# Query and Computational Complexity of Combinatorial Auctions

Jan Vondrák

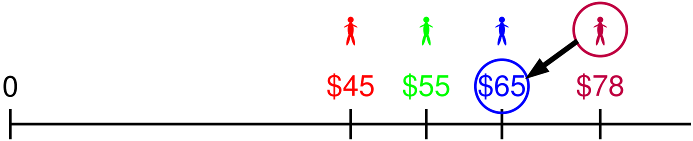
IBM Almaden Research Center
San Jose, CA

Algorithmic Frontiers, EPFL, June 2012

# Vickrey (2nd price) auction

Suppose we are selling 1 item in an auction:

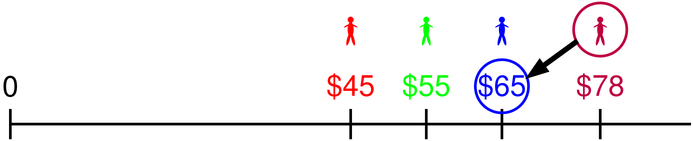
- Assume agent $i$'s true valuation of the item is $v_i$.
- We ask the agents to submit their bids $v_i'$ and announce that the highest bidder will get the item at the 2nd highest price.



# Vickrey (2nd price) auction

Suppose we are selling 1 item in an auction:

- Assume agent $i$'s true valuation of the item is $v_i$.
- We ask the agents to submit their bids $v_i'$ and announce that the highest bidder will get the item at the 2nd highest price.




This satisfies 3 properties:

1. A rational agent knows that the best strategy is to submit $v_i' = v_i$
2. Computing the outcome is easy
3. The item goes to the agent who benefits the most

# Auctions with multiple items

**Meta-question:**
*Is there such a mechanism for auctions with multiple (related) items?*

# Auctions with multiple items

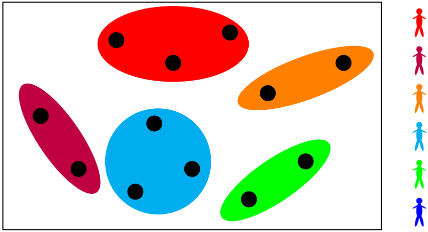
**Meta-question:**
*Is there such a mechanism for auctions with multiple (related) items?*

**Examples:**

- *Google AdWords:*
  agents = potential advertisers
  items = ads associated with search keywords.
- *FCC spectrum auctions:*
  agents = wireless communication companies
  items = licences to use certain frequencies in certain areas.

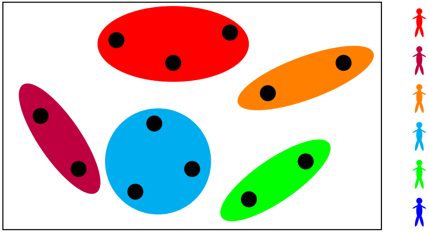
# Combinatorial auctions [Lehman, Lehman, Nisan '01]

**Problem:** $|M| = m$ items are to be sold to $n$ agents with (monotone) valuations $v_i : 2^M \to \mathbb{R}_+$.



# Combinatorial auctions [Lehman, Lehman, Nisan '01]

**Problem:** $|M| = m$ items are to be sold to $n$ agents with (monotone) valuations $v_i : 2^M \to \mathbb{R}_+$.



How do we sell the items, so that

1. Agents are incentivized to reveal their true valuations
2. The mechanism is computationally efficient
3. The "social welfare" $\sum_{i=1}^{n} v_i(S_i)$ is close to optimal

# Truthful mechanisms

What is a mechanism for combinatorial auctions?

- Agents submit their valuation functions $v_i : 2^M \to \mathbb{R}_+$ (succinct description / oracle)
- Mechanism computes a (possibly random) *allocation* $(A_1, \ldots, A_n)$ and *payments* $(p_1, \ldots, p_n)$
- Agent $i$ pays $p_i$ and receives set $A_i$.

# Truthful mechanisms

What is a mechanism for combinatorial auctions?

- Agents submit their valuation functions $v_i : 2^M \to \mathbb{R}_+$ (succinct description / oracle)
- Mechanism computes a (possibly random) *allocation* $(A_1, \ldots, A_n)$ and *payments* $(p_1, \ldots, p_n)$
- Agent $i$ pays $p_i$ and receives set $A_i$.

## Definition

A mechanism is *universally truthful*, if for every agent $i$, his true valuation $v_i$, reported valuation $v_i'$ and others' reported valuations $v_{-i}'$, with probability 1,

$$v_i(A_i(v_i, v'_{-i})) - p_i(v_i, v'_{-i}) \geq v_i(A_i(v'_i, v'_{-i})) - p_i(v'_i, v'_{-i}).$$

A mechanism is *truthful in expectation*, if

$$\mathbb{E}[v_i(A_i(v_i, v'_{-i})) - p_i(v_i, v'_{-i})] \geq \mathbb{E}[v_i(A_i(v'_i, v'_{-i})) - p_i(v'_i, v'_{-i})].$$

# VCG mechanism

**The Vickrey-Clarke-Groves mechanism**:
Given reported valuations $v_i$, find an allocation $(A_1, \ldots, A_n)$ maximizing the *social welfare*, $\sum_{i=1}^n v_i(A_i)$, and charge prices $p_i$ that reflect the "damage" that agent $i$ inflicts on the other agents by participating.

**Theorem (VCG '73)**

*The VCG mechanism is truthful.*

# VCG mechanism

**The Vickrey-Clarke-Groves mechanism**:
Given reported valuations $v_i$, find an allocation $(A_1, \ldots, A_n)$ maximizing the *social welfare*, $\sum_{i=1}^{n} v_i(A_i)$, and charge prices $p_i$ that reflect the "damage" that agent $i$ inflicts on the other agents by participating.

**Theorem (VCG '73)**

*The VCG mechanism is truthful.*

Is the problem solved?

1. it is truthful
2. it optimizes the social welfare $\sum_{i=1}^{n} v_i(A_i)$
3. ~~it is computationally efficient~~ - NO!

Social welfare optimization is NP-hard in most non-trivial settings...

# Truthfulness vs. computational complexity

**Let's relax our requirements:** we want

1. a truthful mechanism (maybe in expectation)
2. computationally efficient
3. optimizing $\sum_{i=1}^{n} v_i(A_i)$ *approximately*, for some class of valuations

# Truthfulness vs. computational complexity

**Let's relax our requirements:** we want

1. a truthful mechanism (maybe in expectation)
2. computationally efficient
3. optimizing $\sum_{i=1}^{n} v_i(A_i)$ *approximately*, for some class of valuations

*Without truthfulness*, such approximation algorithms are known:

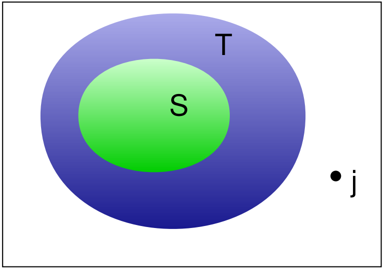
- $3/4$ for budget-additive valuations [Chakrabarty, Goel '08]
- $1 - 1/e$ for coverage valuations [Dobzinski, Schapira '07]
- $1 - 1/e$ for submodular valuations [V. '08]

(For general valuations, the problem is inapproximable within $m^{\epsilon - 1/2}$.)

# Submodular functions

Submodularity = property of *diminishing returns*.

Let the *marginal value* of element j be $f_S(j) = f(S + j) - f(S)$.



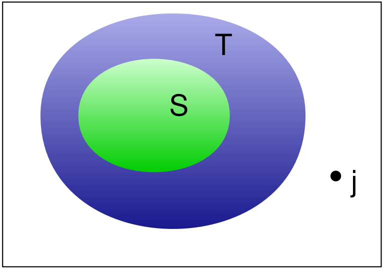

**Definition:** $f$ is submodular, if
$j$ cannot add more value to $T$ than $S$.

$$f_S(j) \geq f_T(j)$$

# Submodular functions

Submodularity = property of *diminishing returns*.

Let the *marginal value* of element j be $f_S(j) = f(S + j) - f(S)$.




**Definition:** $f$ is submodular, if $j$ cannot add more value to $T$ than $S$.

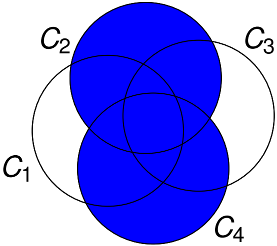
$$f_S(j) \geq f_T(j)$$

*Representation:* in general by an *oracle* (e.g. value oracle: $f(S) = ?$), some subclasses can be succinctly represented (poly-size encoding $e(f)$ + efficient procedure to evaluate $f(S)$, given $(e(f), S)$).

# Subclasses of submodular functions

**Coverage functions:**
Given $C_1, \ldots, C_m \subset \mathcal{U}$,

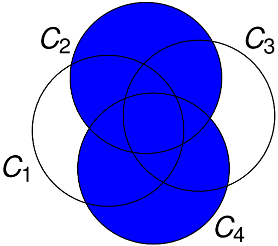
$$f(S) = \left| \bigcup_{j \in S} C_j \right|.$$

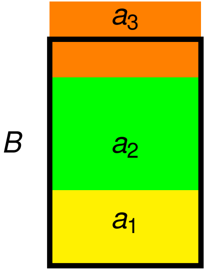


# Subclasses of submodular functions

**Coverage functions:**
Given $C_1, \ldots, C_m \subset \mathcal{U}$,

$$f(S) = \left| \bigcup_{j \in S} C_j \right|.$$




**Budget-additive functions:**

$$f(S) = \min\{\sum_{j \in S} a_j, B\}$$

# How to optimize social welfare? (without truthfulness)

**Greedy algorithm:** allocate each item to an agent of maximum marginal value $\Rightarrow \frac{1}{2}$-approximation [Fisher,Nemhauser,Wolsey '78]

# How to optimize social welfare? (without truthfulness)

**Greedy algorithm:** allocate each item to an agent of maximum marginal value $\Rightarrow \frac{1}{2}$-approximation [Fisher,Nemhauser,Wolsey '78]

**Continuous greedy algorithm:** allocate items greedily in a fractional fashion, with respect to the *multilinear extension* $F : [0,1]^{m \times n} \to \mathbb{R}$:

- $F(x) = \sum_{i=1}^{n} \mathbb{E}[v_i(\hat{x}_i)]$, where $\hat{x}_i$ is obtained by rounding each $x_{ij}$ randomly to $0/1$ with probabilities $x_{ij}$.

constrained by the *assignment polytope*:

- $P = \{x \in [0,1]^{m \times n} : \forall j; \sum_i x_{ij} \leq 1\}$.

$\Rightarrow$ $(1 - 1/e)$-approximation [V. '08],
optimal unless $P = NP$ [Khot,Lipton,Markakis,Mehta '05]

# How to optimize social welfare? (without truthfulness)

**Greedy algorithm:** allocate each item to an agent of maximum marginal value $\Rightarrow \frac{1}{2}$-approximation [Fisher,Nemhauser,Wolsey '78]

**Continuous greedy algorithm:** allocate items greedily in a fractional fashion, with respect to the *multilinear extension* $F : [0,1]^{m \times n} \to \mathbb{R}$:

- $F(x) = \sum_{i=1}^{n} \mathbb{E}[v_i(\hat{x}_i)]$, where $\hat{x}_i$ is obtained by rounding each $x_{ij}$ randomly to $0/1$ with probabilities $x_{ij}$.

constrained by the *assignment polytope*:

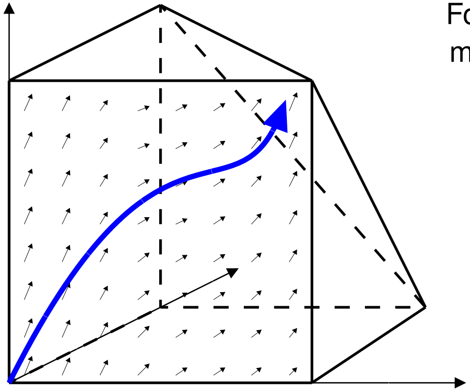
- $P = \{x \in [0,1]^{m \times n} : \forall j; \sum_i x_{ij} \leq 1\}$.

$\Rightarrow (1 - 1/e)$-approximation [V. '08],
optimal unless $P = NP$ [Khot,Lipton,Markakis,Mehta '05]

But these algorithms do not have any truthfulness properties.

# Continuous Greedy Algorithm

**Problem:** $\max\{F(x) : x \in P\}$; we know $\frac{\partial F}{\partial x_i} \geq 0, \frac{\partial^2 F}{\partial x_i \partial x_j} \leq 0$.



For each $x \in P$, define $v(x)$ by maximizing $v \cdot \nabla F$ over $v \in P$.

Define a curve $x(t)$:

$$y(0) = 0$$

$$\frac{dx}{dt} = v(x)$$

Run this process for $t \in [0, 1]$ and return $x(1)$.

**Claim:** This algorithm gives a $(1 - 1/e)$-approximation.

# Back to truthful mechanisms

## Central Question of Algorithmic Mechanism Design

*Is it possible to achieve an (approximately) optimal solution under the requirements of*

1. *truthfulness*
2. *polynomial running time*

*when each can be achieved separately?*

# Back to truthful mechanisms

## Central Question of Algorithmic Mechanism Design

*Is it possible to achieve an (approximately) optimal solution under the requirements of*

1. *truthfulness*
2. *polynomial running time*

*when each can be achieved separately?*

*Known answer:* NO - for the problem of *combinatorial public projects* and universally truthful mechanisms [Papadimitriou,Schapira,Singer '08].

Combinatorial auctions?
Truthful-in-expectation mechanisms?

# Some known results for combinatorial auctions

On the positive side:

- There is a truthful $O(1/\log m \log\log m)$-approximation for submodular valuations (with "demand queries") [Dobzinski '07]

On the negative side:

- Any non-trivial "VCG-based" mechanism for submod. valuations would require exponential communication [Dobzinski, Nisan '07]
- Any non-trivial "VCG-based" mechanism for budget-additive or coverage valuations would imply $NP \subseteq P/poly$ [BDFKMPSSU '10]

# Some known results for combinatorial auctions

On the positive side:

- There is a truthful $O(1/\log m \log\log m)$-approximation for submodular valuations (with "demand queries") [Dobzinski '07]

On the negative side:

- Any non-trivial "VCG-based" mechanism for submod. valuations would require exponential communication [Dobzinski, Nisan '07]
- Any non-trivial "VCG-based" mechanism for budget-additive or coverage valuations would imply $NP \subseteq P/poly$ [BDFKMPSSU '10]

What is "VCG-based"? A mechanism which optimizes social welfare over some subset of possible allocations.

In some settings, all truthful mechanisms are known to be VCG-based.
But not for combinatorial auctions!

# Recent progress

1. [Dobzinski '11] proved that if valuations are submodular and the only access to them is through a *value oracle*, then *no universally truthful mechanism* gives $m^{\epsilon-1/2}$-approximation for $\epsilon > 0$.
2. [Dughmi, Roughgarden & Yan '11] a *truthful-in-expectation* $(1-1/e)$-approximation for *coverage valuations*.

| Valuations | Without truthfulness | Truthful-in-expectation | Univ. truthful |
|---|---|---|---|
| Coverage | $1-1/e$ | $1-1/e$ | ??? |
| Submodular (succinct rep.) | $1-1/e$ | ??? | ??? |
| Submodular (value oracle) | $1-1/e$ | ??? | $m^{o(1)-1/2}$ |

# Recent progress

1. [Dobzinski '11] proved that if valuations are submodular and the only access to them is through a *value oracle*, then *no universally truthful mechanism* gives $m^{\epsilon-1/2}$-approximation for $\epsilon > 0$.
2. [Dughmi, Roughgarden & Yan '11] a *truthful-in-expectation* $(1-1/e)$-approximation for *coverage valuations*.

**NEW RESULTS:** [Dughmi, V. '11], [Dobzinski, V. '12]

| Valuations | Without truthfulness | Truthful-in-expectation | Univ. truthful |
|---|---|---|---|
| Coverage | $1-1/e$ | $1-1/e$ | ??? |
| Submodular (succinct rep.) | $1-1/e$ | NEW: $n^{-\Omega(1)}$ | NEW: $m^{o(1)-1/2}$ |
| Submodular (value oracle) | $1-1/e$ | NEW: $m^{-\Omega(1)}$ | $m^{o(1)-1/2}$ |

# Our new results

## Theorem (Dughmi, V. '11)

*There is* $\gamma > 0$ *such that no truthful-in-expectation mechanism for submodular valuations in the value oracle model achieves a better than* $m^{-\gamma}$*-approximation.*

# Our new results

**Theorem (Dughmi, V. '11)**

*There is $\gamma > 0$ such that no truthful-in-expectation mechanism for submodular valuations in the value oracle model achieves a better than $m^{-\gamma}$-approximation.*

**Theorem (Dobzinski, V. '12)**

*There is a class of succinctly represented submodular valuations such that unless $NP \subseteq P/poly$,*

- *No deterministic truthful mechanism achieves $m^{\epsilon-1/2}$-approximation, for any $\epsilon > 0$.*
- *No truthful-in-expectation mechanism achieves $n^{-\gamma}$-approximation, for some $\gamma > 0$.*

*I.e. we identify a variant of combinatorial auctions where computational efficiency and truthfulness (even in expectation) are incompatible.*

# How do we prove this?

**Some notable points:**

- We **do not** give any characterization of truthful-in-expectation mechanisms for combinatorial auctions.

# How do we prove this?

**Some notable points:**

- We **do not** give any characterization of truthful-in-expectation mechanisms for combinatorial auctions.
- We **do not** appeal to the "complexity of the range" of the mechanism (e.g. VC-dimension), like some previous work.

# How do we prove this?

**Some notable points:**

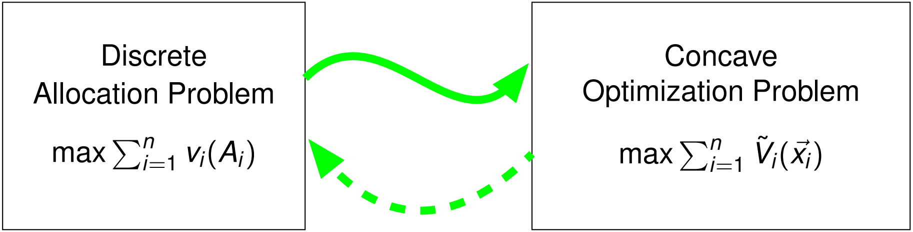
- We **do not** give any characterization of truthful-in-expectation mechanisms for combinatorial auctions.
- We **do not** appeal to the "complexity of the range" of the mechanism (e.g. VC-dimension), like some previous work.
- Rather, we use the **symmetry gap** technique to prove the existence of a certain sequence of distributions possibly output by the mechanism, which leads to a contradiction.

**Some notable points:**

- We **do not** give any characterization of truthful-in-expectation mechanisms for combinatorial auctions.
- We **do not** appeal to the "complexity of the range" of the mechanism (e.g. VC-dimension), like some previous work.
- Rather, we use the **symmetry gap** technique to prove the existence of a certain sequence of distributions possibly output by the mechanism, which leads to a contradiction.
- To prove a computational hardness result, we encode the valuations succinctly using **list decodable codes**.

# The DRY mechanism for coverage valuations

Let's start from the mechanism of Dughmi, Roughgarden & Yan:

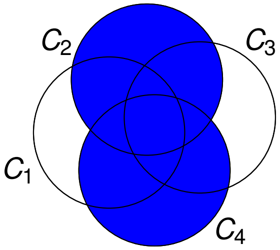



- The discrete allocation problem is replaced by a concave optimization problem, which can be solved *optimally*.
- Solutions correspond to *distributions over allocations*.
- For any such algorithm, VCG payments can be defined so that the resulting mechanism is *truthful in expectation*.

# DRY mechanism: more details

**Continuous optimization problem:**
$\max\{\sum_{i=1}^{n} \tilde{V}_i(\vec{x}_i) : \sum_{i=1}^{n} \vec{x}_i = (1, 1, \ldots, 1)\}$, where

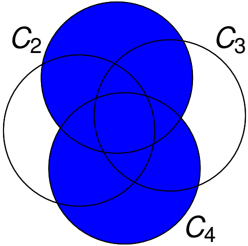
$\tilde{V}_i(\vec{x}_i) =$ expected utility of agent $i$ if he receives $C_j$ independently with prob. $\tilde{x}_{ij} = 1 - e^{-x_{ij}}$.



# DRY mechanism: more details

**Continuous optimization problem:**
$\max\{\sum_{i=1}^{n} \tilde{V}_i(\vec{x}_i) : \sum_{i=1}^{n} \vec{x}_i = (1, 1, \ldots, 1)\}$, where

$\tilde{V}_i(\vec{x}_i) =$ expected utility of agent $i$ if he receives $C_j$ independently with prob. $\tilde{x}_{ij} = 1 - e^{-x_{ij}}$.



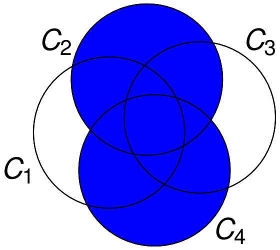

**Lemma:** If $v_i$ is a coverage function, then $\tilde{V}_i$ is a concave function.
*Proof:* $\Pr[a \in \bigcup_{j \in A_i} C_j] = 1 - \prod_{j: a \in C_j} e^{-x_{ij}} = 1 - e^{-\sum_{j: a \in C_j} x_{ij}}$.

$\Rightarrow$ the continuous optimization problem can be solved exactly and everything works!

# DRY mechanism: more details

**Continuous optimization problem:**
$\max\{\sum_{i=1}^{n} \tilde{V}_i(\vec{x}_i) : \sum_{i=1}^{n} \vec{x}_i = (1,1,\ldots,1)\}$, where

$\tilde{V}_i(\vec{x}_i)$ = expected utility of agent $i$ if he receives $C_j$ independently with prob. $\tilde{x}_{ij} = 1 - e^{-x_{ij}}$.




**Lemma:** If $v_i$ is a coverage function, then $\tilde{V}_i$ is a concave function.
*Proof:* $\Pr[a \in \bigcup_{j \in A_i} C_j] = 1 - \prod_{j:a \in C_j} e^{-x_{ij}} = 1 - e^{-\sum_{j:a \in C_j} x_{ij}}$.

$\Rightarrow$ the continuous optimization problem can be solved exactly and everything works! So why not for submodular functions?

# Stage 1: why the DRY mechanism fails

## Lemma

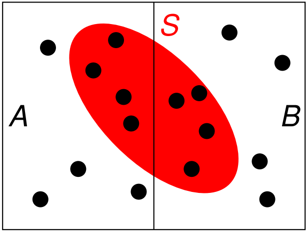
*In the value oracle model, solving the optimization problem* $\max\{\sum_{i=1}^{2} \tilde{V}_i(\vec{x}_i) : \sum_{i=1}^{2} \vec{x}_i = \mathbf{1}\}$ *for submodular functions even within factor* 0.9 *would require exponentially many value queries.*

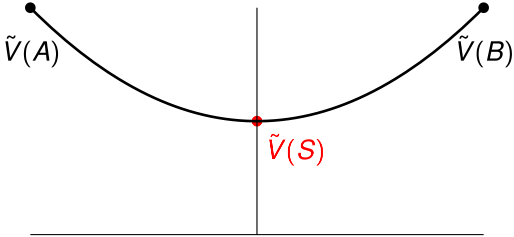
# Stage 1: why the DRY mechanism fails

## Lemma

*In the value oracle model, solving the optimization problem* $\max\{\sum_{i=1}^{2} \tilde{V}_i(\vec{x}_i) : \sum_{i=1}^{2} \vec{x}_i = \mathbf{1}\}$ *for submodular functions even within factor* 0.9 *would require exponentially many value queries.*

**Approach:** We use the following valuation functions $v(S)$:



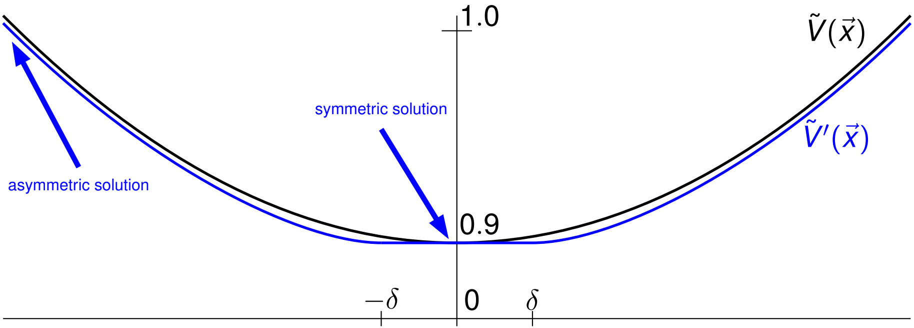

- $v(S) = 1 - (1 - \alpha|S \cap A|)_+(1 - \alpha|S \cap B|)_+$,
- the extension $\tilde{V}(\vec{x})$ is *not concave*, and in particular
- there is a gap of 0.9 between *symmetric* and *asymmetric* solutions

# Stage 1: the symmetry gap argument

*The technique of symmetry gap [V. '09] implies:*

For a suitable perturbation of $v(S)$, the partition $(A, B)$ cannot be found using poly-many value queries

$\Rightarrow$ only symmetric solutions can be found efficiently.




$\Rightarrow \max\{\sum_{i=1}^{n} \tilde{V}_i(\vec{x}_i) : \sum_{i=1}^{n} \vec{x}_i = \mathbf{1}\}$ cannot be solved better than within 0.9.

# Stage 2: ruling out all T.I.E. mechanisms

**High-level sketch:** we combine several ingredients:

1. The construction of symmetric valuation functions from Step 1.
2. Dobzinski's "direct hardness" approach [STOC'11] for deterministic mechanisms.
3. An inductive argument, boosting the hardness factor from a constant to a polynomial $m^\gamma$.

# Stage 2: ruling out all T.I.E. mechanisms

**High-level sketch:** we combine several ingredients:

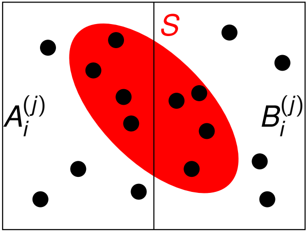
1. The construction of symmetric valuation functions from Step 1.
2. Dobzinski's "direct hardness" approach [STOC'11] for deterministic mechanisms.
3. An inductive argument, boosting the hardness factor from a constant to a polynomial $m^{\gamma}$.

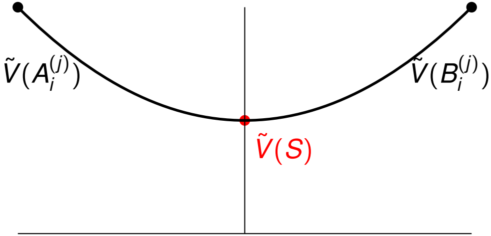
From [Dobzinski '11]:

- *Taxation principle*: For any truthful mechanism, if we fix the valuations $v_{-i}$ of agents $i' \neq i$, the mechanism must maximize $\mathbb{E}[v(S) - p_S]$ over the "menu" of all distributions of set $S$ and price $p_S$, possibly allocated to $i$.
- *Rich menu:* If the menu is sufficiently complicated, it is hard for a mechanism to figure out what is the best distribution to return.

# Step 2: existence of distributions on the menu $\mathcal{M}_i$

At level $j$, we use $(A_i^{(j)}, B_i^{(j)})$, $|A_i^{(j)}| = |B_i^{(j)}| = m/2^j$:

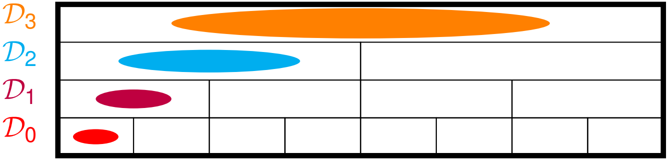



- **Idea:** Assume (by induction) that there is a distribution $\mathcal{D} \in \mathcal{M}_i$ that allocates (in expectation) a good portion of $A_i^{(j)}$ to agent $i$.
- But the sets $(A_i^{(j)}, B_i^{(j)})$ cannot be found efficiently.
- So the mechanism must return a distribution $\mathcal{D}' \in \mathcal{M}_i$ that does not depend on $(A_i^{(j)}, B_i^{(j)})$ and still beats $\mathcal{D}$ in expected profit!
- Such a distribution must be "bigger / cheaper" than $\mathcal{D}$.

# Stage 2: how to combine everything

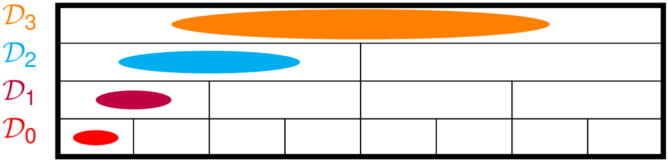
**Sketch of the proof ruling out all T.I.E. mechanisms:**




1. If the mechanism provides a $c$-approximation, there must be a agent $i^*$ and a choice of valuations (a *basic instance*) such that agent $i^*$ receives a $c$-fraction of a random set $A_{i^*}^{(\ell)}$ of size $m/2^\ell$.

# Stage 2: how to combine everything

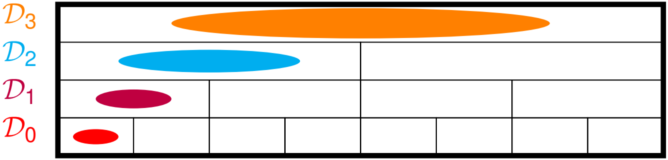
**Sketch of the proof ruling out all T.I.E. mechanisms:**




1. If the mechanism provides a $c$-approximation, there must be a agent $i^*$ and a choice of valuations (a *basic instance*) such that agent $i^*$ receives a $c$-fraction of a random set $A_{i^*}^{(\ell)}$ of size $m/2^\ell$.
2. By the inductive boosting argument + 2-dim convex separation argument (to deal with prices), there are distributions on the menu allocating *larger and larger sets* to agent $i^*$, at *decreasing prices*.

# Stage 2: how to combine everything

**Sketch of the proof ruling out all T.I.E. mechanisms:**




1. If the mechanism provides a $c$-approximation, there must be a agent $i^*$ and a choice of valuations (a *basic instance*) such that agent $i^*$ receives a $c$-fraction of a random set $A^{(\ell)}_{i^*}$ of size $m/2^\ell$.
2. By the inductive boosting argument + 2-dim convex separation argument (to deal with prices), there are distributions on the menu allocating *larger and larger sets* to agent $i^*$, at *decreasing prices*.
3. Eventually, we prove the existence of a distribution on the menu of agent $i^*$ that would be more profitable to him in the basic instance, than what he receives when reporting truthfully $\Rightarrow$ CONTRADICTION.

# Stage 3: oracle hardness ⇒ computational hardness

**What is the issue:** imagine we want to present the function

$$v(S) = 1 - (1 - \alpha|S \cap A|)_+(1 - \alpha|S \cap B|)_+$$

explicitly on the input. We can encode it by specifying $(\alpha, A, B)$. However, then it's easy to determine the "desired set" $A$ (or $B$)!

**What is the issue:** imagine we want to present the function

$$v(S) = 1 - (1 - \alpha|S \cap A|)_+(1 - \alpha|S \cap B|)_+$$

explicitly on the input. We can encode it by specifying $(\alpha, A, B)$. However, then it's easy to determine the "desired set" $A$ (or $B$)!

**Solution:**

- $(A, B)$ is determined by the solution of a computationally difficult problem (i.e. SAT).
- Rather than $(A, B)$, the encoding contains a SAT instance $\phi$.
- *HOWEVER: this encoding must allow us to evaluate the function!*

# Stage 4: encoding by list-decodable codes

**Encoding:**
We encode a valuation by a set $C$, a Unique-SAT formula $\phi$ on $t$ variables, $\alpha > 0$, and a *list-decodable code* $E : \{0,1\}^t \to \{0,1\}^C$.

# Stage 4: encoding by list-decodable codes

**Encoding:**
We encode a valuation by a set $C$, a Unique-SAT formula $\phi$ on $t$ variables, $\alpha > 0$, and a *list-decodable code* $E : \{0,1\}^t \to \{0,1\}^C$.

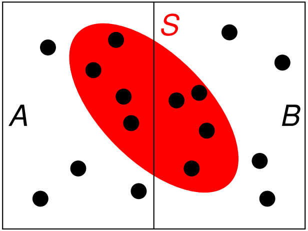
**Interpretation:** $(C, \phi, \alpha, E)$ encodes a *perturbed function* $\tilde{v}(S)$ that we actually use in our hardness proof:

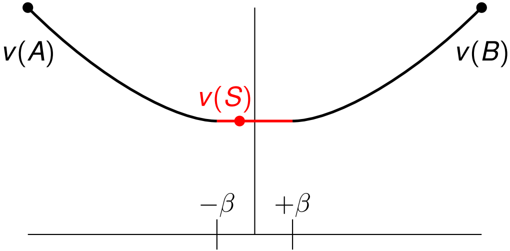
$$\tilde{v}(S) \simeq 1 - (1 - \alpha|S \cap A|)_+(1 - \alpha|S \cap B|)_+$$

where $A = C \setminus B = E(x^*)$, and $x^* =$ unique satisfying assignment to $\phi$.

**Key point:** $\tilde{v}(S)$ depends on the partition $(A, B)$ only if $S$ is "unbalanced" w.r.t. $(A, B)$, in which we are able to determine $(A, B)$ using the list decodable code.

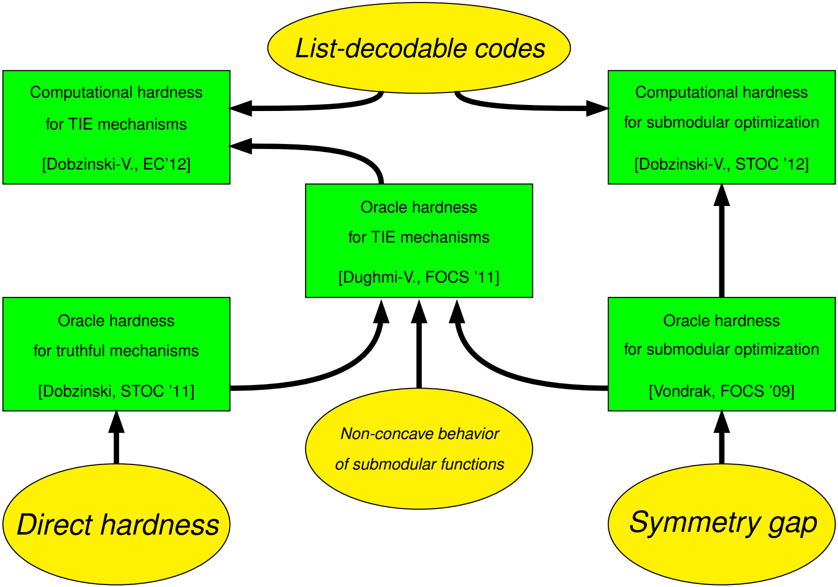
# Stage 4: evaluating $v(S)$ using this representation




- Case 1: If $|S \cap A| - |S \cap B| > \beta$, we find $x^* = E^{-1}(A)$ as one of the codewords obtained by list-decoding $S$. Evaluating $\phi(x^*)$ confirms that $A$ is the correct set.
- Case 2: If $|S \cap B| - |S \cap A| > \beta$, we find $x^* = E^{-1}(A)$ again by list-decoding $\overline{S}$.
- Case 3: If $|S \cap A| - |S \cap B| \in [-\beta, +\beta]$, we are not able to determine $A$ and $B$. But $v(S)$ in this case depends only on $|S|$, so we can still evaluate $v(S)$.

# Overview of techniques



# Summary of results

| Valuations | Without truthfulness | Truthful-in-expectation | Univ. truthful |
|---|---|---|---|
| Coverage | $1 - 1/e$ | $1 - 1/e$ | ??? |
| Submodular (explicit) | $1 - 1/e$ | $n^{-\Omega(1)}$ | $m^{o(1)-1/2}$ |
| Submodular (value oracle) | $1 - 1/e$ | $m^{-\Omega(1)}$ | $m^{o(1)-1/2}$ |

# Summary of results

| Valuations | Without truthfulness | Truthful-in-expectation | Univ. truthful |
|---|---|---|---|
| Coverage | $1 - 1/e$ | $1 - 1/e$ | ??? |
| Submodular (explicit) | $1 - 1/e$ | $n^{-\Omega(1)}$ | $m^{o(1)-1/2}$ |
| Submodular (value oracle) | $1 - 1/e$ | $m^{-\Omega(1)}$ | $m^{o(1)-1/2}$ |

**Some open questions:**

- Hardness for more natural valuation functions?
- Stronger oracle models?
  (demand queries: $\max_S(v(S) - \sum_{j \in S} p_j) = ?$)
- Communication complexity lower bounds?
- Or positive results for more special classes: budget-additive?