Reducing Mechanism to Algorithm Design

Constantinos Daskalakis (MIT)

joint work with: Yang Cai (McGill)
Matt Weinberg (Princeton)

longer tutorial: http://people.csail.mit.edu/costis/EC14/
Algorithm Design

(given) Input \rightarrow \textbf{Algorithm} \rightarrow (desired) Output
Mechanism Design

Agents’ (given) Input -> Algorithm

Mechanism

Algorithm -> Agents’ (desired) Output
E.g. Computing Max

- **Input:** $x_1, x_2, \ldots, x_n$
- **Goal:** compute $\max(x_1, \ldots, x_n)$
- **Algorithm:** Trivial
- But what if inputs are strategic?
  - suppose input $i$ has value $x_i$ for being selected
  - facing trivial algorithm, every input reports $+\infty$
- **A better Algorithm [Vickrey’61]:**
  - collect reported inputs: $b_1, \ldots, b_n$ (can’t enforce $b_i = x_i$ a priori)
  - select $i^* = \arg \max b_i$
  - charge winner $i^*$ the 2nd highest $b_i = \arg \max_{j \neq i^*} b_j$
  - **Claim:** It is in every $i$’s best interest to report $b_i = x_i$.

$\Rightarrow$ **Vickrey auction** is the new $\max$. 
Year: 1797

Goethe’s Problem: sell the royalties of epic poem, Herman and Dorothea, to publisher Vieweg

Goethe’s (Conjectured) Goal: learn how much the royalties are really worth (for selling future royalties)

Asking publisher directly/take-it-or-leave-it offer wouldn’t achieve goal

Instead, Goethe sent the following letter to Vieweg:

“Concerning the royalties we will proceed as follows: I will hand over to Mr. Counsel Bottiger a sealed note which contains my demand, and wait for what Mr. Vieweg will suggest to offer for my work. If his offer is lower than my demand, then I take my note back, unopened, and the negotiation is broken. If, however, his offer is higher, then I will not ask for more than what is written in the note to be opened by Mr. Bottiger.”

essentially Vickrey auction with 2 bidders: Goethe and Vieweg]
Mechanism Design

Agents’ Reports

(given) Input

Mechanism

Agents’ Payoffs

(desired) Output
[Nisan-Ronen’99]:
How much more difficult are optimization problems on “strategic” input compared to “honest” input?

The Dream:

Black-box reduction from mechanism- to algorithm-design for all optimization problems

Want:
Algorithm that works on strategic input

Have:
Algorithm that works on honest input

Output
[Nisan-Ronen’99]: How much more difficult are optimization problems on “strategic” input compared to “honest” input?

The Dream:

Black-box reduction from mechanism- to algorithm-design for all optimization problems
[Nisan-Ronen’99]: How much more difficult are optimization problems on “strategic” input compared to “honest” input?

The Dream:

Black-box reduction from mechanism- to algorithm-design for all optimization problems

Want: Algorithm that works on strategic input

Have: Algorithm that works on honest input
Why Black-Box Reductions?

1) More is known about algorithms than mechanisms
   - Hope: unsolved problems might reduce to already-solved problems

2) Allows larger community to tackle important problems
   - Without necessarily learning game theory

3)Provides deeper understanding of Mechanism Design
   - What makes incentives so difficult to deal with?

[Cai-D-Weinberg’12,’13]: Reduction exists for any optimization problem! (with right qualifications)
The Menu

- Algorithms for Strategic Inputs
- Example Objectives
- Statement of Main Result
- Proof Techniques
The Menu

- Algorithms for Strategic Inputs
- Example Objectives
- Statement of Main Result
- Proof Techniques
Solved Objective: Welfare

- **Setting:** - Several items, several bidders, each bidder $i$ has private value $v_{ij}$ for item $j$
  - Allocation constraints $A$ permit some allocations but not others
- **Goal:** Choose allocation $x \in A$ to optimize $\sum x_{ij} v_{ij}$

[VCG ’73]: Optimal auction:
1. Collect reports $\vec{b}_1, \ldots, \vec{b}_n$
2. Choose $x$ maximizing $\sum x_{ij} b_{ij}$
3. Charge “Clarke payments”
   - ensures $\vec{b}_i \equiv \vec{v}_i$

• Generalizes Vickrey (single-item) auction

Reduction to welfare-optimizing algorithm
- **Solved Objective**: Revenue

- **Setting**: 1 item, several bidders, each bidder $i$ has private value $v_i$ for the item
- **Goal**: Choose allocation $x$ and prices $p_1, \ldots, p_n$ to optimize $\sum_i p_i$ subject to $p_i \leq x_i v_i$

**Reduction to revenue-optimizing algorithm**

1. Collect bids $b_1, \ldots, b_n$
2. For all $i$:
   $$b_i \leftrightarrow b_i - \frac{1-F_i(b_i)}{f_i(b_i)} \equiv \hat{b}_i$$
3. Choose $x$ maximizing $\sum_i x_i \hat{b}_i$
4. Charge “Myerson payments”
   - ensures $b_i = v_i$

- When $F_1 = F_2 = \ldots = F_n$:
  - Myerson’s auction $\equiv$ 2$^{\text{nd}}$ Price auction with reserve price

**Big Challenge**: Revenue-Optimal Multi-Item Mechanisms
Multi-Item Mechanisms

- Online marketplaces
- Sponsored search
- Selling banner ads
- Spectrum auctions
Optimal Multi-Item Auctions

• Large body of work in Economics:
  – e.g. [Laffont-Maskin-Rochet’87], [McAfee-McMillan’88], [Wilson’93], [Armstrong’96],
    [Rochet-Chone’98], [Armstrong’99], [Zheng’00], [Basov’01], [Kazumori’01],
    [Thanassoulis’04], [Vincent-Manelli ’06,’07], [Figalli-Kim-McCann’10], [Pavlov’11], [Hart-
    Nisan’12],...

• Progress slow. No general approach.
• Challenge already with selling 2 items to 1 bidder.
Optimal Multi-Item Auctions

• Large body of work in Economics:
  – e.g. [Laffont-Maskin-Rochet’87], [McAfee-McMillan’88], [Wilson’93], [Armstrong’96], [Rochet-Chone’98], [Armstrong’99], [Zheng’00], [Basov’01], [Kazumori’01], [Thanassoulis’04], [Vincent-Manelli ’06,’07], [Figalli-Kim-McCann’10], [Pavlov’11], [Hart-Nisan’12],...

• Progress slow. No general approach.
• Challenge already with selling 2 items to 1 bidder.
• [D-Deckelbaum-Tzamos’14]: Closed-form characterization of single-bidder multi-item mechanisms.
• Unlikely to exist for multi-bidder settings.

• Recent algorithmic work: Constant Factor Approximations in restricted settings
  – e.g. [Chawla-Hartline-Kleinberg ’07], [Chawla et al’10], [Bhattacharya et al’10], [Alaei’11], [Hart-Nisan ’12], [Kleinberg-Weinberg ’12], [Alaei et al. ’12]

• [Cai-D-Weinberg ’12,’13]:
  Polynomial-time computable Revenue-optimal Multi-Item Auctions in general (Bayesian) settings.
Uncharted Territory: Non-linear Objectives

- [Nisan-Ronen’99] challenge: $n$ strategic machines, $m$ jobs
- Machine $i$ has (private) processing time $p_{ij}$ for job $j$
- **Goal**: choose allocation $x$ of jobs to machines to minimize makespan: $\max_i \sum_j x_{ij} p_{ij}$

**Prior Best:**
- Honest input: factor 2 appx [LST’87]
- Strategic input: factor $n$ appx [NR’99]

[D-Weinberg’14]:
- Strategic input (Bayesian): factor 2 appx
The Menu

- Algorithms for Strategic Inputs
- Example Objectives
- Statement of Main Result
- Proof Techniques
The Menu

- Algorithms for Strategic Inputs
- Example Objectives
- Statement of Main Result
- Proof Techniques
Statement of Main Result

**Mechanism Design**

**Given:**
1. Input set $X$, solution set $S$
2. objective $O : X^n \times S \rightarrow \mathbb{R}$
3. access to strategic agents $1 \ldots n$ s.t. agent $i$:
   • knows the $i^{th}$ input $x_i$
   • has keen interest $x_i(s)$ in our choice of $s \in S$
   • saving grace: $x_i \sim F_i$ for known $F_i$

**Goal:** Find truthful mechanism optimizing objective $O$ (in expectation over $\times F_i$), among all possible mechanisms


**Algorithm Design**

**Given:**
1. same $X$, $S$, $O$
2. known inputs $x_1, \ldots, x_n \in X$

**Goal:** Find $s \in S$ to optimize $O(x_1, \ldots, x_n, s)$

• [Cai-Daskalakis-Weinberg'13]: Polynomial-time reduction from mechanism design for arbitrary objective $O$ to algorithm design for same objective $O$ plus linear cost function (virtual welfare).
  • i.e. if RHS tractable, then LHS tractable
  • also approximation preserving:
    • i.e. $\alpha$-approximation to RHS $\Rightarrow \alpha$-approximation to LHS
**Mechanism Design**

**Given:**
1. Input set $X$, solution set $S$
2. Objective $O : X^n \times S \rightarrow \mathbb{R}$
3. Access to strategic agents $1 \ldots n$ s.t. agent $i$
   - knows the $i^{th}$ input $x_i$
   - has keen interest $x_i(s)$ in our choice of $s \in S$
   - saving grace: $x_i \sim F_i$ for known $F_i$

**Goal:** Find truthful mechanism optimizing objective $O$ (in expectation over $\times F_i$), among all possible mechanisms

---

**Algorithm Design**

**Given:**
1. Same $X$, $S$, $O$
2. Known inputs $x_1, \ldots, x_n \in X$
3. Known inputs $y_1, \ldots, y_n \in X^\pm$

**Goal:** Find $s \in S$ to optimize $O(x_1, \ldots, x_n, s) + \sum_i y_i(s)$

---

**Makespan with costs**

- **Input**
  - $x_1, \ldots, x_n \in \mathbb{R}^m_{\geq 0}$
  - $y_1, \ldots, y_n \in \mathbb{R}^m$

- **Minimization Objective:**
  $$\max_i \sum_j s_{ij} x_{ij} - \sum_j \sum_j s_{ij} y_{ij}$$

---

**Examples**

- **Makespan** ($n$ strategic machines, $m$ jobs)
  - **Input set:** $X = \mathbb{R}^m_{\geq 0}$
  - **Solution set:** $S = \{s : [m] \rightarrow [n]\}$
  - **Agent** $i$
    - knows $x_i = (x_{i1}, \ldots, x_{im}) \sim F_i$
    - has value $-\sum_j s_{ij} x_{ij}$ for schedule $s$

  **Minimization Objective:**
  $$O(x_1, \ldots, x_n ; s) = \max_i \sum_j s_{ij} x_{ij}$$
Main Result

Mechanism Design

**Given:**
1. Input set $\mathcal{X}$, solution set $\mathcal{S}$
2. objective $O : \mathcal{X}^n \times \mathcal{S} \rightarrow \mathbb{R}$
3. access to strategic agents $1...n$ s.t. agent $i$:
   - knows the $i^{th}$ input $x_i$
   - has keen interest $x_i(s)$ in our choice of $s \in \mathcal{S}$
   - saving grace: $x_i \sim F_i$ for known $F_i$

**Goal:** Find truthful mechanism optimizing objective $O$ (in expectation over $\times F_i$), among all possible mechanisms

e.g. Makespan ($n$ strategic machines, $m$ jobs)

- Input set: $\mathcal{X} = \mathbb{R}^m_{\geq 0}$
- Solution set: $\mathcal{S} = \{s : [m] \rightarrow [n]\}$
- Agent $i$:
  - knows $x_i = (x_{i1},...,x_{im}) \sim F_i$
  - has value $-\sum_j s_{ij} x_{ij}$ for schedule $s$

- Minimization Objective: $O(x_1,...,x_n ; s) = \max_i \sum_j s_{ij} x_{ij}$

Algorithm Design

**Given:**
1. same $\mathcal{X}, \mathcal{S}, O$
2. known inputs $x_1,...,x_n \in \mathcal{X}$
3. known inputs $y_1,...,y_n \in \mathcal{X}^\pm$

**Goal:** Find $s \in \mathcal{S}$ to optimize $O(x_1,...,x_n,s) + \sum_i y_i(s)$

Makespan with costs

- Input:
  - $x_1,...,x_n \in \mathbb{R}^m_{\geq 0}$
  - $y_1,...,y_n \in \mathbb{R}^m$

- Minimization Objective:
  $$\max_i \sum_j s_{ij} x_{ij} - \sum_j \sum_i s_{ij} y_{ij}$$
Main Result

e.g. Makespan ($n$ strategic machines, $m$ jobs)

- **Input** set: $\mathcal{X} = \mathbb{R}^m_{\geq 0}$
- **Solution** set: $\mathcal{S} = \{s : [m] \rightarrow [n]\}$
- **Agent** $i$:
  - knows $x_i = (x_{i1},...,x_{im}) \sim F_i$
  - has value $-\sum_j s_{ij} x_{ij}$ for schedule $s$

- **Minimization Objective**:
  $$O(x_1,...,x_n ; s) = \max_i \sum_j s_{ij} x_{ij}$$

Makespan with costs

- **Input**:
  - $x_1,...,x_n \in \mathbb{R}^m_{\geq 0}$
  - $y_1,...,y_n \in \mathbb{R}^m$
- **Minimization Objective**:
  $$\max_i \sum_j s_{ij} x_{ij} + \sum_{i,j} s_{ij} y_{ij}$$

- Our reduction ($\Rightarrow$) says that, if RHS is $\alpha$-approximable in poly-time, then so is LHS
- However, RHS is NP-hard to $\alpha$-approximate, to within any finite $\alpha$
  - unsurprising given the mixed sign objective on the RHS
- Nevertheless, still exist **bi-criterion** approximation algorithms:
  - in poly-time, can find schedule $s$ s.t.
  $$\beta \times \max_i \sum_j s_{ij} x_{ij} + \sum_{i,j} s_{ij} y_{ij} \leq \alpha \times \text{OPT}$$
  [ST’93]: $\alpha=1$, $\beta=0.5$
- Our reduction ($\Rightarrow$) shows that, if RHS is $(\alpha,\beta)$-approximable in poly-time, then LHS is $(\alpha/\beta)$-approximable in poly-time.
- Hence, factor 2 for makespan.
Statement of Main Result

**Mechanism Design**

**Given:**
1. Input set \( \mathcal{X} \), solution set \( \mathcal{S} \)
2. objective \( \mathcal{O} : \mathcal{X}^n \times \mathcal{S} \rightarrow \mathbb{R} \)
3. access to strategic agents \( 1 \ldots n \) s.t. agent \( i \):
   - knows the \( i \)th input \( x_i \)
   - has keen interest \( x_i(s) \) in our choice of \( s \in \mathcal{S} \)
   - **saving grace:** \( x_i \sim F_i \) for known \( F_i \)

**Goal:** Find truthful mechanism optimizing objective \( \mathcal{O} \) (in expectation over \( \times F_i \)), among all possible mechanisms

**Algorithm Design**

**Given:**
1. same \( \mathcal{X} , \mathcal{S} , \mathcal{O} \)
2. known inputs \( x_1 , \ldots , x_n \in \mathcal{X} \)
3. known inputs \( y_1 , \ldots , y_n \in \mathcal{X}^\pm \)

**Goal:** Find \( s \in \mathcal{S} \) to optimize \( \mathcal{O}(x_1, \ldots , x_n , s) + \sum_i y_i (s) \)

- \[\text{Cai-Daskalakis-Weinberg’13}\]: Polynomial-time reduction from mechanism design for arbitrary objective \( \mathcal{O} \) to algorithm design for same objective \( \mathcal{O} \) plus linear cost function.
  - i.e. if RHS tractable, then LHS tractable
  - also approximation sensitive:
    - i.e. \((\alpha, \beta)\)-approximation to RHS \( \Rightarrow (\alpha/\beta)\)-approximation to LHS
- **techniques:** probability and convex programming: approximation-sensitive versions of the equivalence of optimization and separation \[\text{Grötschel-Lovász-Schrijver’80, Karp-Papadimitriou’80}\]
The Menu

- Algorithms for Strategic Inputs
- Example Objectives
- Statement of Main Result
- Proof Idea
**Approach (really sketchy)**

### Mechanism Design

**Given:**
1. Input set \( \mathcal{X} \), solution set \( \mathcal{S} \)
2. Objective \( \mathcal{O}: \mathcal{X}^n \times \mathcal{S} \rightarrow \mathbb{R} \)
3. Access to strategic agents \( 1 \ldots n \) s.t. agent \( i \):
   - knows the \( i \)th input \( x_i \)
   - has keen interest \( x_i(s) \) in our choice of \( s \in \mathcal{S} \)
   - saving grace: \( x_i \sim F_i \) for known \( F_i \)

**Goal:** Find truthful mechanism optimizing objective \( \mathcal{O} \) (in expectation over \( \times F_i \)), among all possible mechanisms

### Algorithm Design

**Given:**
1. Same \( \mathcal{X}, \mathcal{S}, \mathcal{O} \)
2. Known inputs \( x_1, \ldots, x_n \in \mathcal{X} \)
3. Known inputs \( y_1, \ldots, y_n \in \mathcal{X}^\pm \)

**Goal:** Find \( s \in \mathcal{S} \) to optimize
\[ \mathcal{O}(x_1, \ldots, x_n, s) + \sum_i y_i(s) \]

Write Configuration LP
- incentives, feasibility linear
issue: exponential size

(distribution over \( \mathcal{S} \) for every possible type profile)

Compress to Polynomial Size (non-configuration) LP
issue: feasibility hard to check

need separation oracle
Approximate Equivalence of Separation and Optimization?

- **Question:** If \( P \) is a convex region and \( A \) is an \( a \)-approximation algorithm satisfying \( w^T \cdot A(w) \geq a \cdot \max_{y \in P} \{ w^T \cdot y \} \), can we get a meaningful approximate separation oracle for \( P \) when our only access to \( P \) is via \( A \)?

- **[GLS’80, KP’80] framework breaks down:** Roughly, unless all directions are queried, we cannot separate out a convex region inside \( P \).

- **Our Approach:** Forget about convex regions.

  - Plug in approximation algorithm inside [GLS’80, KP’80] framework to get **weird separation oracle** (accepting a non-convex region).
  - Show that using WSO inside Ellipsoid doesn’t hurt Ellipsoid!
  - in particular, recover feasible solution with value at least \( \max_{y \in \alpha P \cap Q} \{ w^T \cdot y \} \)
Approximate Equivalence of Separation and Optimization

**Question:** If $P$ is a convex region and $A$ is an $a$-approximation algorithm satisfying $\mathbf{w}^T \cdot A(\mathbf{w}) \geq a \cdot \max_{y \in P} \{\mathbf{w}^T \cdot y\}$, can we get a meaningful approximate separation oracle for $P$ when our only access to $P$ is via $A$?

**[GLS’80, KP’80] framework breaks down:** Roughly, unless all directions are queried, we cannot separate out a convex region inside $P$.

**Our Approach:** Forget about convex regions.

Plug in approximation algorithm inside [GLS’80, KP’80] framework to get **weird separation oracle** (accepting a non-convex region).

Show that using WSO inside Ellipsoid doesn’t hurt Ellipsoid!

in particular, recover **feasible** solution with value at least $\max_{y \in a \cap Q} \{\mathbf{w}^T \cdot y\}$

$= P_a^- = \{x \mid \mathbf{w} \cdot x \leq \mathbf{w} \cdot A(\mathbf{w}), \forall \mathbf{w}\}$

$= \text{yes}$

$= P_a^+ = \text{Conv}(A(\mathbf{w}) \mid \mathbf{w} \in \mathbb{R}^d)$
Conclusions/Further Questions

- **This Talk:** Optimization on strategic input
- VCG auction optimizes welfare in all settings
- Myerson’s auction maximizes revenue in single-item settings
- Revenue maximization in multi-item settings is challenging:
  - optimal auctions have richer structure
  - lots of work in Economics, TCS
- Ditto for mechanism design for non-linear objectives
- We obtain a generic, poly-time algorithm design
  - “optimizing on strategic inputs is no harder than adding linear cost to your objective”
- As corollaries of our framework we obtain:
  - revenue-optimal multi-item auctions
  - approximately optimal mechanisms for non-linear objectives such as makespan

\[(\alpha,\beta)\)-approximation algorithms for objectives + linear cost?
Conclusions/Further Questions

- **Structural Understanding of Optimal Mechanisms**
- **Example:** $n$ bidders, $m$ houses
  - Bidder $i$ has values $(v_{i1}, \ldots, v_{im}) \sim F_i$ for getting each house
  - Bidders are independent
  - Objective: revenue optimization

Optimal auction:
1. Collect reports $\tilde{b}_1, \ldots, \tilde{b}_n$
2. For all $i$: $\tilde{w}_i = h_i(\tilde{b}_i)$
3. Find max-weight allocation
4. Serve agents involved in this allocation

$\tilde{v}_1 \sim F_1$

$\vdots$

$\tilde{v}_i \sim F_i$

$\vdots$

$\tilde{v}_n \sim F_n$

$\tilde{b}_1$

$\vdots$

$\tilde{b}_n$

$h_i : \mathbb{R}^m \rightarrow \mathbb{R}^m$

(depending on $F_1, \ldots, F_n$)

Better understanding of virtual transformations $h_i$?

Thanks!