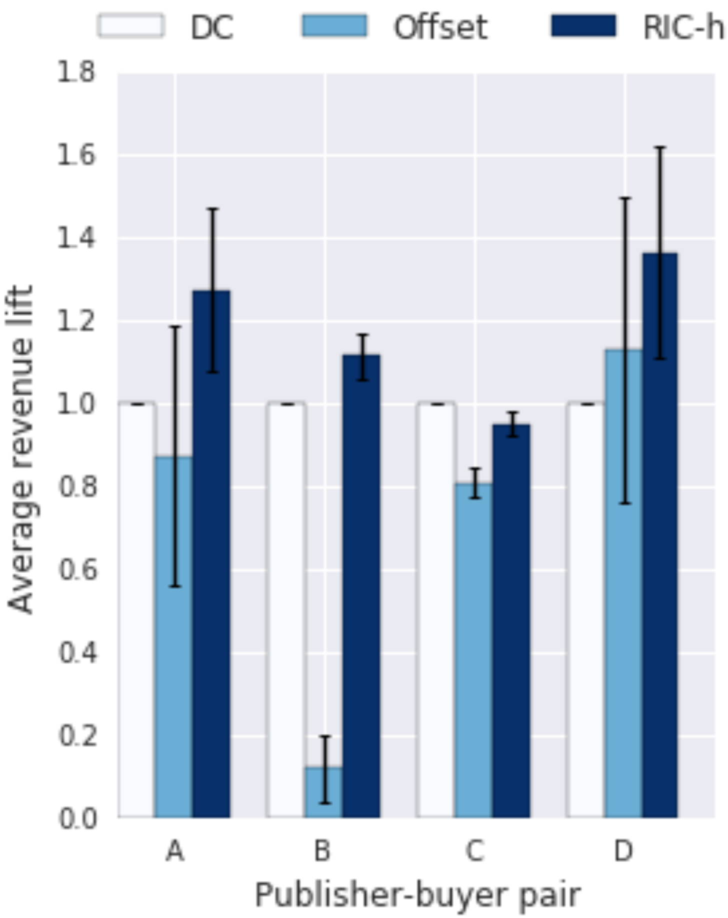
$$\text{Optimal revenue} - \mathbb{E}_{(\boldsymbol{z},v)\sim\mathcal{D}}[\text{revenue}_M(\boldsymbol{z},v)] \le \tilde{O}\left(\eta^{2/3} + \frac{1}{N^{2/7}}\right)$$

Accuracy of bid predictor h



[Medina and Vassilvitskii, NIPS‘17]

Batch learning for ad auctions: Experiments



- Experiments on AdExchange data.
- RIC-h achieves on average up to 30% improvement over the DC algorithm [Mohri and Medina, ICML'14].
- Offset is a simple algorithm also proposed by Medina and Vassilvitskii in this paper.

[Medina and Vassilvitskii, NIPS'17]

Outline

1. Mechanism design basics
2. Introduction to automated mechanism design (AMD)
3. Sample complexity guarantees for AMD
4. Additional AMD algorithms
5. Other learning models

Online learning

Day of the week	Price	Sold?
Monday	\$3	✓
Tuesday	\$4	✗
Wednesday	\$2	✗
Thursday	\$2.5	✓

How should seller set prices to minimize **regret**?

(Cumulative rev of best price in hindsight) – (seller's cumulative rev)

Online learning

Regret bounds for learning nearly optimal auctions:

- Same set of items sold again and again at each time step
- Compete with the best mechanism (highest revenue) in hindsight

Buyers' values **adversarial**	Buyers' values **stochastic**
E.g., Kleinberg and Leighton, FOCS'03; Blum, Kumar, Rudra, and Wu, SODA'03 Koren, Livni, and Mansour, COLT'17	E.g., Cesa-Bianchi, Gentile, and Mansour, IEEE Transactions on Information Theory, '15
Buyers behave truthfully	**Buyers behave strategically**
E.g., Dudík, Haghtalab, Luo, Schapire, Syrgkanis, and Vaughan, FOCS'17; **Balcan**, Dick, and **Vitercik**, FOCS'18	E.g., Amin, Rostamizadeh, and Syed, NIPS'14; Mohri and Medina, NIPS'14, '15; Drutsa, WWW'17, ICML'18

Online learning

Prophet inequalities and secretary problems

- Fixed set of items to sell
- **Prophet**: Buyers arrive in adversarial order and have random valuations [E.g., Hajiaghayi, Kleinberg, and **Sandholm**, AAAI'07; Kleinberg and Weinberg, STOC'12]
- **Secretary**: Buyers arrive in random order and have adversarial valuations [E.g., Hajiaghayi, Kleinberg, and Parkes, EC'04; Kleinberg, SODA '05]

Learning to bid

- At beginning, buyer knows nothing about his values for items
- At each round, learns values only for the items he wins
- Goal is to compete with the best fixed bids in hindsight
- [E.g., Weed, Perchet, and Rigollet, COLT'16; Feng, Podimata, and Syrgkanis, EC'18]

Future directions

- Optimization algorithms for automated mechanism design
  - Structural revenue maximization gives guarantees for optimizing over a nested hierarchy, but how do we actually optimize?
- Beyond-worst-case complexity bounds: sample complexity bounds that adapt to the "niceness" of the distribution
  - E.g., using Rademacher complexity [Mohri and Medina, ICML'14; Syrgkanis, NIPS'17; Balcan, Sandholm, and Vitercik, NIPS'16, EC'18] and covering-style analyses [E.g., Cai and Daskalakis, FOCS'17 ; Balcan, Sandholm, and Vitercik, EC'19]
- Other objectives beyond revenue maximization
- Other AMD applications beyond selling