Mechanism Design
&
Automated Mechanism Design

Tuomas Sandholm

Outline

1. Introduction
2. Mechanism design basics
   a) Single-item
   b) Multi-item
3. Automated mechanism design (AMD)
4. Sample complexity guarantees for automated mechanism design

Mechanism design

Field of game theory with significant real-world impact.

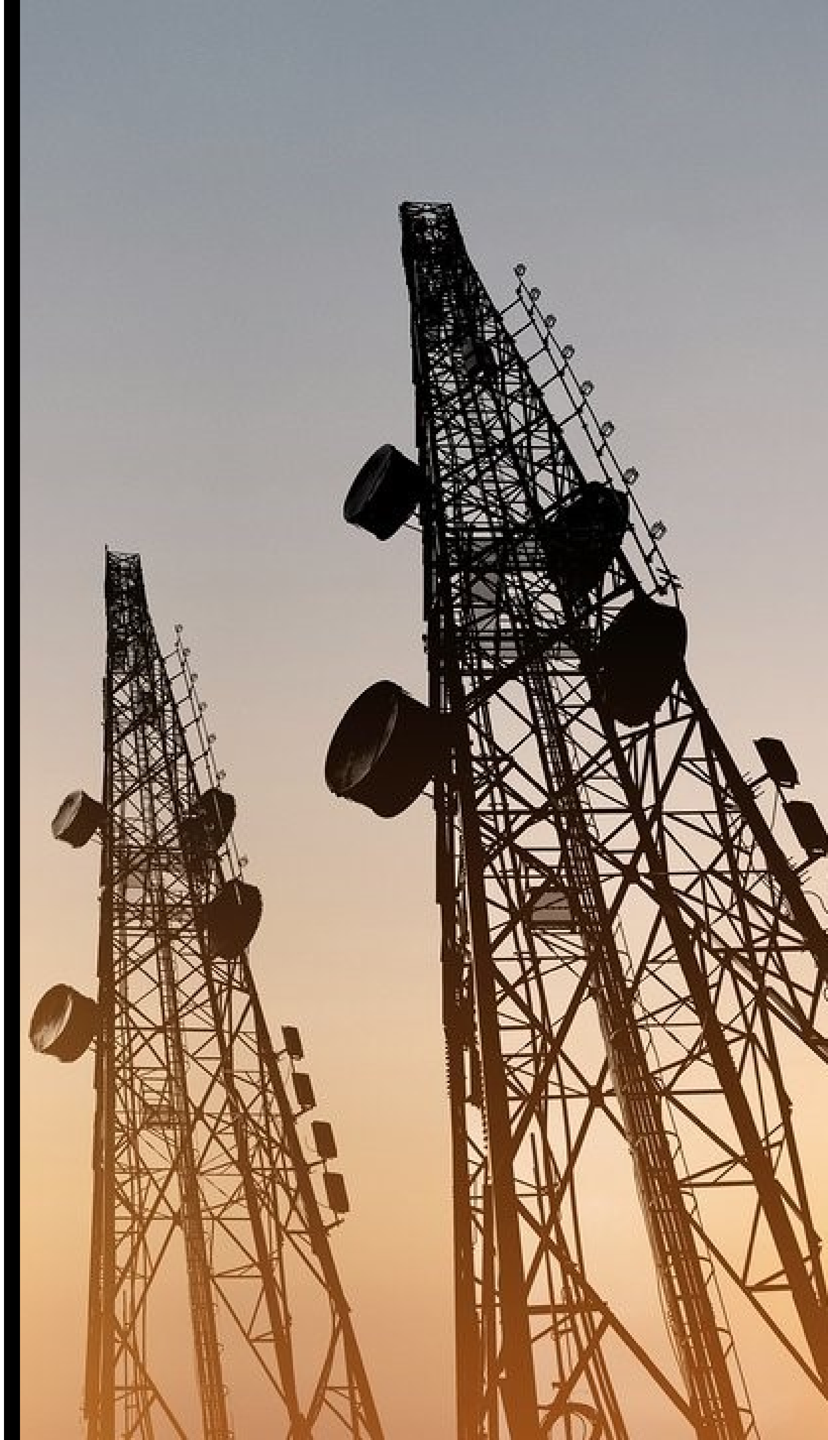
Encompasses areas such as pricing and auction design.

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

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

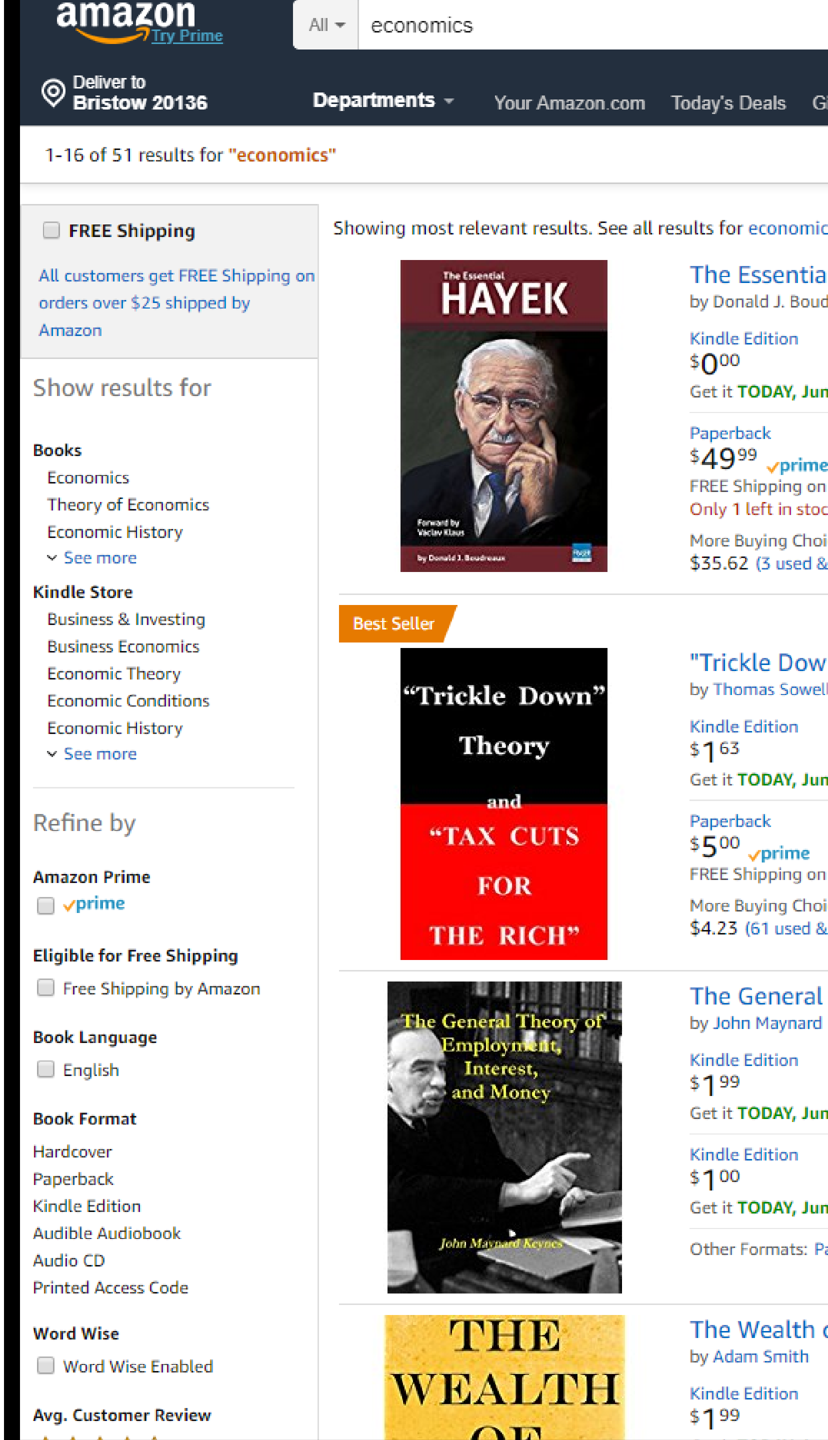
Bidding in government auction of airwaves reaches $34B

[NYTimes ‘14]

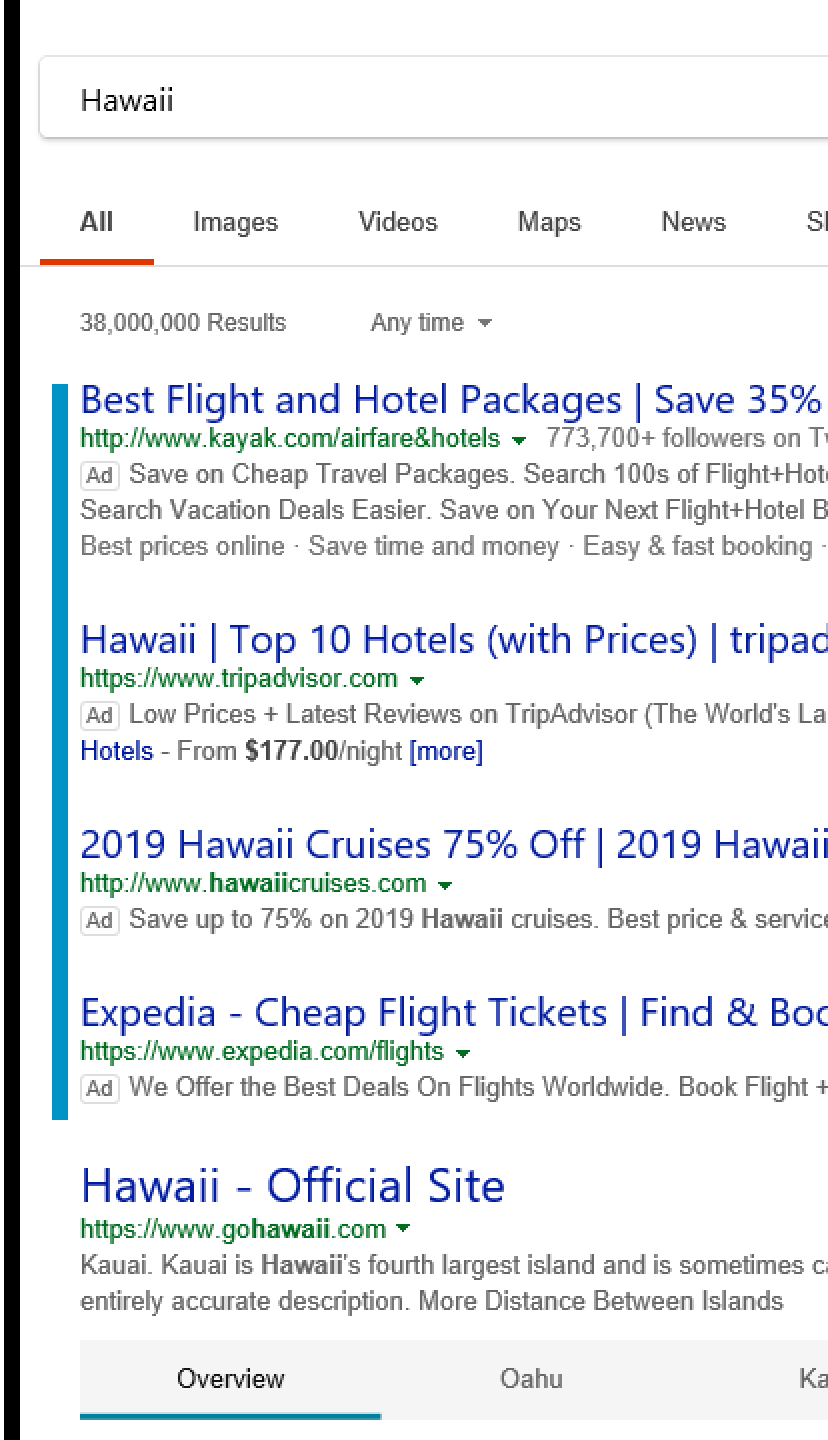


Amazon’s profit swells to $1.6B

[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]

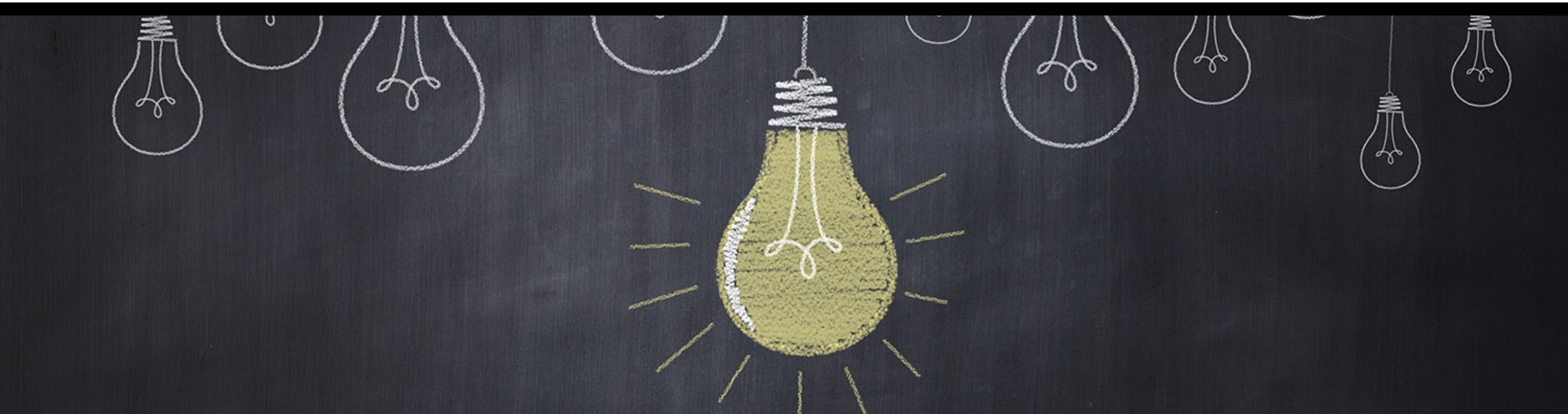
Use optimization, ML, & data to design mechanisms

- Helps overcome challenges faced by manual approaches:

  2 items for sale: Revenue-maximizing mechanism unknown

In these two lectures, we:

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



Outline

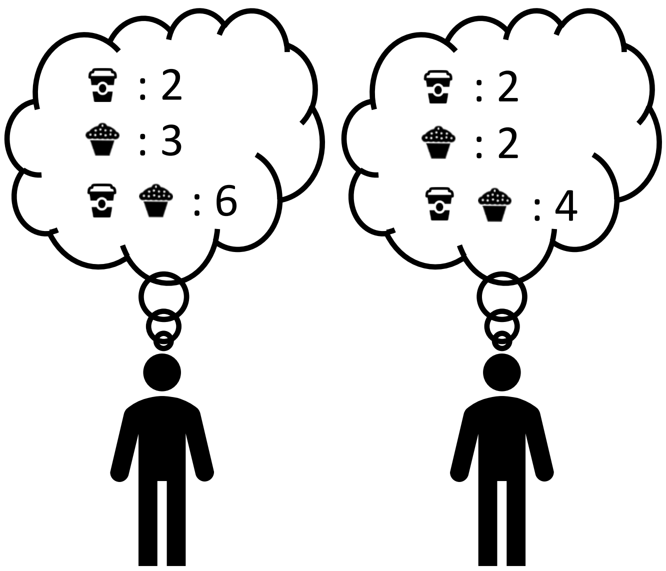
1. Introduction
2. Mechanism design basics
3. Automated mechanism design (AMD)
4. Sample complexity guarantees for AMD

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

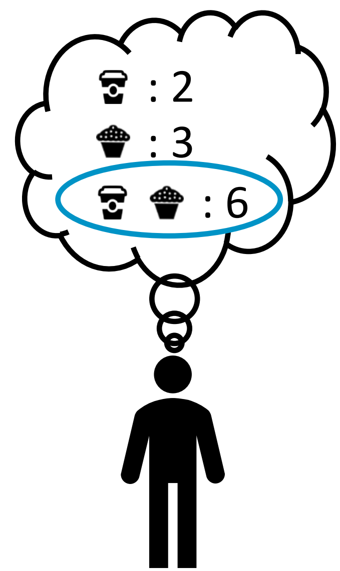


Mechanism design example: Posted price mechanisms

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 r

Highest bidder wins if bid $\geq r$

Pays maximum of second highest bid and r

Reserve price: \$8 ⇨ Revenue = \$8

Reserve price: \$6 ⇨ Revenue = \$7

Second-price auction

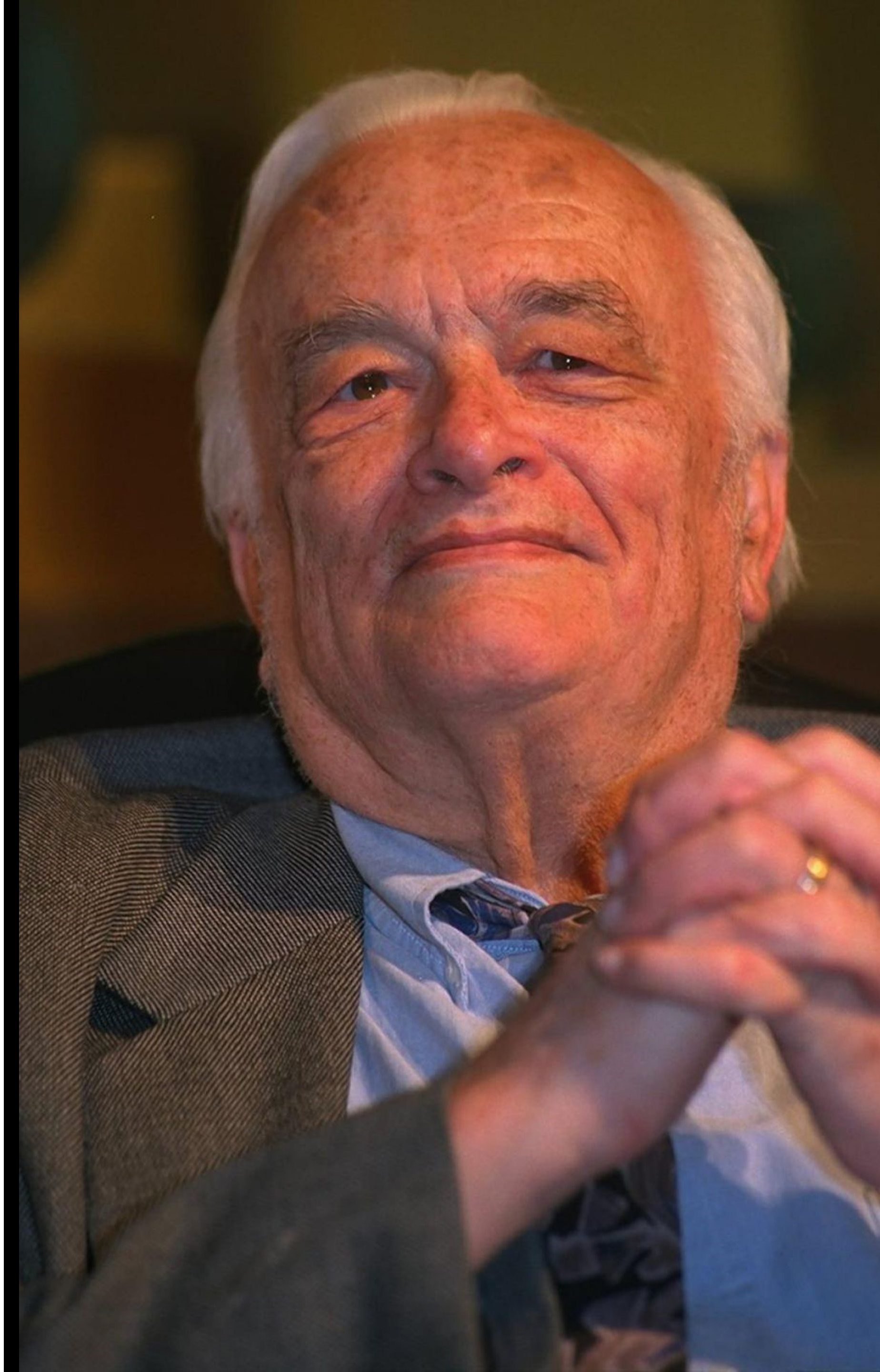
1961: Introduced by Vickrey

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

Buyer i's "type"

Example

Items = {  ,  }

v_i(∅) = 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 $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ is sum of payments:

$$\text{revenue}_M(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$$

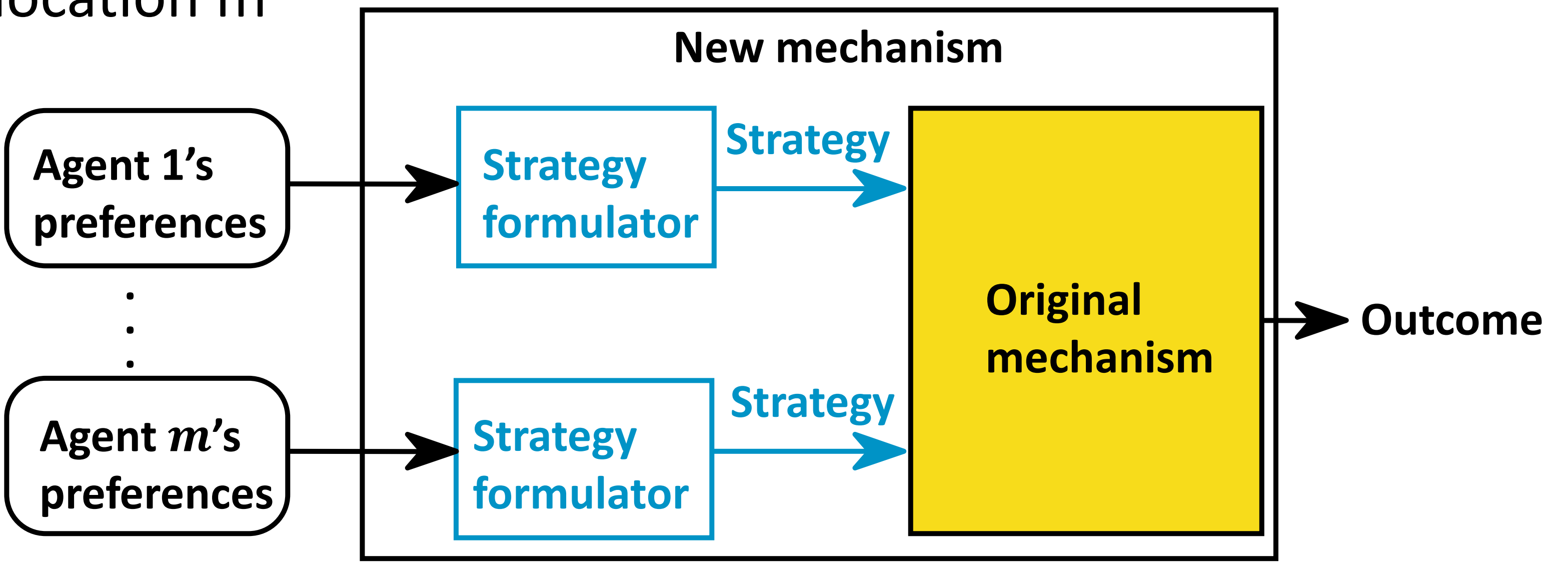
Sometimes, each buyer i might need to submit a set of bids:

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

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

Why can we restrict attention to single-shot IC mechanisms?

Revelation principle (informal): If some allocation and payment fns are implementable by a mechanism, then there's a single-shot incentive compatible mechanism with same payment and allocation fn



Mechanism lies for the agents!

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.*

(value() - payment) • 1 (wins item)

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.*

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

Why not bid **below** 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( )?

– 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.*

(value() - payment) • 1 (wins item)

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

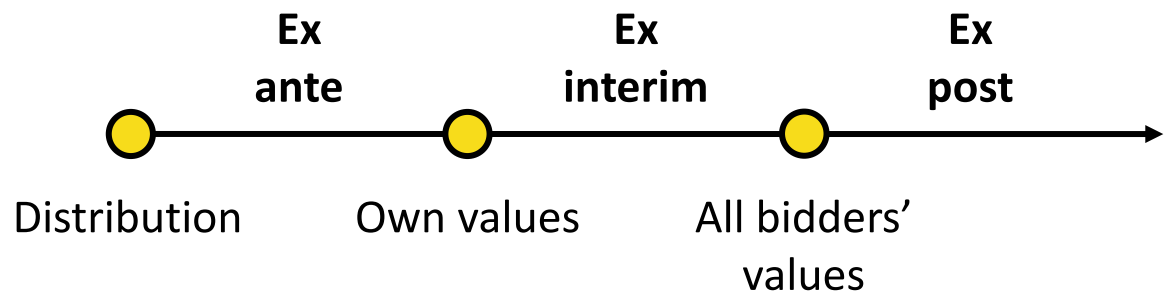
Buyers' values are drawn from a probability **distribution**.

Example

$(\boldsymbol{v}_1, \dots, \boldsymbol{v}_n) \sim \mathcal{D}$, where $\boldsymbol{v}_i = \left[v_i(\emptyset), v_i(\;), v_i(\;), v_i(\;,\;) \right]$

Different types of incentive compatibility

Each bidder knows:

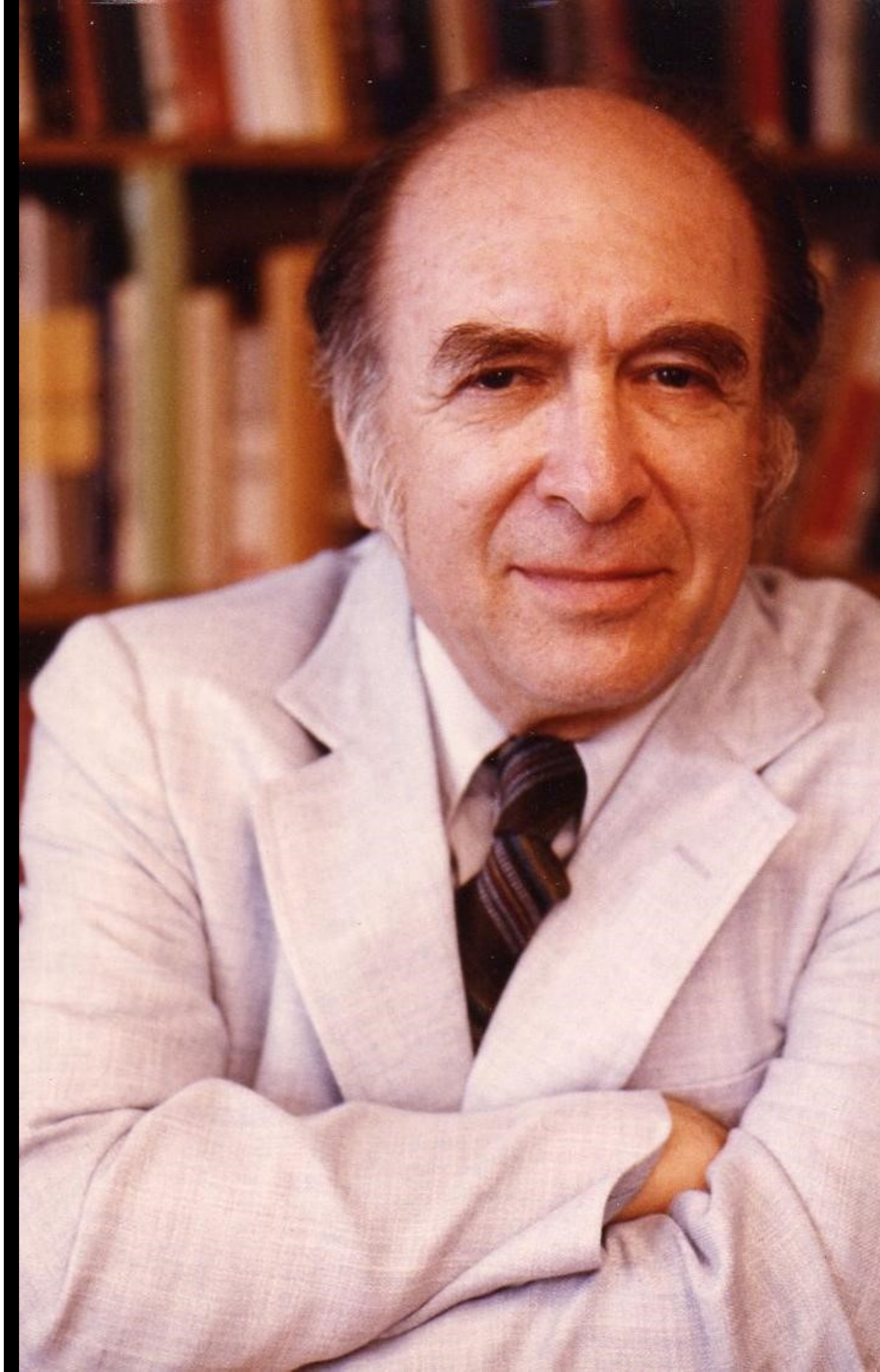


Incentive compatibility

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.



Optimal single-item sales mechanism

1981: Myerson discovered "optimal" 1-item auction

Revenue-maximizing

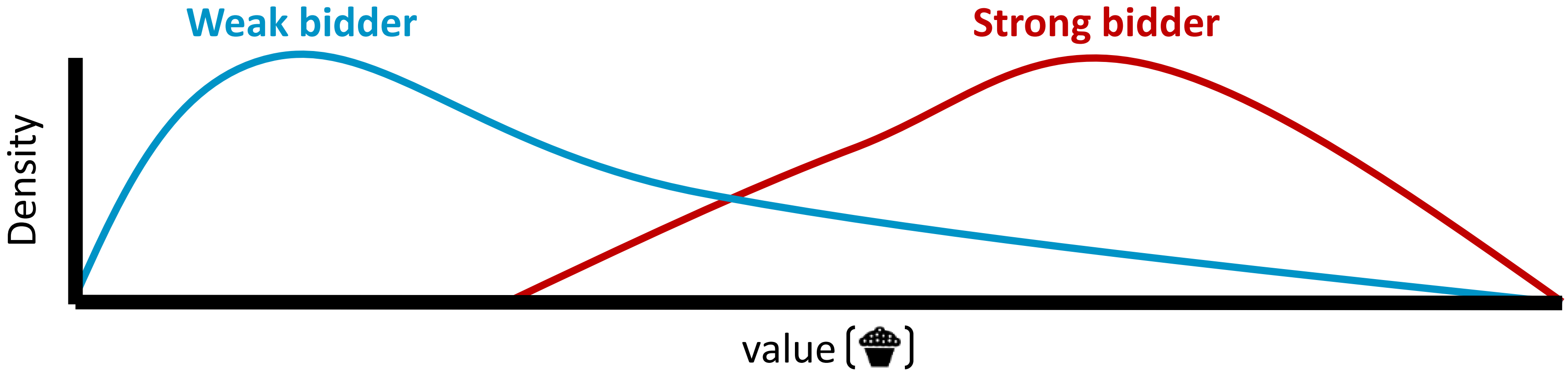
2007: Won Nobel prize

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. Introduction
2. Mechanism design basics
3. Automated mechanism design (AMD)
4. Sample complexity guarantees for AMD

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

---

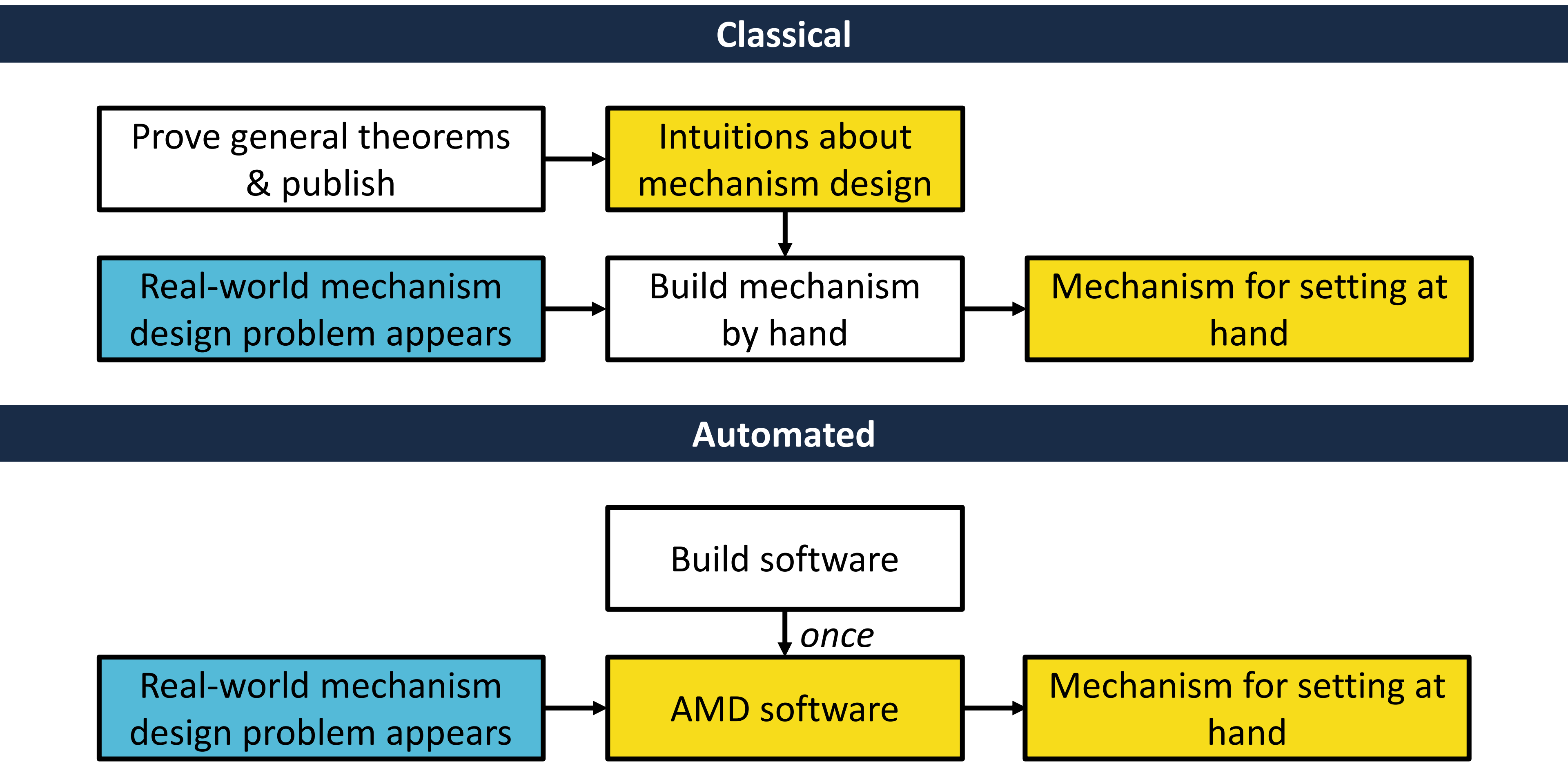
- Can lead to greater value of designer’s objective than known mechanisms
- Sometimes circumvents economic impossibility results
  - Always minimizes the pain implied by them
- Can be used in new settings & for unusual objectives
- Can yield stronger incentive compatibility & participation properties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

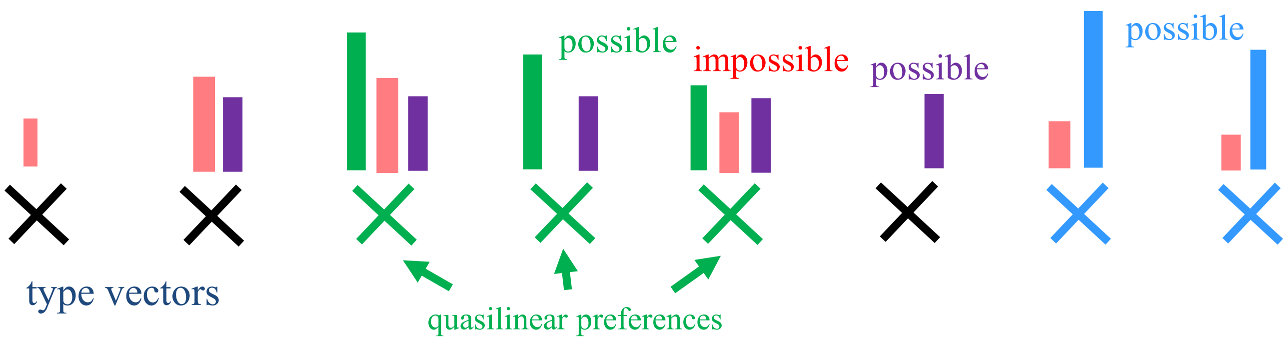


Input

- 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**: $k\hat{}(2^{\#items})$
  - k is the number of possible values a buyer might have for a bundle

Classes of automated mechanism design

1. "Flat-representation" *de novo* design
2. Search in a parametric mechanism class
3. Incremental automated mechanism design

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

[Likhodedov & Sandholm AAAI-04, AAAI-05, Operations Research 2015]

- 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; Balcan, Blum, Hartline, and Mansour, FOCS'05; Morgenstern and Roughgarden, COLT'16; Syrgkanis, NIPS'17; Cai and Daskalakis, FOCS'17; Gonczarowski and Weinberg, FOCS'18…

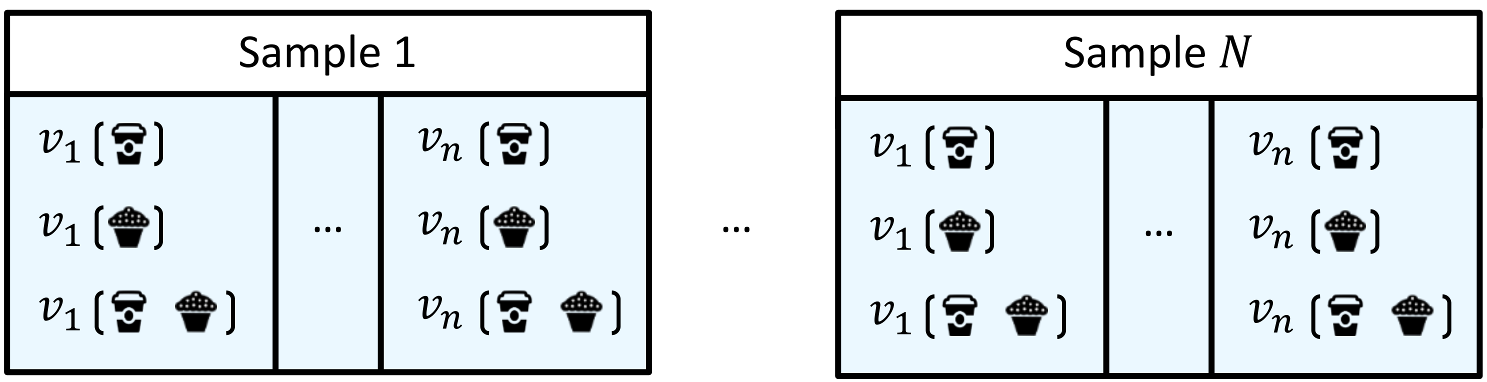
Single-item

E.g., 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.

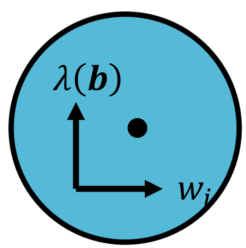


Two key ideas to get scalability and avoid the need to discretize type space

[Likhodedov & S., AAAI-04, AAAI-05; S. & Likhodedov, *Operations Research*-15]

- Don't assume that the distribution over bidders' valuations is given, only **samples** from it
  - Now an active research field in TCS & AI

- Automated mechanism design 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 $SW(\cdot)$

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

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 $SW(\cdot)$

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

$\boldsymbol{b}^{-i} = (b_1^{-i}, \dots, b_n^{-i})$ 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 $(b_1, \dots, b_n)$ 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 $SW(\cdot)$

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

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

Allocation: $\boldsymbol{b}^*$

The social-welfare-maximizing allocation.

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

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

Bidder	**1**	**2**
	1	0
	2	1
 	2.5	1

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

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

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

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

Bidder	1	2
	1	0
	2	1
	2.5	1

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

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

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

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

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:

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

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

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

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

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

$$\left[\left(\sum_{j \in \text{Bidders}-\{i\}} v_j\left(b_j^{-i}\right)\right) - \left(\sum_{j \in \text{Bidders}-\{i\}} v_j\left(b_j^*\right)\right)\right]$$

- 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: $\boldsymbol{w_i}$

$\boldsymbol{\lambda}(b_1, \ldots, b_n)$ 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:

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

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

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

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

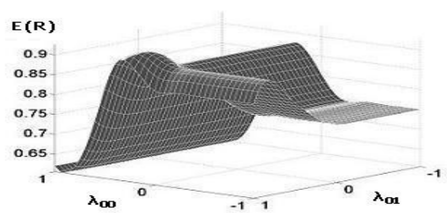
(*How much happier everyone would be if bidder i hadn't participated*):

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

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 $\lambda(b_1, \dots, b_n) = 0$ for all allocations $(b_1, \dots, b_n)$).
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

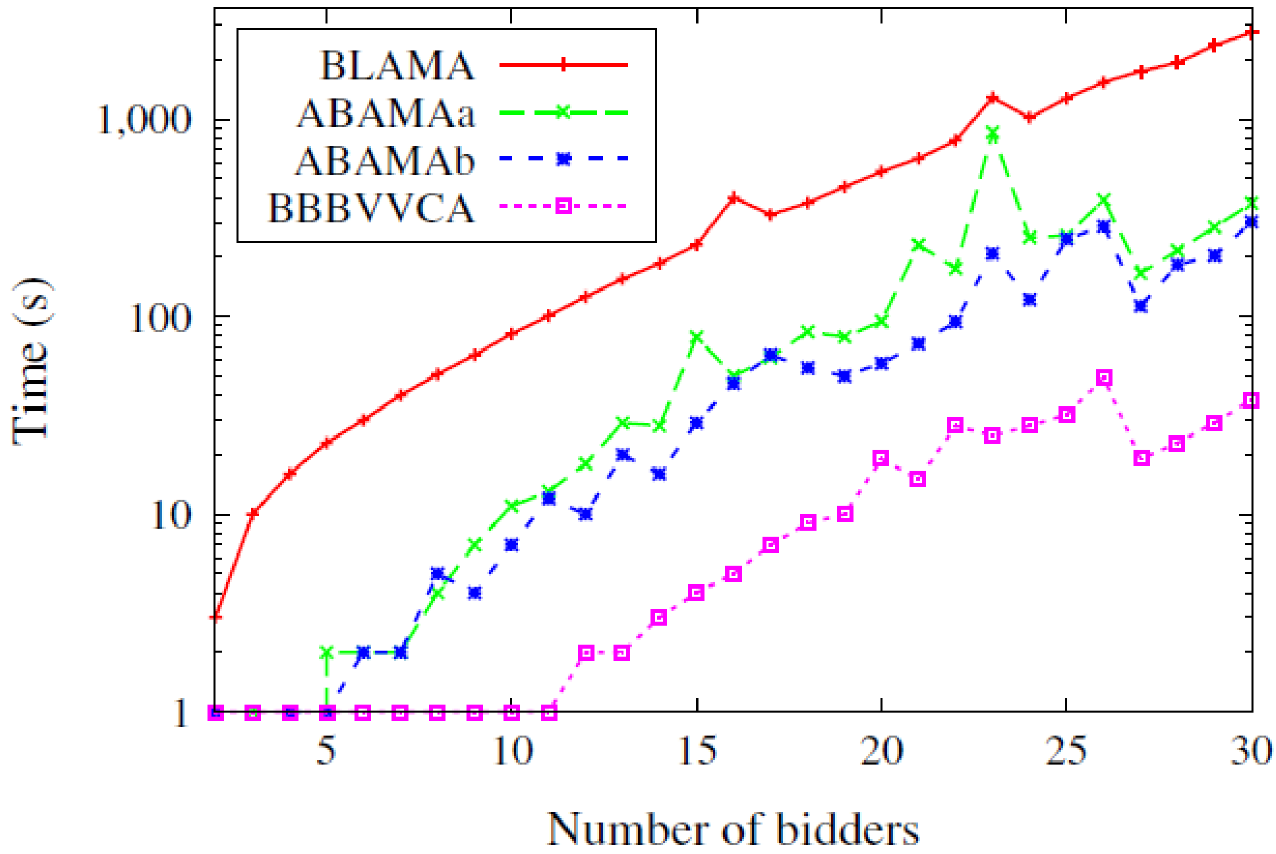
	Setting 1	**Setting 2**	**Setting 3**
$\boldsymbol{f_1}$	U[0, 1]	U[1, 2]	U[1, 2]
$\boldsymbol{f_2}$	U[0, 1]	U[1, 2]	U[1, 5]
$\boldsymbol{f_c}$	0	U[-1, 1]	U[-1, 1]
VCG	2/3	2.45	2.85
AMA*	+32%	+14%	+48%
VVCA*	+31%	+13%	+47%
BLAMA	+17%	+13%	+31%
ABAMA	+17%	+13%	+32%
BBBVVCA	+18%	+14%	+30%

Table shows the revenue lift of various mechanisms over VCG

[Sandholm and Likhodedov, OR'15]

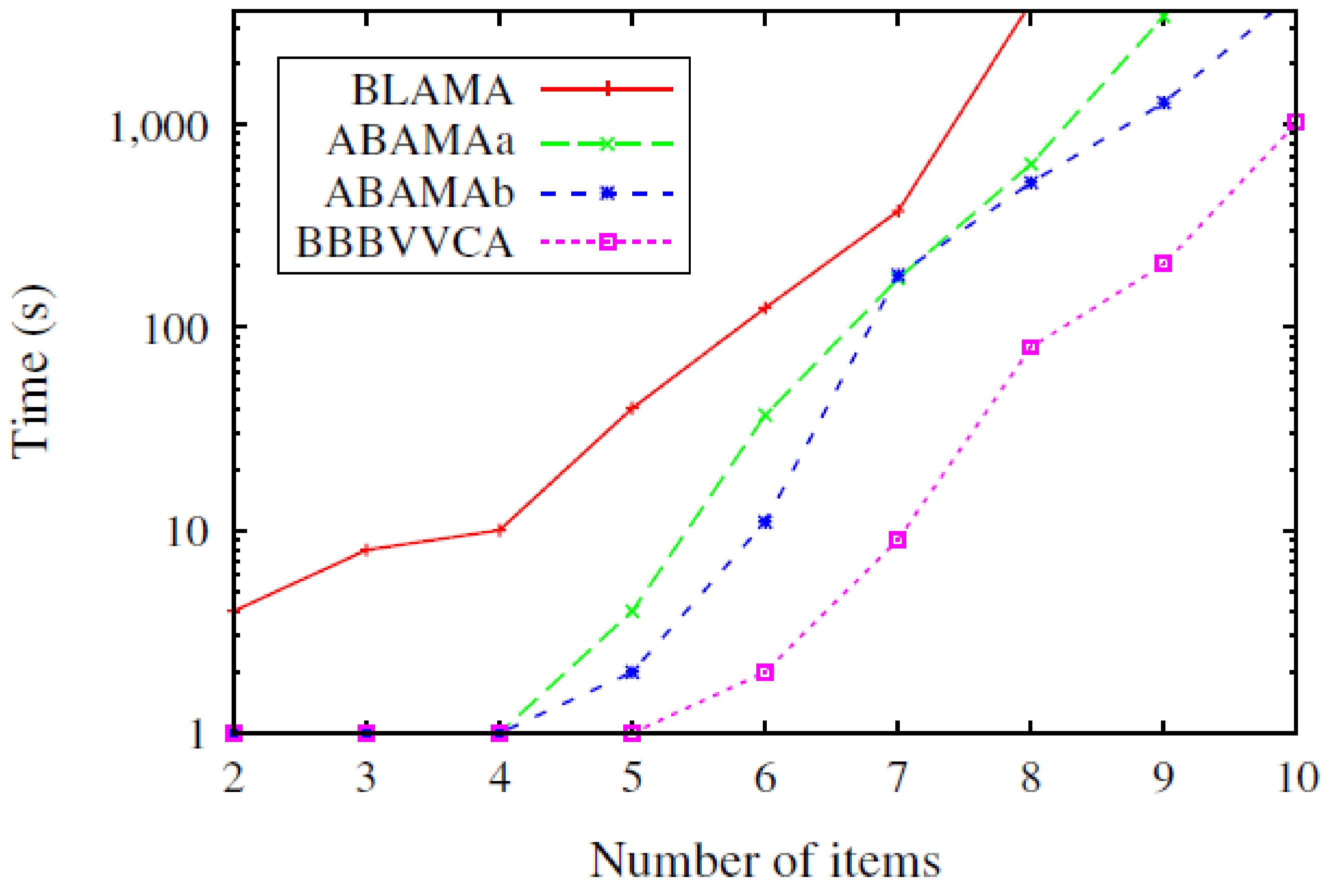
In Setting I, generalizing the mechanism design from MBARPs to VVCAs doesn't yield additional revenue, but generalizing further to AMAs does.

Scalability experiments
(3 items, symmetric distribution)



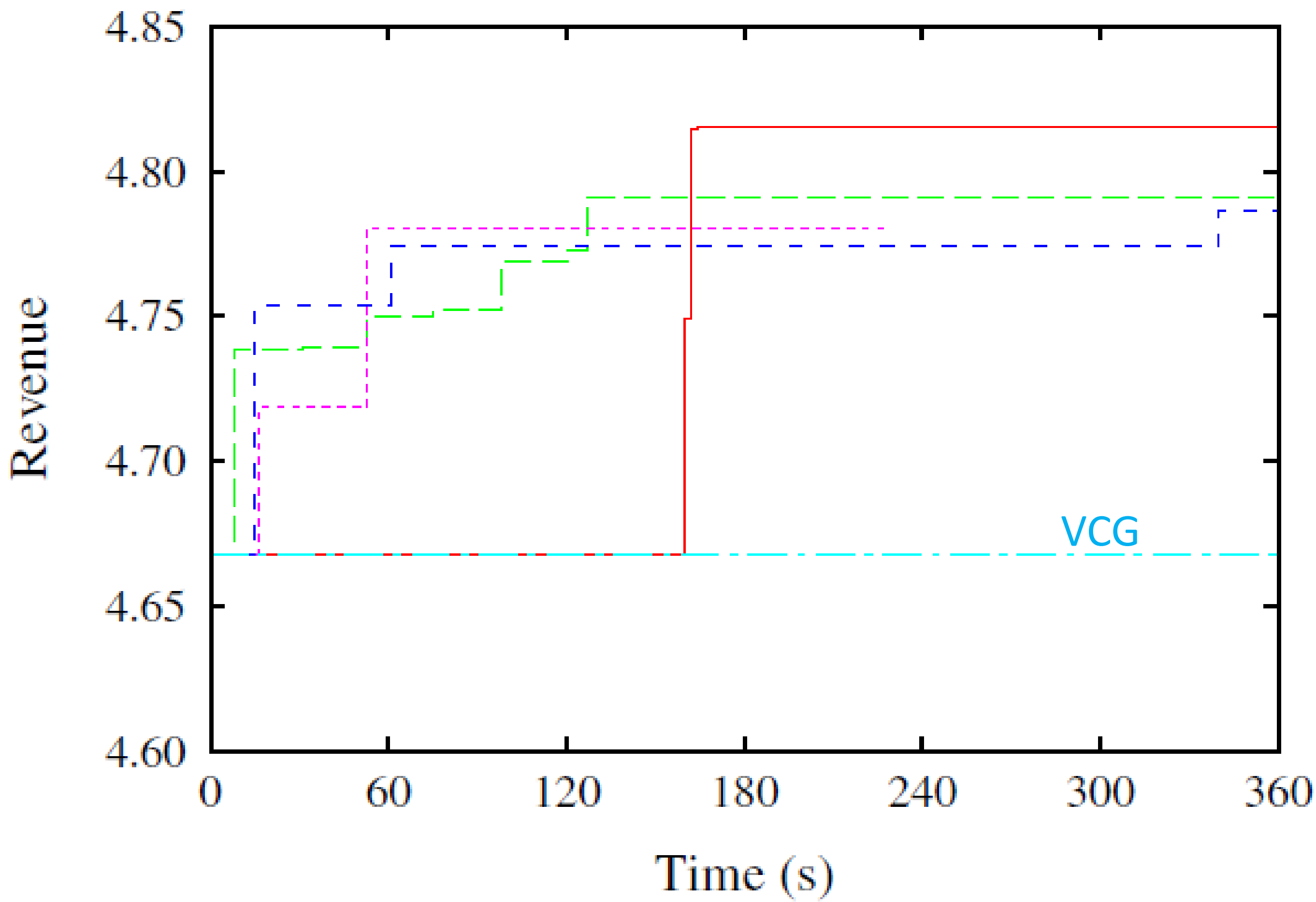
[Sandholm and Likhodedov, OR'15]

Scalability experiments
(3 bidders, symmetric distribution)



[Sandholm and Likhodedov, OR'15]

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



[Sandholm and Likhodedov, OR'15]

Classes of automated mechanism design

1. "Flat-representation" *de novo* design
2. Search in a parametric mechanism class
3. Incremental automated mechanism design

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 that maximizes 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
- *Designs **plurality-with-runoff** voting rule*
  - In that voting rule, if no candidate gets more than 50% of the vote, simulate a second election between the 2 candidates with the most votes in the first round

Incremental AMD via deep learning

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

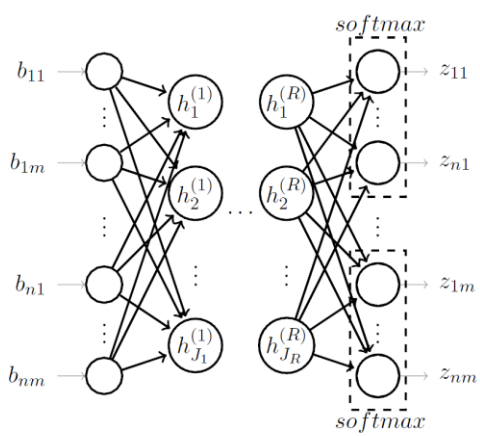
m items, n additive bidders

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

Parameters w

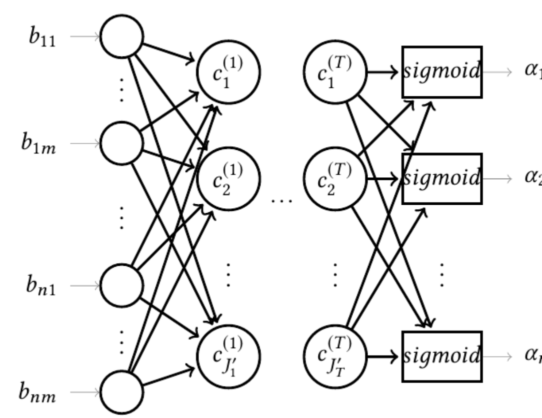
Feedback: Revenue and bidders' regret

Allocation Net



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

Payment Net



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

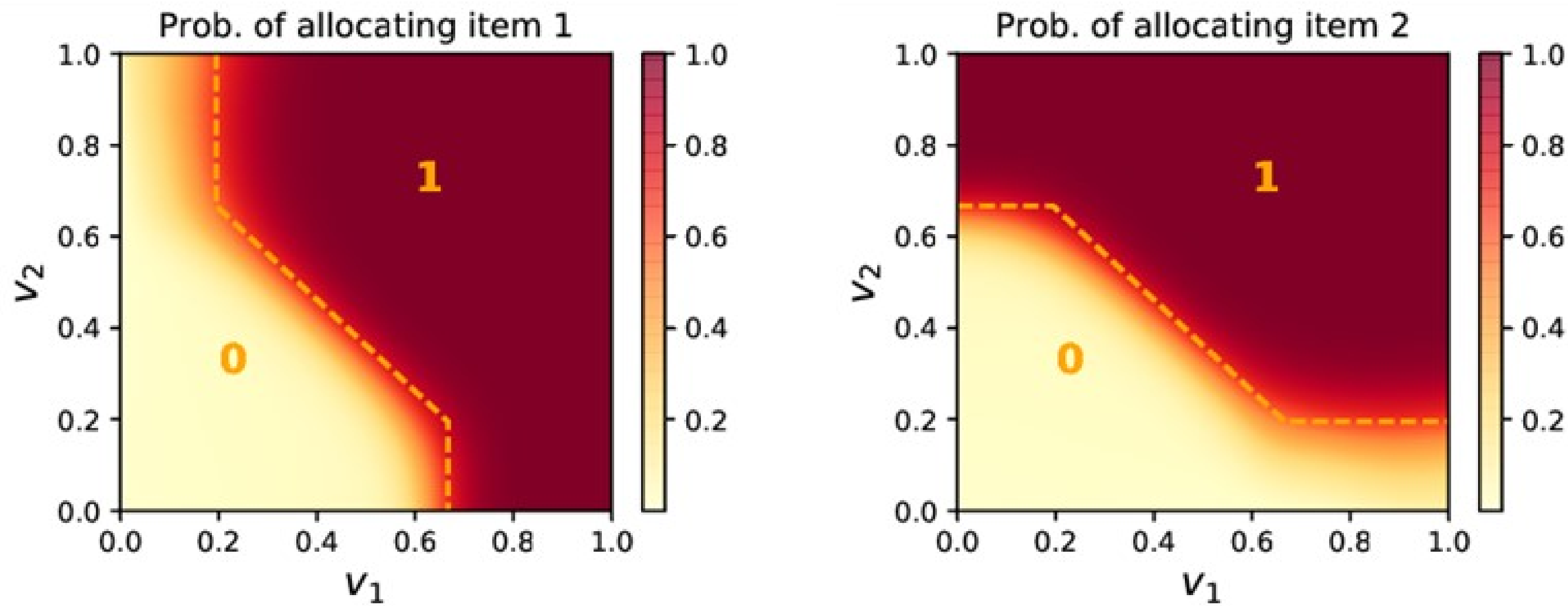
Fractional payment:

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

(Guarantees IR)

Incremental AMD via deep learning

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



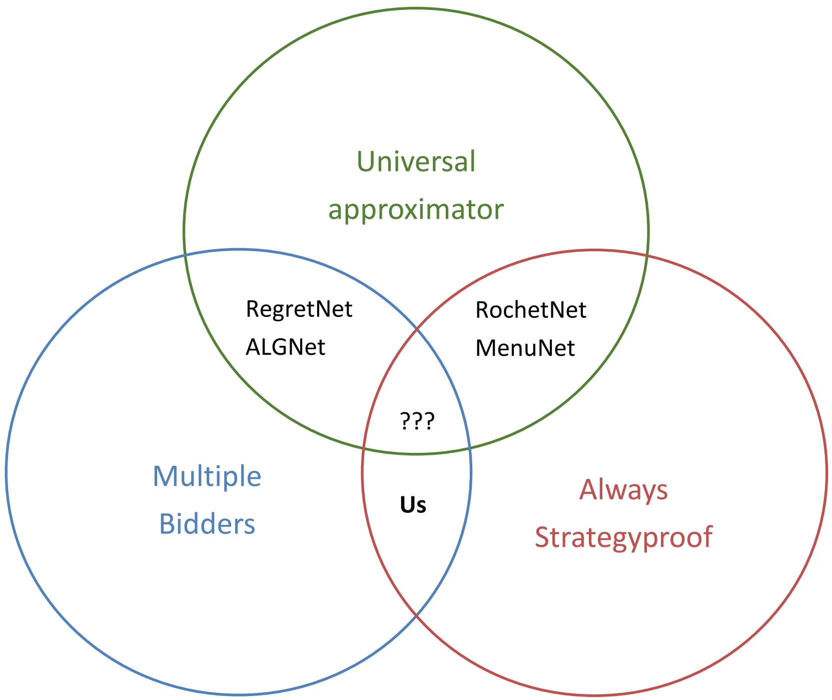
Solid regions: Learned allocation probability when single bidder with $v_1, v_2 \sim U[0,1]$

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

Our new architecture: **Differentiable economics for randomized affine maximizer auctions**

[Curry, Sandholm & Dickerson, arXiv-22]

- A strategyproof multiagent, multi-item architecture
- Modification of affine maximizer auctions
- New in our work: learn all parameters, including offered allocations, end-to-end
  - This additionally allows the offered allocations to be *lotteries*



Differentiable end-to-end

- AMA procedure describes forward pass at *test time*
- At *train time*, replace max and argmax operations with soft versions
- Compute gradients of learned parameters **including the allocations** with respect to objective (revenue) and optimize
- Contrast with RegretNet: objective is simply revenue (no Lagrangian regret terms)

Experimental results

- 2x2 iid uniform auction:

Auction	Best Revenue	Regret
Lottery AMA (ours)	0.868	0
Combinatorial AMA	0.862	0
Separate Myerson	0.833	0
Grand Bundle	0.839	0
MBARP	0.871	0
RegretNet	0.878	< 0.001
ALGNet	0.879	0.00058

- A larger setting (3x10):

Auction	Best Revenue	Regret
Lottery AMA (ours)	5.345	0
Separate Myerson	5.31	0
Grand bundle	5.009	0
RegretNet	5.541	0.002
ALGNet	5.562	0.002

- Learned auctions are sparse (2048 allocations allowed, only 10 used at end)
- Randomized version yields dramatically better revenue than deterministic version (e.g., 2.158 vs. 1.462)

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

(Recall optimal transport problem: Move one mass to another, minimizing cost)

Dual is tight

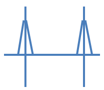
Consequences:

- In that setting, every optimal auction has a certificate in form of transportation flow
  - Can 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

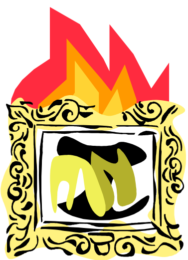
Automated mechanism design in sponsored search auctions

- Generalized second price auction was 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

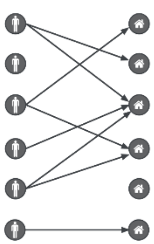



- Combinatorial public goods problems [Conitzer and Sandholm, UAI'03 Bayesian Modeling Applications Workshop]



- 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

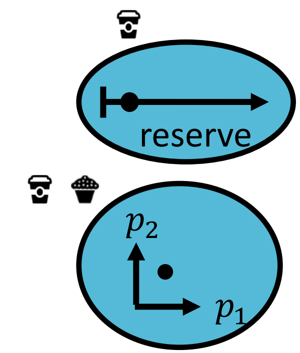
1. Introduction
2. Mechanism design basics
3. Automated mechanism design (AMD)
4. Sample complexity guarantees for AMD

Note: There's been a lot of recent work on batch learning for AMD. We focus on that.

Mechanism design as a learning problem

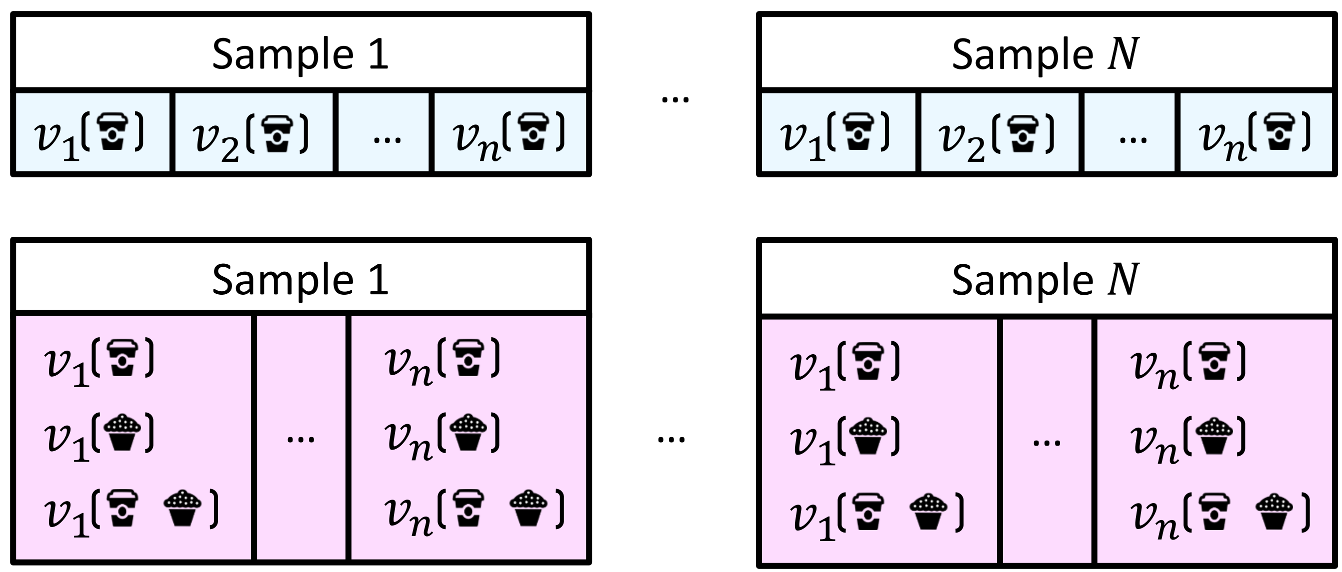
Goal: Given family of mechanisms $\mathcal{M}$ and set of buyers' values sampled from unknown distr. $\mathcal{D}$, find mechanism with high expected revenue.

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



2^{nd} price auctions with reserves:

Posted price mechanisms:

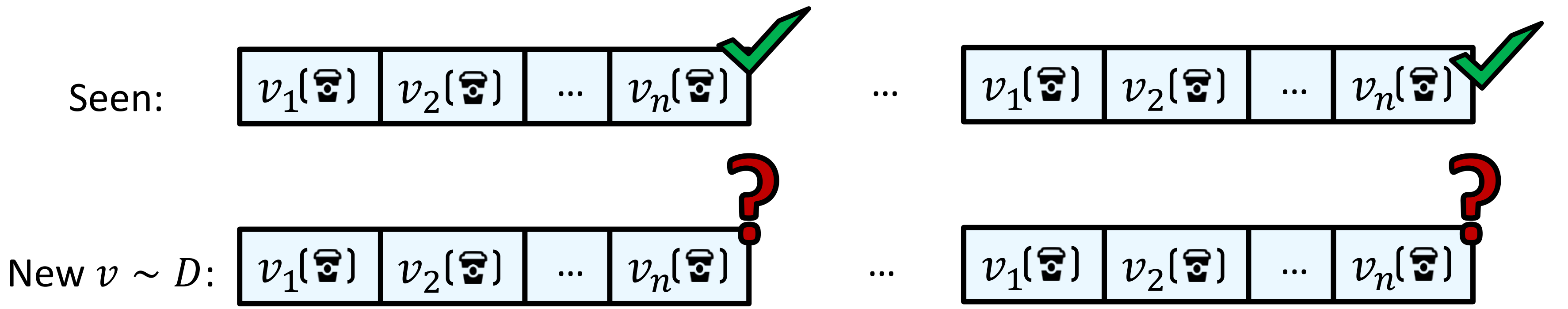


Mechanism design as a learning problem

Goal: Given family of mechanisms $\mathcal{M}$ and set of buyers' values sampled from unknown distr. $\mathcal{D}$, 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?



Will $\widehat{\mathrm{M}}$ have high revenue over $\mathcal{D}$?

Mechanism design as a learning problem

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

Approach: Find $\widehat{M}$ (nearly) optimal mechanism over the set of samples. Will $\widehat{M}$ have high expected revenue?

Key technical tool: uniform convergence, for any mechanism in class $\mathcal{M}$, average revenue over samples "close" to its expected revenue.

Mechanism design as a learning problem

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

Approach: Find $\widehat{M}$ (nearly) optimal mechanism over the set of samples.
Will $\widehat{M}$ have high expected revenue?

Key technical tool: uniform convergence, for any mechanism in class $\mathcal{M}$, average revenue over samples "close" to its expected revenue.

- Imply that $\widehat{M}$ have high expected revenue.

Mechanism design as a learning problem

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

Approach: Find $\widehat{M}$ (nearly) optimal mechanism over the set of samples. Will $\widehat{M}$ have high expected revenue?

Key technical tool: uniform convergence, for any mechanism in class $\mathcal{M}$, average revenue over samples "close" to its expected revenue.

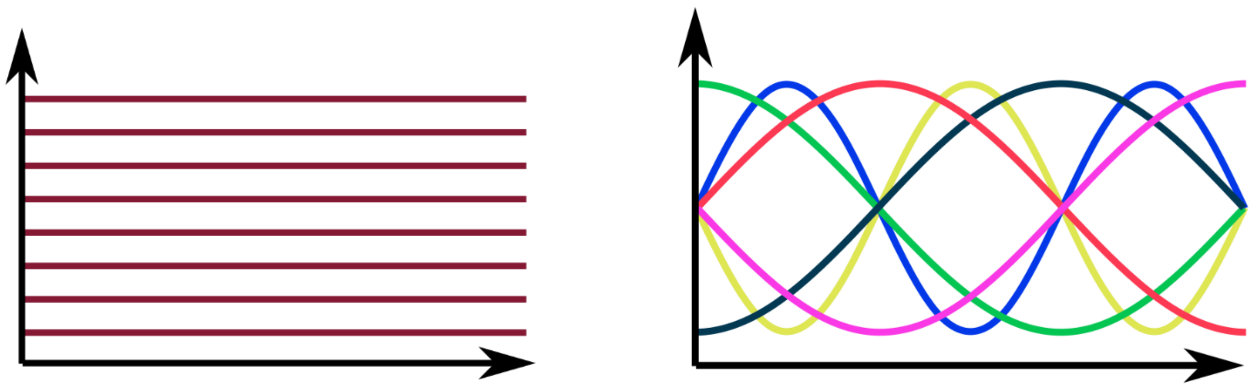
Learning theory: $\mathrm{N} = \mathrm{O}(\dim(\mathcal{M})\,/\epsilon^2)$ instances suffice for ϵ-close

Mechanism design as a learning problem

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

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

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

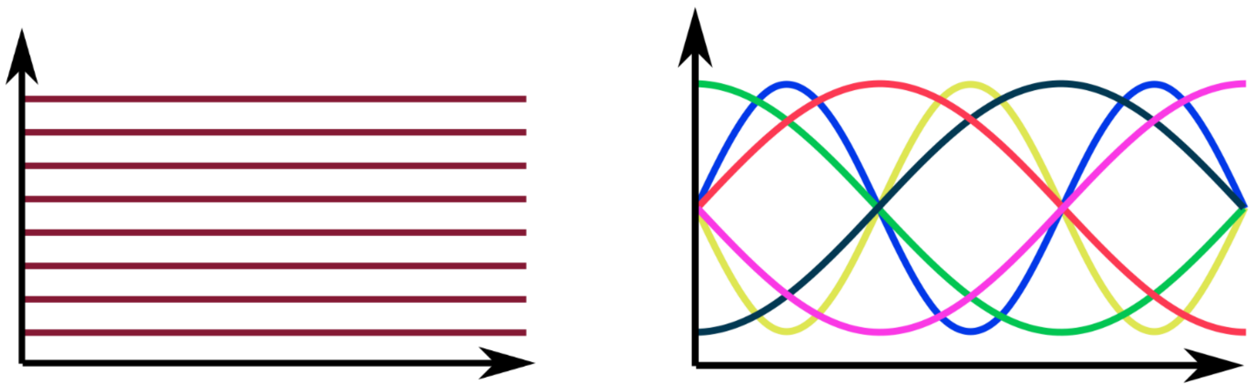


Mechanism design as a learning problem

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

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

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



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

Uniform Convergence of Auctions

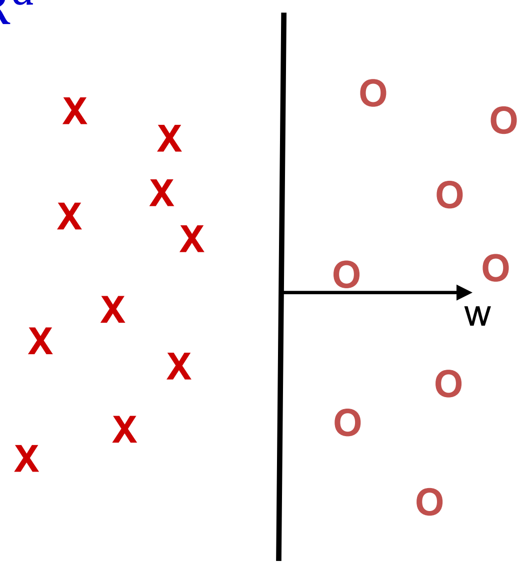
- Digital goods (unrestricted supply): Balcan, Blum, Hartline, and Mansour [FOCS'05] were first to use **learning-theoretic tools** to design and analyze auctions.
- Mohri and Medina [ICML'14] use a combination of **pseudo-dimension** and **Rademacher complexity** to analyze second-price auctions with 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.

Brief tour of VC theory

VC-dimension [Vapnik-Chervonenkis, 1971]

VC-dimension: complexity measure that characterizes the sample complexity of binary-valued function classes.

E.g., H= Linear separators in R^d



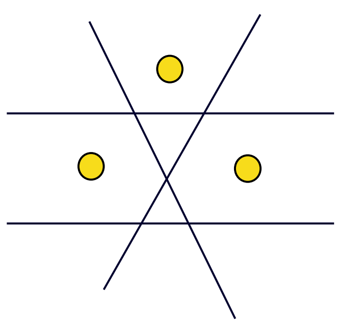
VC-dimension [Vapnik-Chervonenkis, 1971]

VC-dimension of a function class H is the cardinality of the largest set S that can be labeled in all possible ways $2^{|S|}$ by H.

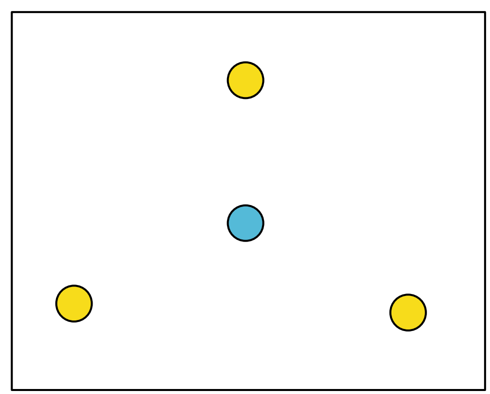
[If arbitrarily large finite sets can be shattered by H, then $\mathrm{VCdim(H)} = \infty$]

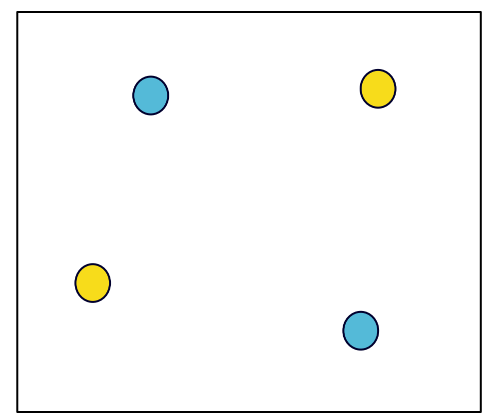
E.g., *H*= linear separators in $\mathbf{R}^2$ $\quad \mathrm{VCdim(H)} = 3$

$\mathrm{VCdim(H)} \geq 3$



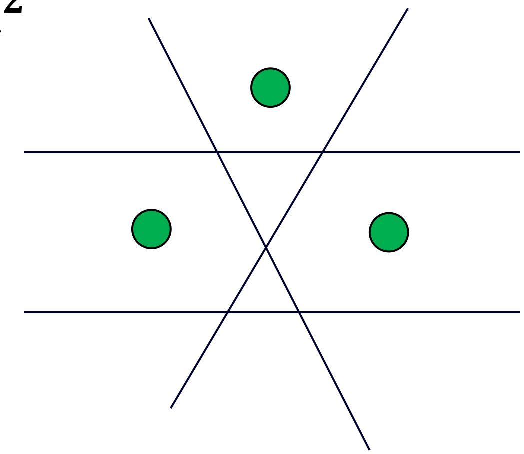
$\mathrm{VCdim(H)} < 4$





Example: VC-dimension of linear separators

E.g., *H*= linear separators in $\mathbf{R}^2$

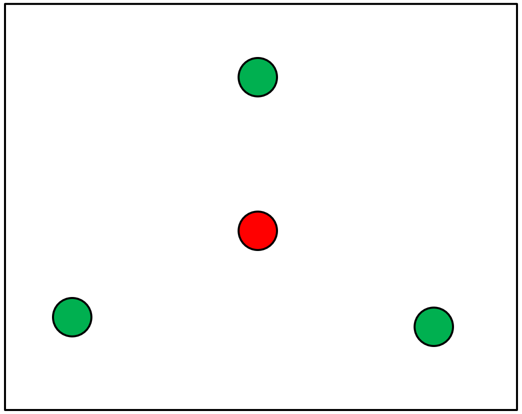


$\mathrm{VCdim}(\mathrm{H}) \geq 3$

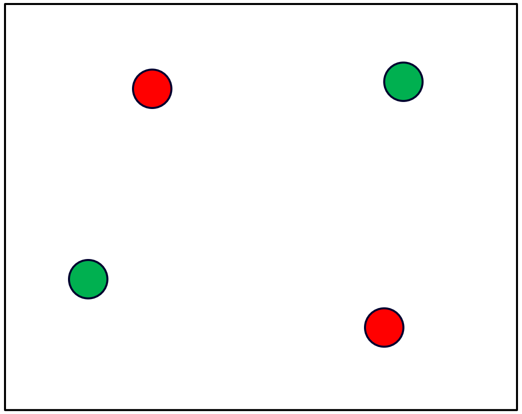
Example: VC-dimension of linear separators

E.g., H= linear separators in $\mathbf{R}^2$

$\mathrm{VCdim}(\mathrm{H}) < 4$



Case 1: one point inside the triangle formed by the others. Cannot label inside point as positive and outside points as negative.



Case 2: all points on the boundary (convex hull). Cannot label two diagonally as positive and other two as negative.

Fact: VCdim of linear separators in $\mathbf{R}^{\mathbf{d}}$ is d+1

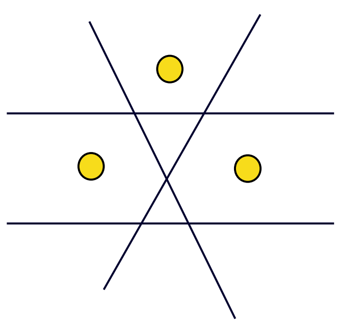
VC-dimension [Vapnik-Chervonenkis, 1971]

VC-dimension of a function class H is the cardinality of the largest set S that can be labeled in all possible ways $2^{|S|}$ by H.

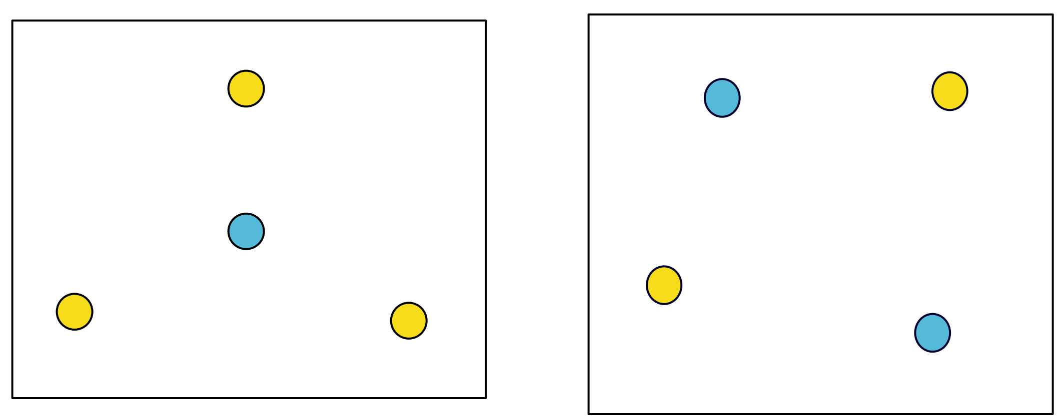
[If arbitrarily large finite sets can be shattered by H, then $\mathrm{VCdim(H)} = \infty$]

E.g., *H*= linear separators in $\mathbf{R}^2$ $\quad \mathrm{VCdim(H)} = 3$

$\mathrm{VCdim(H)} \geq 3$



$\mathrm{VCdim(H)} < 4$

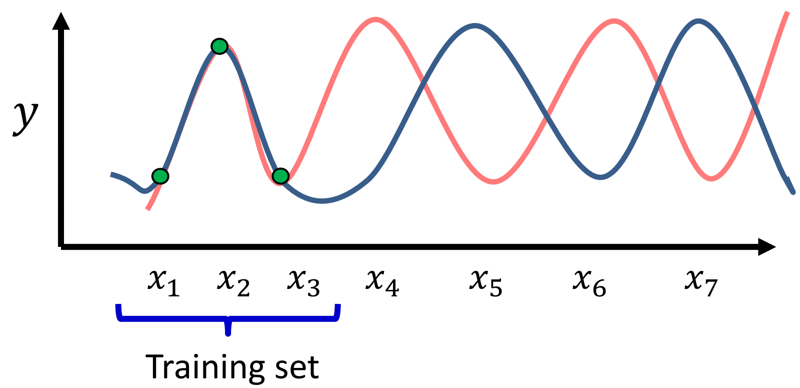


Why VC-dimension matters

Why does it matter "how many points we can label in all possible ways with functions from the class"?

<u>Example</u>: H={all 0/1 fns over some domain}, then any set of points can be labelled in all possible ways with fns H, $\mathrm{VCdim(H)} = \infty$.

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



No hope to generalize.

Why VC-dimension matters

Why does it matter "how many points we can label in all possible ways with functions from the class"?

Classes of finite VC-dimension

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: $\mathrm{N} = \mathrm{O}(\mathrm{VCdim}(H)\,/\epsilon^{2})$ training instances suffice for generalizability.

Pseudo-dimension [Pollard 1984]

Pseudo-dimension: complexity measure that characterizes the sample complexity of ***real-valued*** function classes.

The **pseudo-dimension** of a function class F is the cardinality of the largest set $S = \{x_1, \dots, x_N\}$ and thresholds $y_1, \dots, y_N$ s.t. all 2^N above/below patterns can be achieved by functions $f \in F$.

- E.g., for $N = 2$, there should exist $f_1 \in F$ s.t. $f_1(x_1) < y_1, f_1(x_2) < y_2$

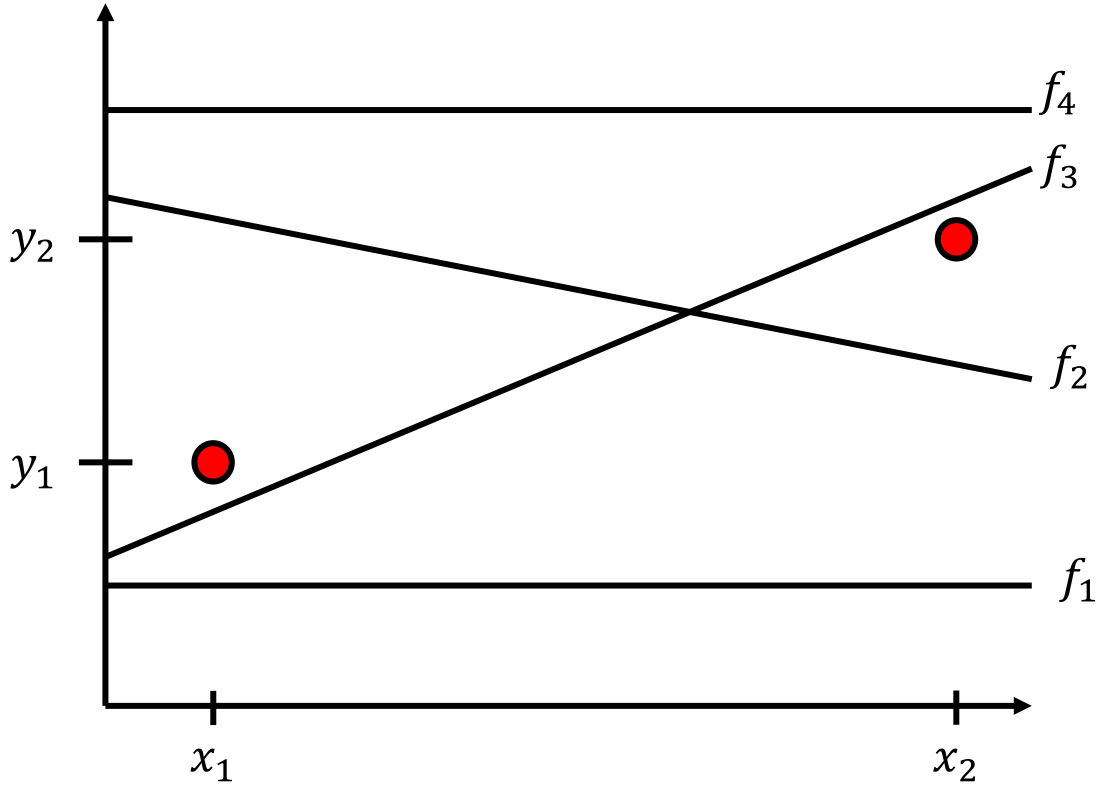
  $f_2 \in F$ s.t. $f_2(x_1) > y_1, f_2(x_2) < y_2$

  $f_3 \in F$ s.t. $f_3(x_1) < y_1, f_3(x_2) > y_2$

  $f_4 \in F$ s.t. $f_4(x_1) > y_1, f_4(x_2) > y_2$

Equivalently, the **pseudo-dimension** of F is the VC dimension of the class of "below-the-graph" indicator functions $\{B_f(x, y) = \mathrm{sgn}(f(x) - y) : f \in F\}$

Example: Affine functions on $\mathbb{R}$



Consider points $x_1, x_2 \in \mathbb{R}$ with thresholds y_1, y_2. All four above/below patterns can be realized by the class F of affine functions on $\mathbb{R}$, $\mathrm{F} = \{x \longmapsto ax + b : a, b \in \mathbb{R}\}$.

f_1 produces (below,below); f_2 produces (above,below); f_3 produces (below,above); f_4 produces (above,above)

Uniform convergence guarantees

Theorem [Pollard'84; Dudley '67]

For any $\delta \in (0,1)$ and any distribution $\mathcal{D}$ over $\mathcal{X}$, with prob. $1-\delta$ over the draw $\{x_1, \dots, x_N\} \sim \mathcal{D}^N$, for all $f \in F$,

$$\left|\mathbb{E}_{x\sim\mathcal{D}}[f(x)] - \frac{1}{N}\sum_{i=1}^{N} f(x_i)\right| = O\left(U\sqrt{\frac{\mathbf{Pdim}(\mathcal{F})}{N}} + U\sqrt{\frac{\log(1/\delta)}{N}}\right),$$

true expectation

Empirical average

Decay rate

Bounding Pdim of auction classes.
Example: Second–price auction with a reserve

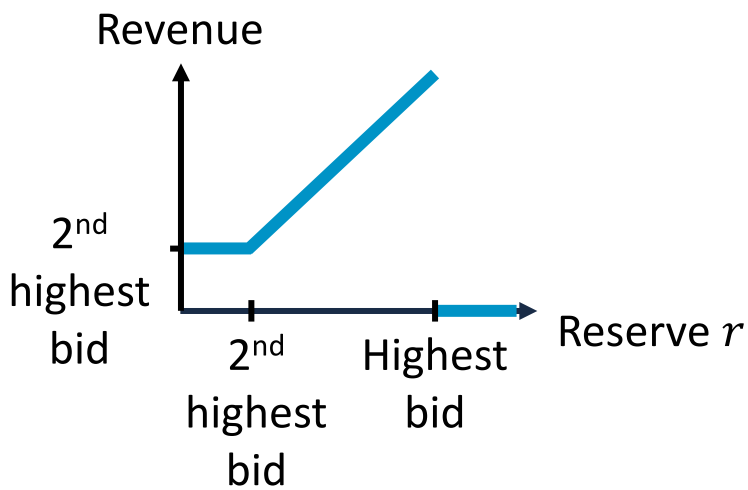
Setup: single-item, multi-bidder.

1. Auctioneer sets a reserve price r.
2. Highest bidder wins if bid $\geq r$. Pays maximum of the second highest bid and r.

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

Key idea:

Revenue = $\max\{r, \text{2nd highest bid}\} \cdot \mathbf{1}_{\{\text{highest bid} \geq r\}}$



Bounding Pdim of auction classes.

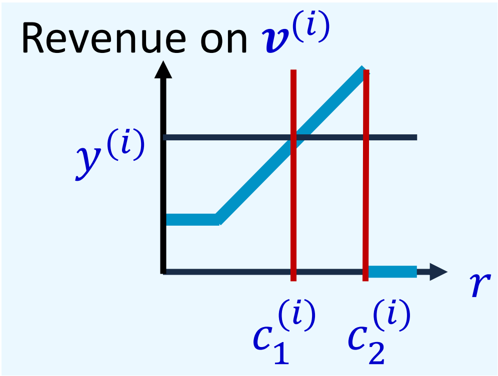
Example: Second–price auction with a reserve

Theorem [Mohri and Medina, ICML'14; Morgenstern and Roughgarden, COLT'16; Balcan, Sandholm, and Vitercik, EC'18]

$\mathcal{M} = \{\mathrm{rev}_r :=$ revenue function of 2nd-price auction w/ reserve $r\}$. $\mathrm{Pdim}(\mathcal{M}) \leq 2$.

Key idea: Consider some example $\mathbf{v}^{(\mathrm{i})}$ and revenue-threshold $\mathrm{y}^{(\mathrm{i})}$.

- Scanning r from 0 to ∞ there will be (at most) two cutoff values $\mathrm{c}_1^{(\mathrm{i})}$, $\mathrm{c}_2^{(\mathrm{i})}$ where revenue goes from "below" to "above" to "below".
- With N examples, look at all $2\mathrm{N}$ cutoff values.
- All r in the same interval between consecutive cutoff values will give the same binary pattern.
- So, at most $2\mathrm{N}+1$ binary patterns.



- Pseudo-dimension is max N s.t. all 2^{N} binary patterns are achievable. Need $2^{\mathrm{N}} \leq 2\mathrm{N}+1$ so $\mathrm{N} \leq 2$.

A general theorem for bounding mechanism classes' pseudo-dimension

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

Assume:

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

Then the pseudo-dimension of $\{\text{revenue}_M : M \in \mathcal{M}\}$ is $O(\boldsymbol{d} \log(\boldsymbol{dt}))$.

High level learning theory bit

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

Assume:

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

Then the pseudo-dimension of $\{\text{revenue}_M : M \in \mathcal{M}\}$ is $O(\boldsymbol{d} \log(\boldsymbol{dt}))$.

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

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

- Function class we care about: $\{\text{revenue}_{\boldsymbol{p}} : \text{parameter vectors } \boldsymbol{p}\}$.
- Proof uses structure of **dual class** $\{\text{revenue}_{\boldsymbol{v}} : \text{buyer values } \boldsymbol{v}\}$.

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

High level learning theory bit

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

Assume:

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

Then the pseudo-dimension of $\{\mathrm{revenue}_M : M \in \mathcal{M}\}$ is $O(\boldsymbol{d} \log(\boldsymbol{dt}))$.

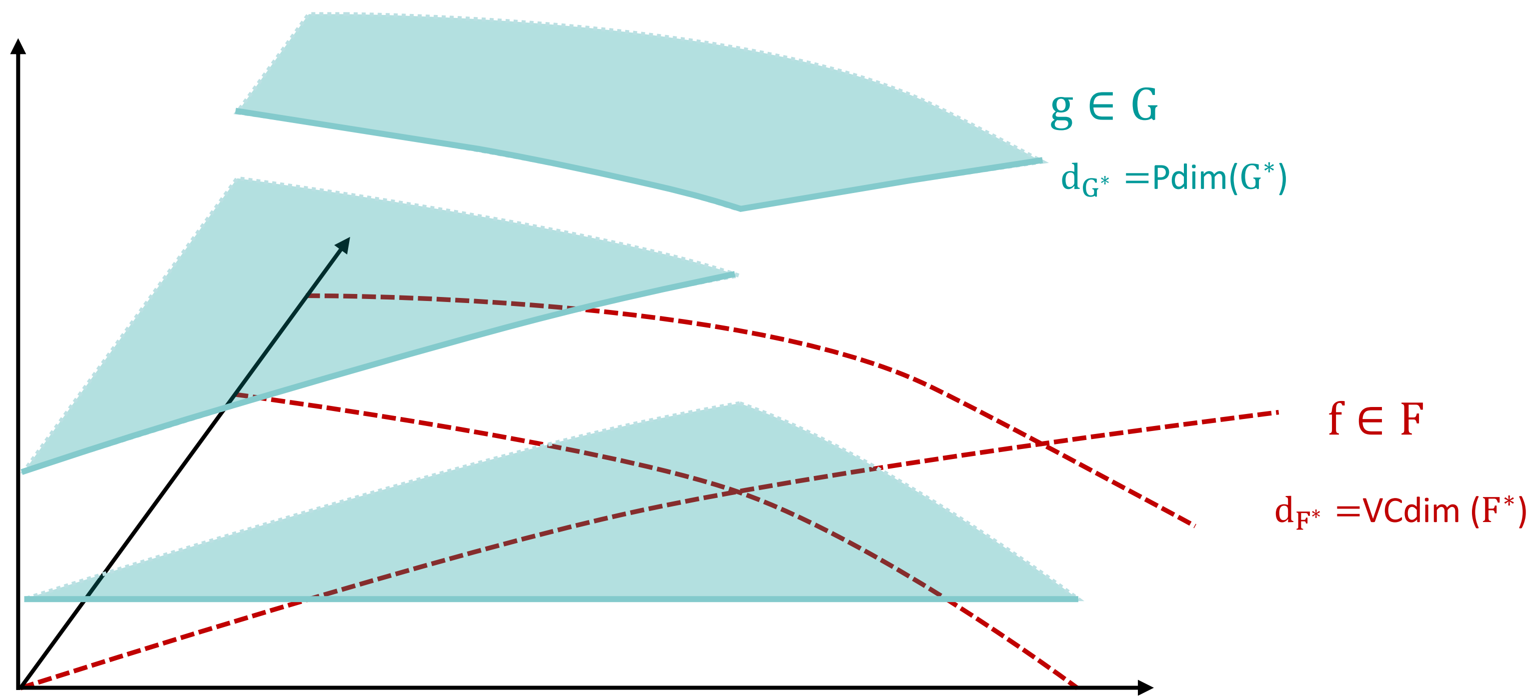
Proof uses structure of **dual class** $\{\mathrm{revenue}_{\boldsymbol{v}} : \text{buyer values } \boldsymbol{v}\}$.

Usefulness of the dual class also exhibited by [Bartlett, Maiorov, Meir, NIPS'99] and [Moran and Yehudayoff, JACM'15].

General Sample Complexity via Dual Classes

Thm: Assume $\mathrm{cost}_I(\alpha)$: boundary fns $f_1, f_2, \dots, f_N \in F$ s.t. within each region, $\mathrm{cost}_I(\alpha) = g(\alpha)$ for some $g \in G$.

$$\mathrm{Pdim}(\{\mathrm{cost}_\alpha(I)\}) = \tilde{O}\big((d_{F^*} + d_{G^*}) + d_{F^*} \log N\big)$$



[Balcan, Dick, DeBlasio, Kingsford, Sandholm, Vitercik, STOC-21: "How much data is sufficient to learn high-performing algorithms?"]

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

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.

Outline

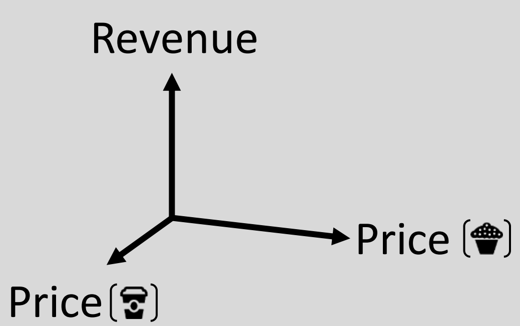
1. Introduction
2. Mechanism design basics
3. Automated mechanism design (AMD)
4. Sample complexity guarantees for AMD
   a) Formal guarantees
   b) Applications of BSV18 to single-item settings
   c) Applications of BSV18 to multi-item/multi-unit settings

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

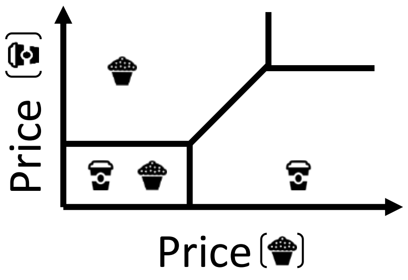
Theorem

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

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

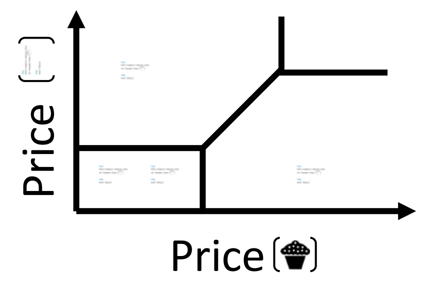
Theorem

$\text{Pdim}(\mathcal{M}) = O(d \log(dt))$ w/ $d = (\#\text{dimensions}) = (\#\text{items})$

and $t = (\#\text{ hyperplanes}) = (\#\text{ buyers}) \cdot \binom{2^{(\#\text{ items})}}{2}$.

Corollary

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



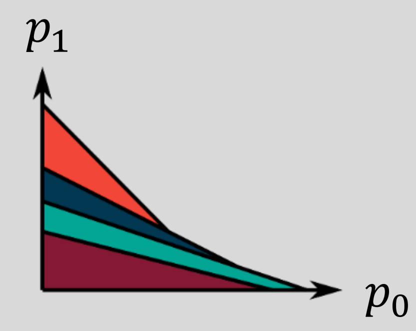
Also shown by Morgenstern and Roughgarden [COLT ‘16]

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

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

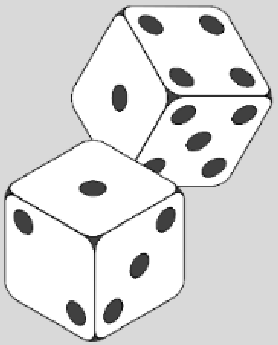
Randomized mechanisms (lotteries)

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

- Lottery represented by vector $\left(\phi_1, \ldots, \phi_{(\#\text{items})}\right)$ and price p
- If buyer buys lottery, pays p and receives each item i w.p. ϕ_i
  - Expected utility is $\sum_{i=1}^{(\#\text{items})} v(\{i\}) \cdot \phi_i - p$
- 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.

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

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

Theorem

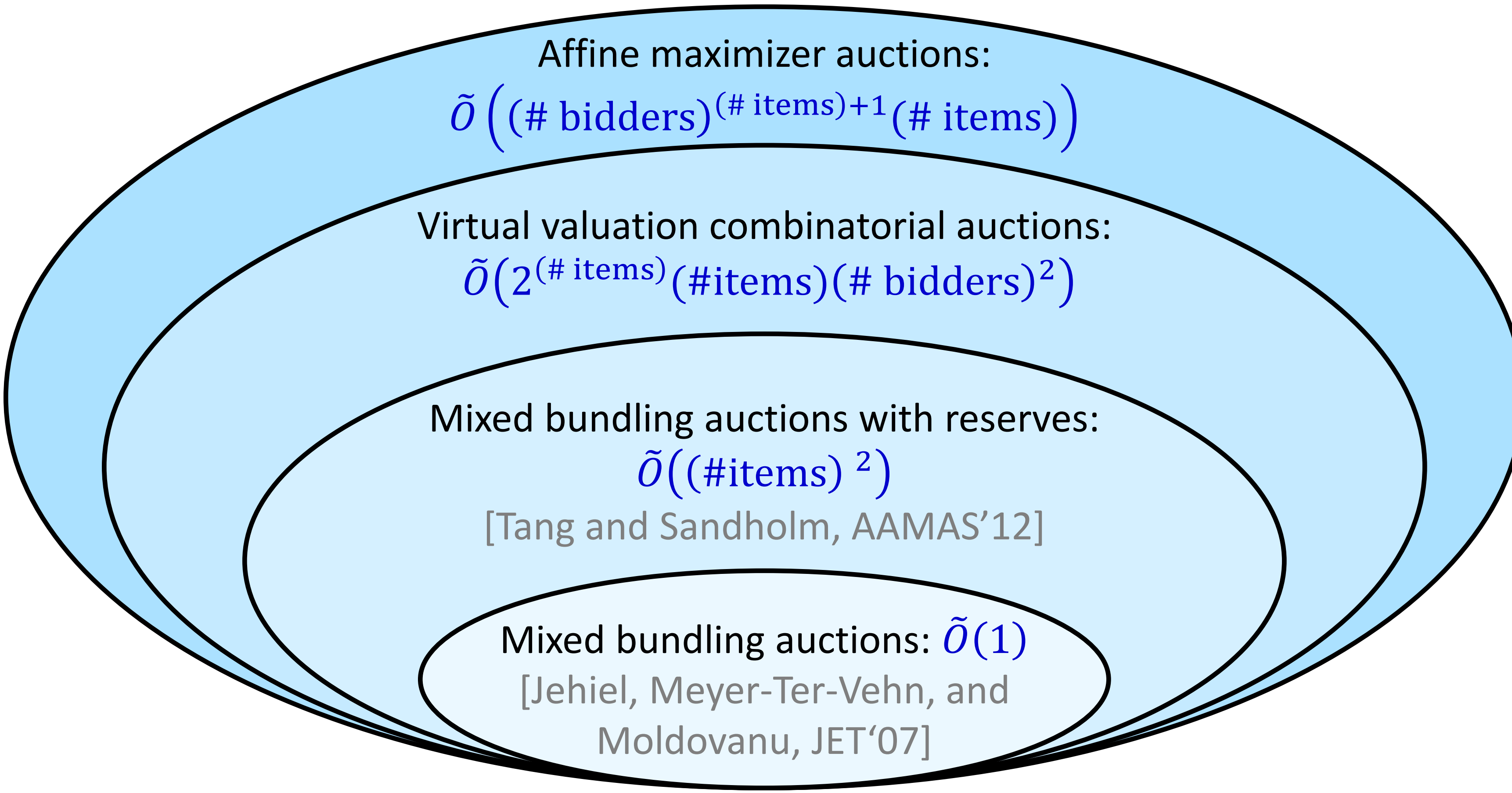
$\text{Pdim}(\mathcal{M}) = O(d \log(dt))$ with $t = (\#\text{ hyperplanes}) = L^2$

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

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

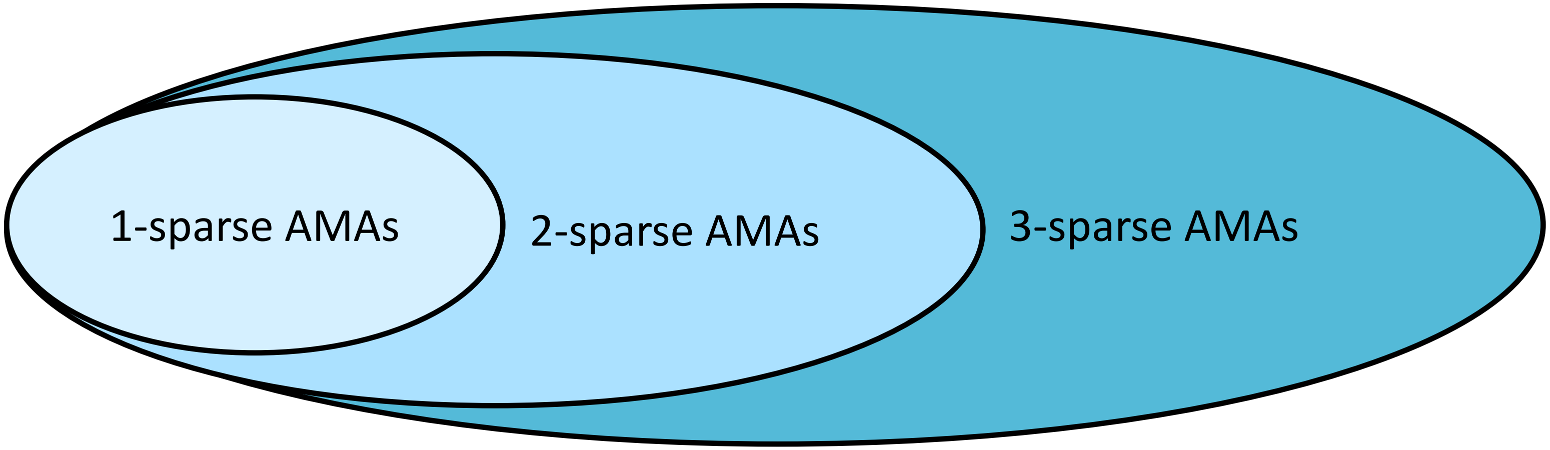
[Jehiel, Meyer-Ter-Vehn, and Moldovanu, J. of Econ. Theory ‘07]

Fine-grained auction hierarchies

Fine-grained **hierarchies** of AMAs:

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

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



Fine-grained auction hierarchies

Fine-grained **hierarchies** of AMAs:

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

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

- A-boosted AMAs: only allocations in A boosted

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

Fine-grained auction hierarchies

Fine-grained **hierarchies** of AMAs:

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

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

- A-boosted AMAs: only allocations in A boosted

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

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

Increases with k

Decreases with k

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

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

Optimizing the revenue-generalization tradeoff

We provide guarantees for **optimizing this tradeoff**

E.g., A-boosted AMAs $\mathcal{M}_A$:

Theorem

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

Increases with $|A|$

Decreases with $|A|$

Let A^* be the set of boosted allocations under optimal AMA.

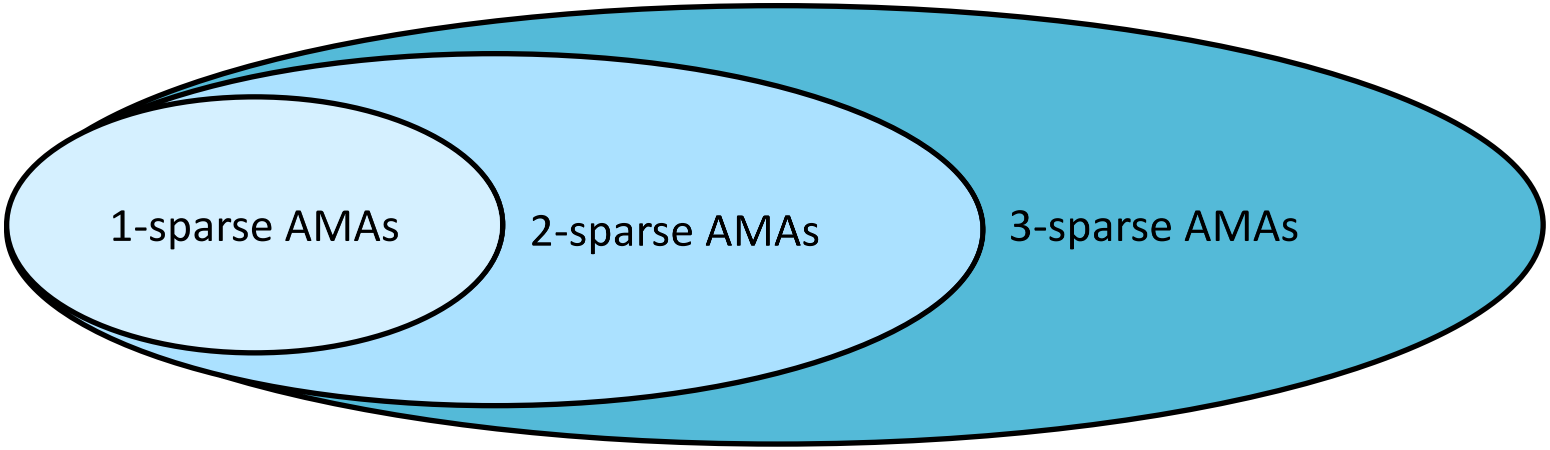
$\widehat{M}$'s revenue is within $\tilde{O}\left(\mathrm{U}\sqrt{\frac{\#\text{bidders} + |A^*|}{|S|}}\right)$ of optimal AMA's revenue.

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]