CS599: Algorithm Design in Strategic Settings
Fall 2012
Lecture 9: Prior-Free Multi-Parameter Mechanism Design (Continued)

Instructor: Shaddin Dughmi

Administrivia

- HW2 Out, due in two weeks
- Projects
  - Meetings
  - Partners
- Mini Homeworks graded. Pick up.

Outline

1. Review
2. Rounding Anticipation
3. Characterizations of Incentive Comapatibility
   - Direct Characterization
   - Characterizing the Allocation rule
4. Lower Bounds in Prior Free AMD

Outline

1. Review
2. Rounding Anticipation
3. Characterizations of Incentive Comapatibility
   - Direct Characterization
   - Characterizing the Allocation rule
4. Lower Bounds in Prior Free AMD

Recall: Mechanism Design Problem in Quasi-linear Settings

Public (common knowledge) inputs describes

- Set Ω of allocations.
- Typespace T_i for each player i.
  - $T = T_1 \times T_2 \times \ldots \times T_n$
- Valuation map $v_i : T_i \times \Omega \to \mathbb{R}$

Recall: Mechanism Design Problem in Quasi-linear Settings

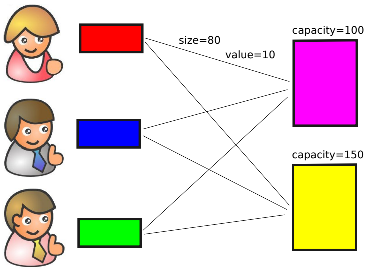
Public (common knowledge) inputs describes

- Set Ω of allocations.
- Typespace T_i for each player i.
  - $T = T_1 \times T_2 \times \ldots \times T_n$
- Valuation map $v_i : T_i \times \Omega \to \mathbb{R}$

Terminology Note

- When convenient, we think of the typespace T_i directly as a set functions mapping outcomes to the real numbers — i.e. $T_i \subseteq \mathbb{R}^\Omega$.
- In that case, we prefer denoting the typespace of player i by $\mathcal{V}_i \subseteq \mathbb{R}^\Omega$. Analogously, the set of valuation profiles is $\mathcal{V} = \mathcal{V}_1 \times \ldots \times \mathcal{V}_n$.
- We refer to $\mathcal{V}_i$ also as the “valuation space” of player i, and each $v_i \in \mathcal{V}_i$ as a “private valuation” of player i.

Example: Generalized Assignment

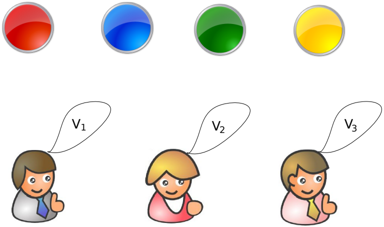


- n self-interested agents (the players), m machines.
- s_{ij} is the size of player i's task on machine j. (public)
- C_j is machine j's capacity. (public)
- $v_i(j)$ is player i's value for his task going on machine j. (private)

Goal

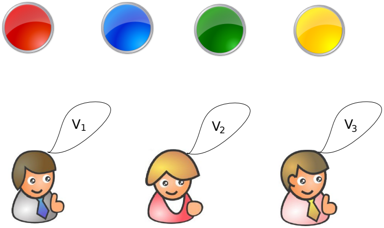
Partial assignment of jobs to machines, respecting machine budgets, and maximizing total value of agents (welfare).

Example: Combinatorial Allocation



- n players, m items.
- Private valuation v_i : set of items $\rightarrow \mathbb{R}$.
  - $v_i(S)$ is player i's value for bundle S.

Example: Combinatorial Allocation



- n players, m items.
- Private valuation v_i : set of items $\rightarrow \mathbb{R}$.
    - $v_i(S)$ is player i's value for bundle S.

Goal

Partition items into sets $S_1, S_2, \ldots, S_n$ to maximize welfare:
$v_1(S_1) + v_2(S_2) + \ldots v_n(S_n)$

Note: This is underspecified. We consider families of restricted valuations with a succinct representation.

Recall: Maximal in Distributional Range

Maximal-in-Distributional-Range (MIDR)

Al allocation rule $f : \mathcal{V}_1 \times \ldots \times \mathcal{V}_n \to \Omega$ is maximal in distributional range if there exists a set $\mathcal{R} \subseteq \Delta(\Omega)$, known as the distributional range of f, such that

$$f(v_1, \ldots, v_n) \sim \underset{D \in \mathcal{R}}{\operatorname{argmax}} \underset{\omega \sim D}{\mathbf{E}} \sum_i v_i(\omega)$$

Recall: Maximal in Distributional Range

Maximal-in-Distributional-Range (MIDR)

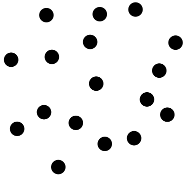
Al allocation rule $f : \mathcal{V}_1 \times \ldots \times \mathcal{V}_n \to \Omega$ is maximal in distributional range if there exists a set $\mathcal{R} \subseteq \Delta(\Omega)$, known as the distributional range of f, such that

$$f(v_1, \ldots, v_n) \sim \underset{D \in \mathcal{R}}{\operatorname{argmax}} \underset{\omega \sim D}{\mathbf{E}} \sum_i v_i(\omega)$$

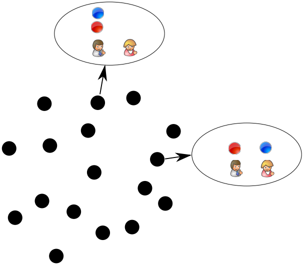
In Other Words

Such an allocation rule samples a distribution in $\mathcal{R}$ maximizing expected social welfare. Maximal in range allocation rules are the special case of MIDR when $\mathcal{R}$ is a family of point distributions.

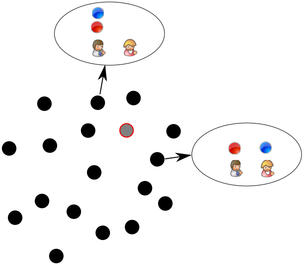
Recall: Maximal in Distributional Range



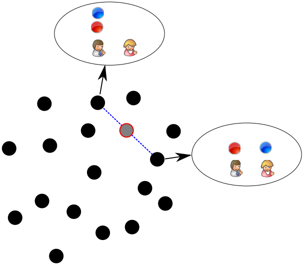
Recall: Maximal in Distributional Range



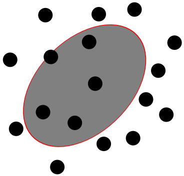
Recall: Maximal in Distributional Range



Recall: Maximal in Distributional Range



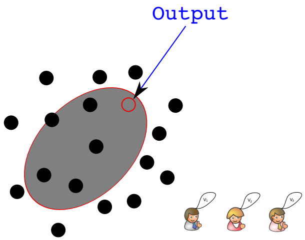
Recall: Maximal in Distributional Range



Maximal in Distributional Range

1. Fix subset $\mathcal{R}$ of distributions over allocations up-front, called the distributional range.
   - Independent of player valuations

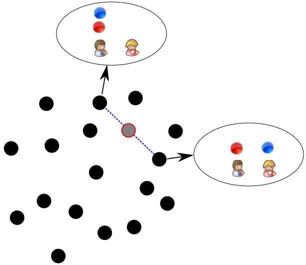
Recall: Maximal in Distributional Range



Maximal in Distributional Range

1. Fix subset $\mathcal{R}$ of distributions over allocations up-front, called the distributional range.
   - Independent of player valuations
2. Given player values, find the distribution in $\mathcal{R}$ maximizing expected social welfare.

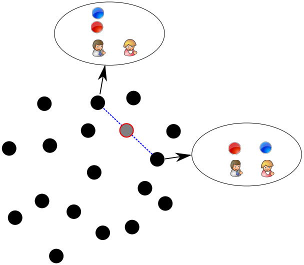
Recall: Maximal in Distributional Range



Maximal in Distributional Range

1. Fix subset $\mathcal{R}$ of distributions over allocations up-front, called the distributional range.
   - Independent of player valuations
2. Given player values, find the distribution in $\mathcal{R}$ maximizing expected social welfare.
3. Sample this distribution

Recall: Maximal in Distributional Range



Maximal in Distributional Range

1. Fix subset $\mathcal{R}$ of distributions over allocations up-front, called the distributional range.
   - Independent of player valuations
2. Given player values, find the distribution in $\mathcal{R}$ maximizing expected social welfare.
3. Sample this distribution

Special case with $\mathcal{R} \subseteq \Omega$ called Maximal-in-Range.

Recall: Maximal in Distributional Range

Fact

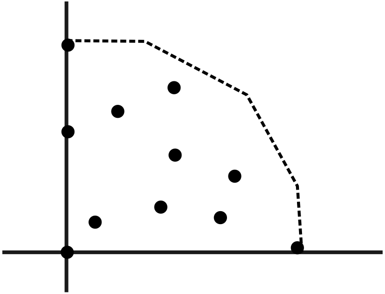
For any mechanism design problem, every maximal in distributional range allocation rule is implementable in dominant-strategies by plugging into VCG. Moreover, if the MIDR algorithm runs in polynomial time, then so does the resulting dominant-strategy truthful mechanism.

Upshot

For NP-hard welfare maximization mechanism design problems (such as GAP, CA, and others), this reduces the design of dominant-strategy truthful, polynomial-time mechanisms to the design of a polynomial-time MIDR allocation algorithms with the desired approximation ratio.

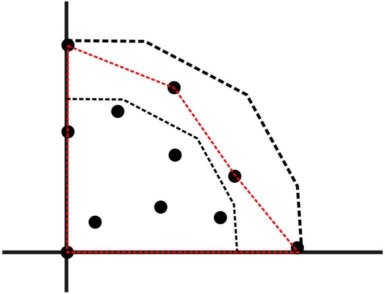
Last Time: The Lavi Swamy Technique

- Considers welfare maximization mechanism design problems.
- Reduces the design of polynomial-time MIDR mechanisms to the design of linear programming relaxations, and accompanying approximation algorithms, satisfying certain conditions.
- Applied to the generalized assignment problem



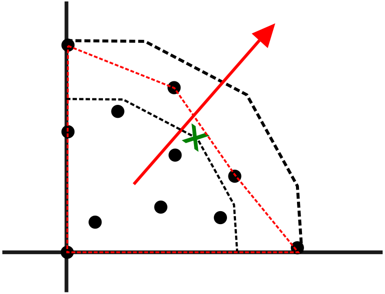
Last Time: The Lavi Swamy Technique

- Considers welfare maximization mechanism design problems.
- Reduces the design of polynomial-time MIDR mechanisms to the design of linear programming relaxations, and accompanying approximation algorithms, satisfying certain conditions.
- Applied to the generalized assignment problem



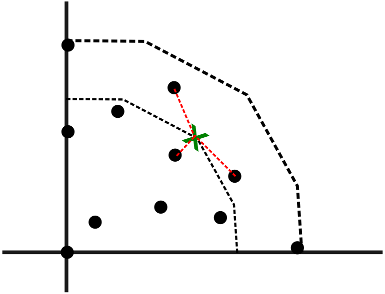
Last Time: The Lavi Swamy Technique

- Considers welfare maximization mechanism design problems.
- Reduces the design of polynomial-time MIDR mechanisms to the design of linear programming relaxations, and accompanying approximation algorithms, satisfying certain conditions.
- Applied to the generalized assignment problem



Last Time: The Lavi Swamy Technique

- Considers welfare maximization mechanism design problems.
- Reduces the design of polynomial-time MIDR mechanisms to the design of linear programming relaxations, and accompanying approximation algorithms, satisfying certain conditions.
- Applied to the generalized assignment problem



Coming Up Today

- Rounding anticipation and the convex rounding technique
- Characterizations of incentive compatibility
- Overview of lower bounds

Outline

1. Review
2. Rounding Anticipation
3. Characterizations of Incentive Comapatibility
   - Direct Characterization
   - Characterizing the Allocation rule
4. Lower Bounds in Prior Free AMD

Overview

- Adapts traditional relax-solve-round framework from approximation algorithms to mechanism design.
- As discussed, MIDR requires exactly solving a sub-problem.
- Whereas relaxations can usually be solved exactly, rounding breaks “maximality-in-range.”

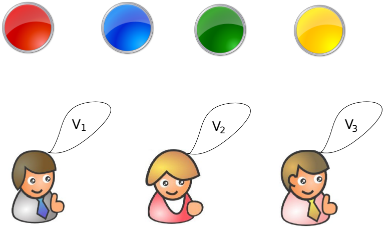
Overview

- Adapts traditional relax-solve-round framework from approximation algorithms to mechanism design.
- As discussed, MIDR requires exactly solving a sub-problem.
- Whereas relaxations can usually be solved exactly, rounding breaks “maximality-in-range.”

Idea: Rounding Anticipation

Anticipate the effect of the rounding algorithm when solving the relaxation, so that solving the relaxation then rounding is MIDR.

Running Application: Combinatorial Allocation



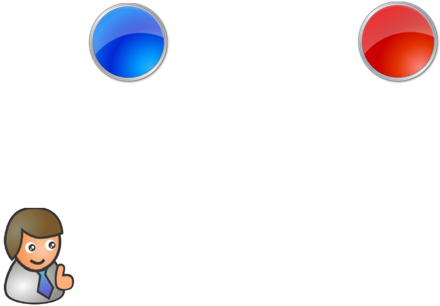
- n players, m items.
- Private valuation v_i : set of items $\rightarrow \mathbb{R}$.
  - $v_i(S)$ is player i's value for bundle S.

Goal

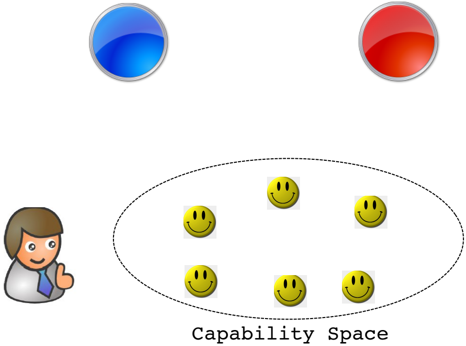
Partition items into sets $S_1, S_2, \ldots, S_n$ to maximize welfare:
$v_1(S_1) + v_2(S_2) + \ldots v_n(S_n)$

As before, we will consider CA with coverage valuations.

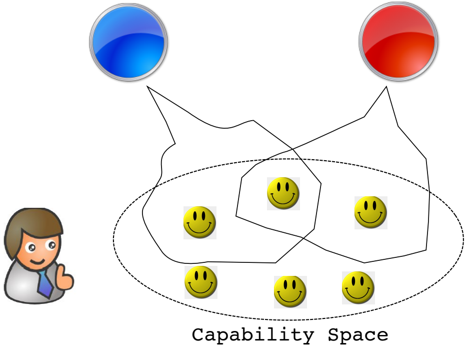
Recall: Coverage Valuations



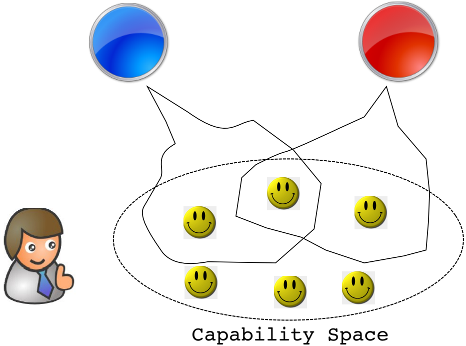
Recall: Coverage Valuations



Recall: Coverage Valuations



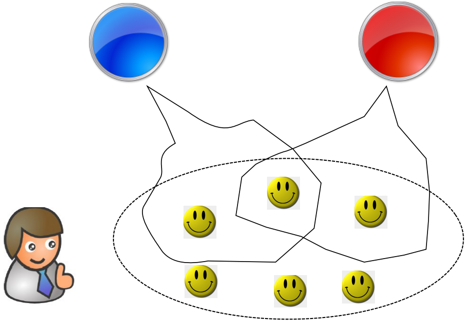
Recall: Coverage Valuations



Recall

Two lectures ago, we used MIR to design a truthful $\sqrt{m}$-approximation mechanism.

Recall: Coverage Valuations



Recall

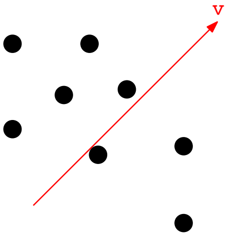
Two lectures ago, we used MIR to design a truthful $\sqrt{m}$-approximation mechanism.

This Time

Using MIDR, via this idea of rounding anticipation, we improve this to a constant, namely $1 - \frac{1}{e} \approx 0.63$.

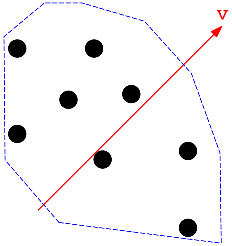
Relax-Solve-Round Framework

Given an optimization problem over some discrete set Ω.



Relax-Solve-Round Framework

Given an optimization problem over some discrete set Ω.

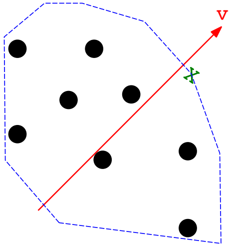


Approximation Algorithm

1. Relax to a linear or convex program over polytope $\mathcal{P}$.

Relax-Solve-Round Framework

Given an optimization problem over some discrete set Ω.

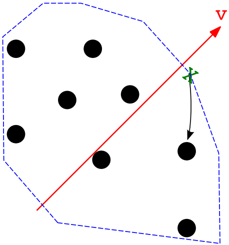


Approximation Algorithm

1. Relax to a linear or convex program over polytope $\mathcal{P}$.
2. Solve the relaxed problem

Relax-Solve-Round Framework

Given an optimization problem over some discrete set Ω.

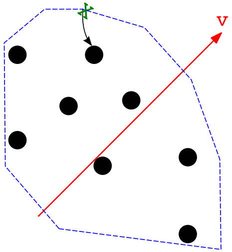


Approximation Algorithm

1. Relax to a linear or convex program over polytope $\mathcal{P}$.
2. Solve the relaxed problem
3. Round the fractional solution to an integral one
   - (Randomized) Rounding scheme $r : \mathcal{P} \to \Omega$.

Relax-Solve-Round Framework

Given an optimization problem over some discrete set Ω.

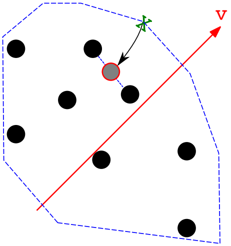


Approximation Algorithm

1. Relax to a linear or convex program over polytope $\mathcal{P}$.
2. Solve the relaxed problem
3. Round the fractional solution to an integral one
   - (Randomized) Rounding scheme $r : \mathcal{P} \to \Omega$.

Relax-Solve-Round Framework

Given an optimization problem over some discrete set Ω.

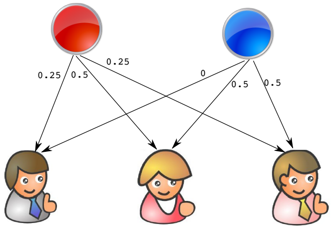


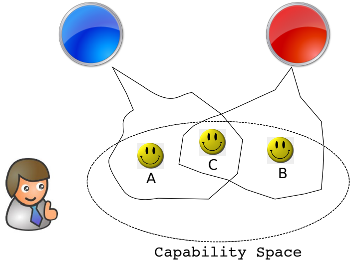
Approximation Algorithm

1. Relax to a linear or convex program over polytope $\mathcal{P}$.
2. Solve the relaxed problem
3. Round the fractional solution to an integral one
   - (Randomized) Rounding scheme $r : \mathcal{P} \to \Omega$.

Example of Relax-Solve-Round: CA

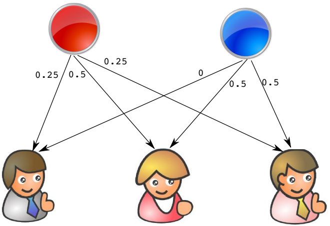
$$
\begin{array}{lll}
\text{maximize} & \sum_{i,A} \min(1, \sum_{\text{j covers A}} x_{ij}) & \\
\text{subject to} & \sum_i x_{ij} \leq 1, & \text{for all } j. \\
 & x_{ij} \geq 0, & \text{for all } i, j.
\end{array}
$$

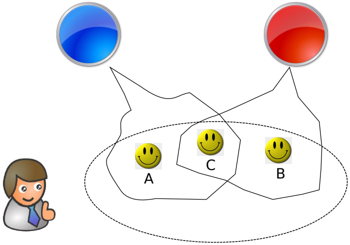




Example of Relax-Solve-Round: CA

$$
\begin{array}{lll}
\text{maximize} & \sum_{i,A} \min(1, \sum_{j \text{ covers } A} x_{ij}) & \\
\text{subject to} & \sum_i x_{ij} \le 1, & \text{for all } j. \\
 & x_{ij} \ge 0, & \text{for all } i, j.
\end{array}
$$





Observe

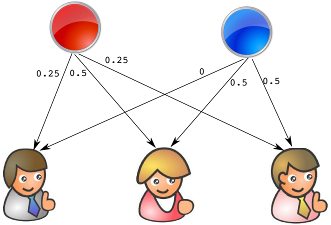
The objective is concave, and this is a convex optimization problem solvable in polynomial time via the ellipsoid method.

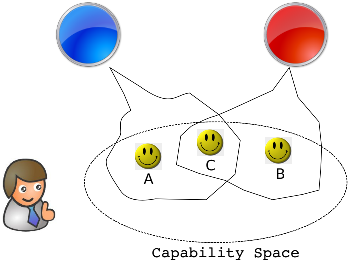
Example of Relax-Solve-Round: CA

$$\text{maximize} \quad \sum_{i,A} \min(1, \sum_{j \text{ covers } A} x_{ij})$$

$$\text{subject to} \quad \sum_i x_{ij} \le 1, \qquad \text{for all } j.$$

$$x_{ij} \ge 0, \qquad \text{for all } i, j.$$





But. . .

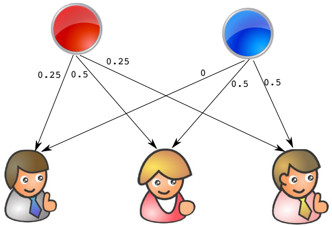
The resulting optimal solution x^* may be fractional, in general.

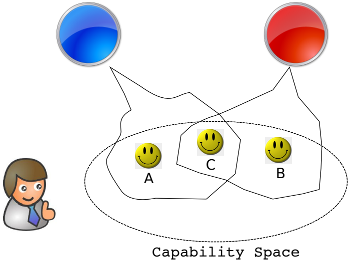
Example of Relax-Solve-Round: CA

maximize $\sum_{i,A} \min(1, \sum_{\text{j covers A}} x_{ij})$

subject to $\sum_i x_{ij} \leq 1,$ for all j.

$x_{ij} \geq 0,$ for all i, j.



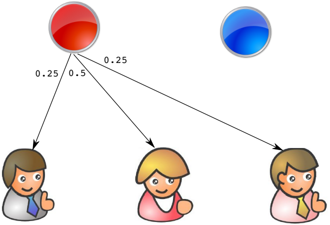


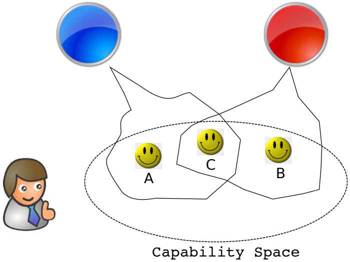
Classical Independent Rounding algorithm

Independently for each item j, give j to player i with probability x^*_{ij}.

Example of Relax-Solve-Round: CA

$$
\begin{array}{lll}
\text{maximize} & \sum_{i,A} \min(1, \sum_{j \text{ covers } A} x_{ij}) & \\
\text{subject to} & \sum_i x_{ij} \le 1, & \text{for all } j. \\
 & x_{ij} \ge 0, & \text{for all } i, j.
\end{array}
$$



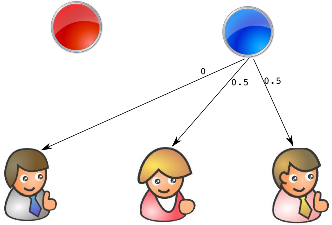


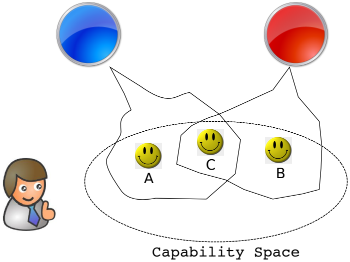
Classical Independent Rounding algorithm

Independently for each item j, give j to player i with probability x_{ij}^*.

Example of Relax-Solve-Round: CA

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{i,A} \min(1, \sum_{\text{j covers A}} x_{ij}) & \\ \text{subject to} \quad & \textstyle\sum_i x_{ij} \leq 1, & \text{for all } j. \\ & x_{ij} \geq 0, & \text{for all } i, j. \end{aligned}$$





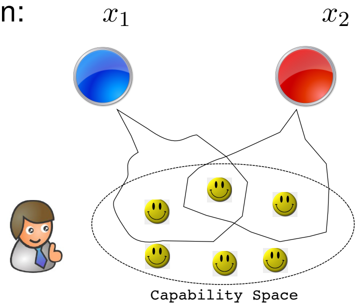
Classical Independent Rounding algorithm

Independently for each item j, give j to player i with probability x^*_{ij}.

Fact

classical independent rounding of the optimal fractional solution gives a $(1 - 1/e)$-approximation algorithm for welfare maximization.

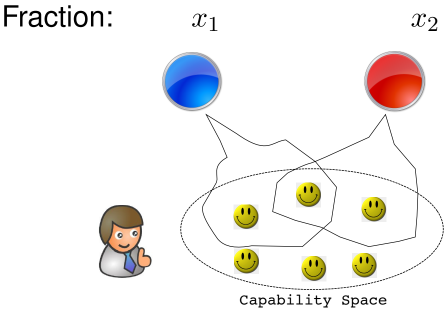
Fraction:



- Fix solution x and player i

Fact

classical independent rounding of the optimal fractional solution gives a $(1-1/e)$-approximation algorithm for welfare maximization.



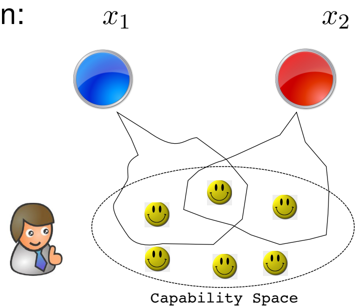
- Fix solution x and player i
- Suffices to show that each capability A covered with probability at least

$$(1-1/e)\min(1, \sum_{\text{j covers A}} x_{ij})$$

Fact

classical independent rounding of the optimal fractional solution gives a $(1 - 1/e)$-approximation algorithm for welfare maximization.

Fraction:



- Fix solution x and player i
- Suffices to show that each capability A covered with probability at least

$$(1-1/e)\min(1, \sum_{\text{j covers A}} x_{ij})$$

$$Pr[\text{cover A}] = 1 - \prod_{\text{j covers A}} (1 - x_j) \geq 1 - \prod_{\text{j covers A}} e^{-x_j}$$
$$= 1 - \exp(-\sum_{\text{j covers A}} x_j) \geq (1 - 1/e) \sum_{\text{j covers A}} x_j$$

Approximation and Truthfulness

Difficulty

Most approximation algorithms in this framework not MIDR, and hence cannot be made truthful.

- Due to “lack of structure” in rounding step.

Approximation and Truthfulness

Difficulty

Most approximation algorithms in this framework not MIDR, and hence cannot be made truthful.

- Due to “lack of structure” in rounding step.

Another Difficulty

The Lavi-Swamy approach does not seem to apply here.

- Welfare is non-linear in encoding of solutions
- Interpreting a fractional solution as a distribution over integer solutions (i.e. rounding) is no longer loss-less
  - Optimize over a set of P of fractional solutions is no longer equivalent to optimizing over corresponding distributions $\{D_x : x \in P\}$.

Proposal: Anticipate the Rounding

Algorithm

1. Relax:
$$\begin{array}{ll} \text{maximize} & welfare(x) \\ \text{subject to} & x \in \mathcal{P} \end{array}$$
2. Solve: Let x^* be the optimal solution of relaxation.
3. Round: Output $r(x^*)$

- Usually, we solve the relaxation then round the fractional solution
- As we discussed, the rounding "disconnects" the fractional optimization problem over P from the MIDR optimization problem over $\{r(x) : x \in P\}$

Proposal: Anticipate the Rounding

Algorithm

1. Relax: maximize ~~$welfare(x)$~~ $welfare(r(x))$

   subject to $x \in \mathcal{P}$
2. Solve: Let x^* be the optimal solution of relaxation.
3. Round: Output $r(x^*)$

- Usually, we solve the relaxation then round the fractional solution
- As we discussed, the rounding “disconnects” the fractional optimization problem over P from the MIDR optimization problem over $\{r(x) : x \in P\}$
- Instead, incorporate rounding into the objective

Proposal: Anticipate the Rounding

Algorithm

1. Relax: maximize ~~$welfare(x)$~~ $welfare(r(x))$
   subject to $x \in \mathcal{P}$
2. Solve: Let x^* be the optimal solution of relaxation.
3. Round: Output $r(x^*)$

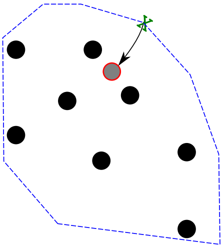
- Usually, we solve the relaxation then round the fractional solution
- As we discussed, the rounding "disconnects" the fractional optimization problem over P from the MIDR optimization problem over $\{r(x) : x \in P\}$
- Instead, incorporate rounding into the objective
- Find fractional solution with best rounded image

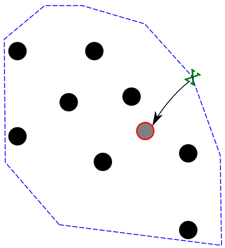
Proposal: Anticipate the Rounding

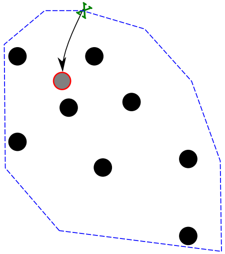
Algorithm

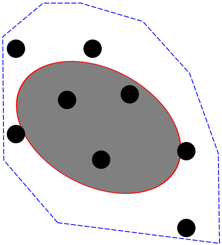
1. Relax:
   maximize $\sout{welfare(x)}$ $\mathrm{E}[welfare(r(x))]$
   subject to $x \in \mathcal{P}$
2. Solve: Let x^* be the optimal solution of relaxation.
3. Round: Output $r(x^*)$

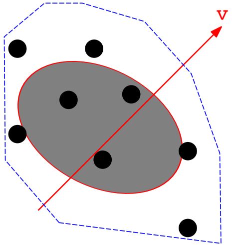
- Usually, we solve the relaxation then round the fractional solution
- As we discussed, the rounding “disconnects” the fractional optimization problem over P from the MIDR optimization problem over $\{r(x) : x \in P\}$
- Instead, incorporate rounding into the objective
- Find fractional solution with best <u>rounded image</u>

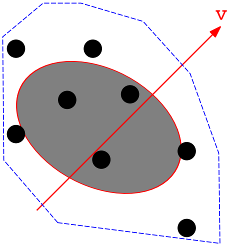








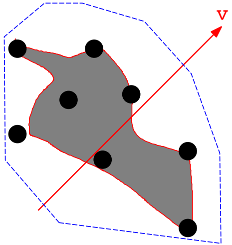

v



Lemma

For any rounding scheme r, this algorithm is maximal in distributional range.

Maximizing over the range of rounding scheme r.



Lemma

For any rounding scheme r, this algorithm is maximal in distributional range.

Maximizing over the range of rounding scheme r.

Difficulty

For most traditional rounding schemes r, this is NP-hard.

NP-Hardness of Anticipating classical independent rounding

- $r(x) = x$ for every integer solution x

NP-Hardness of Anticipating classical independent rounding

- $r(x) = x$ for every integer solution x
- The distributional range $\{r(x) : x \in \mathcal{P}\}$ includes integer solutions

NP-Hardness of Anticipating classical independent rounding

- $r(x) = x$ for every integer solution x
- The distributional range $\{r(x) : x \in \mathcal{P}\}$ includes integer solutions
- The MIDR allocation rule is NP-hard

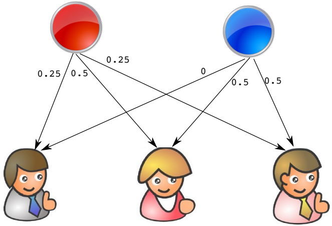
NP-Hardness of Anticipating classical independent rounding

- $r(x) = x$ for every integer solution x
- The distributional range $\{r(x) : x \in \mathcal{P}\}$ includes integer solutions
- The MIDR allocation rule is NP-hard

Next Up

A rounding algorithm which is easier to anticipate!!!

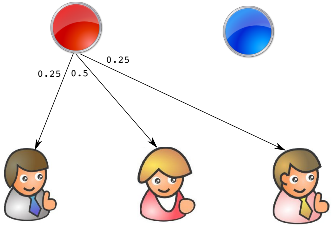
Rounding Algorithms for CA



Classical Independent Rounding (x)

Independently for each item j, give j to player i with probability x_{ij}.

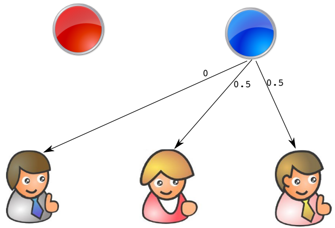
Rounding Algorithms for CA



Classical Independent Rounding (x)

Independently for each item j, give j to player i with probability x_{ij}.

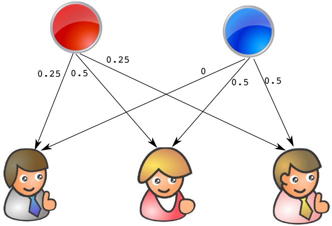
Rounding Algorithms for CA



Classical Independent Rounding (x)

Independently for each item j, give j to player i with probability x_{ij}.

Rounding Algorithms for CA

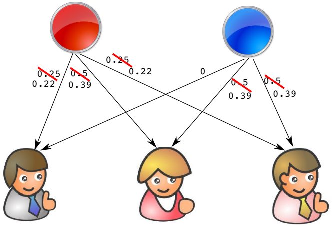


Classical Independent Rounding (x)

Independently for each item j, give j to player i with probability x_{ij}.

Optimizing $welfare(r(x))$ over all $x \in \mathcal{P}$ is NP-hard.

Rounding Algorithms for CA



Classical Independent Rounding (x)

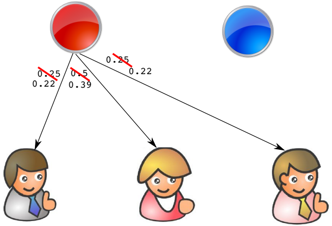
Independently for each item j, give j to player i with probability x_{ij}.

Optimizing $welfare(r(x))$ over all $x \in \mathcal{P}$ is NP-hard.

Poisson Rounding (x)

Independently for each item j, give j to player i with probability $1 - e^{-x_{ij}}$.

Rounding Algorithms for CA



Classical Independent Rounding (x)

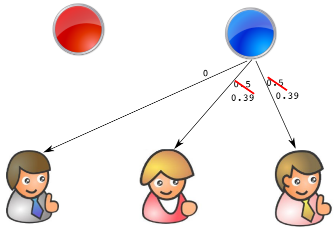
Independently for each item j, give j to player i with probability x_{ij}.

Poisson Rounding (x)

Independently for each item j, give j to player i with probability $1 - e^{-x_{ij}}$.

Optimizing $welfare(r(x))$ over all $x \in \mathcal{P}$ is NP-hard.

Rounding Algorithms for CA



Classical Independent Rounding (x)

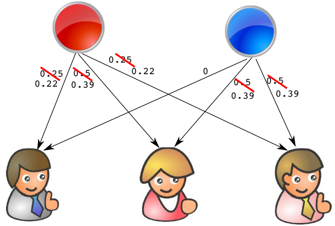
Independently for each item j, give j to player i with probability x_{ij}.

Poisson Rounding (x)

Independently for each item j, give j to player i with probability $1 - e^{-x_{ij}}$.

Optimizing $welfare(r(x))$ over all $x \in \mathcal{P}$ is NP-hard.

Rounding Algorithms for CA



Classical Independent Rounding (x)

Independently for each item j, give j to player i with probability x_{ij}.

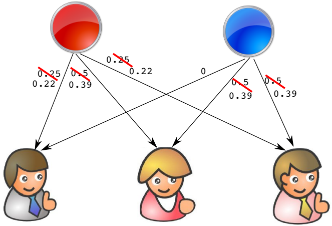
Optimizing $welfare(r(x))$ over all $x \in \mathcal{P}$ is NP-hard.

Poisson Rounding (x)

Independently for each item j, give j to player i with probability $1 - e^{-x_{ij}}$.

Can optimize $welfare(r(x))$ over $x \in \mathcal{P}$ in polynomial time!

Rounding Algorithms for CA



Classical Independent Rounding (x)

Independently for each item j, give j to player i with probability x_{ij}.

Optimizing $welfare(r(x))$ over all $x \in \mathcal{P}$ is NP-hard.

Poisson Rounding (x)

Independently for each item j, give j to player i with probability $1 - e^{-x_{ij}}$.

Can optimize $welfare(r(x))$ over $x \in \mathcal{P}$ in polynomial time!

Note: $(1 - \frac{1}{e})x \leq 1 - e^{-x} \leq x$

Proof Overview

Theorem (Dughmi, Roughgarden, and Yan '11)

There is a polynomial time, $1 - \frac{1}{e}$ approximate, MIDR algorithm for combinatorial auctions with coverage valuations.

Proof Overview

Theorem (Dughmi, Roughgarden, and Yan '11)

There is a polynomial time, $1 - \frac{1}{e}$ approximate, MIDR algorithm for combinatorial auctions with coverage valuations.

Lemma (Polynomial-time solvability)

The expected welfare of rounding $x \in \mathcal{P}$ is a concave function of x.

Implies that finding the rounding-optimal fractional solution is a convex optimization problem, solvable in polynomial time*.

Proof Overview

Theorem (Dughmi, Roughgarden, and Yan '11)

There is a polynomial time, $1 - \frac{1}{e}$ approximate, MIDR algorithm for combinatorial auctions with coverage valuations.

Lemma (Polynomial-time solvability)

The expected welfare of rounding $x \in \mathcal{P}$ is a concave function of x.

Implies that finding the rounding-optimal fractional solution is a convex optimization problem, solvable in polynomial time*.

Lemma (Approximation)

For every set of coverage valuations and integer solution $y \in \mathcal{P}$,

$$welfare(r(y)) \geq (1 - \frac{1}{e}) welfare(y)$$

Implies that optimizing welfare of rounded solution over $\mathcal{P}$ gives a $(1 - \frac{1}{e})$-approximation algorithm.

Proof: Polynomial-time Solvability

Proof.

- Fix fractional solution $\{x_{ij}\}_{ij}$
  - x_{ij} is fraction of item j given to player i.

Proof: Polynomial-time Solvability

Proof.

- Fix fractional solution $\{x_{ij}\}_{ij}$
  - x_{ij} is fraction of item j given to player i.
- Poisson rounding gives j to i with probability $1 - e^{-x_{ij}}$.

Proof: Polynomial-time Solvability

Proof.

- Fix fractional solution $\{x_{ij}\}_{ij}$
  - x_{ij} is fraction of item j given to player i.
- Poisson rounding gives j to i with probability $1 - e^{-x_{ij}}$.
- Let random variable S_i denote set given to i.
- Want to show that $\mathbf{E}[\sum_i v_i(S_i)]$ is concave in variables x_{ij}.

Proof: Polynomial-time Solvability

Proof.

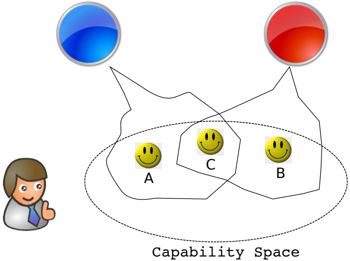
- Fix fractional solution $\{x_{ij}\}_{ij}$
  - x_{ij} is fraction of item j given to player i.
- Poisson rounding gives j to i with probability $1 - e^{-x_{ij}}$.
- Let random variable S_i denote set given to i.
- Want to show that $\mathbf{E}[\sum_i v_i(S_i)]$ is concave in variables x_{ij}.
- By linearity of expectations and the fact concavity is preserved by sum, suffices to show $\mathbf{E}[v_i(S_i)]$ is concave for fixed player i.

□

Proof: Polynomial-time Solvability

Fraction: x_1 x_2

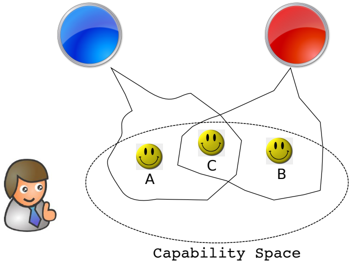
Probability: $1 - e^{-x_1}$ $1 - e^{-x_2}$



Proof: Polynomial-time Solvability

Fraction: x_1 x_2

Probability: $1 - e^{-x_1}$ $1 - e^{-x_2}$

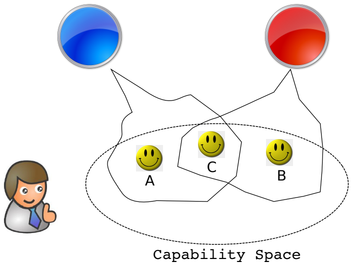


- Value= $\mathbf{Pr}$[Cover A] $+$ $\mathbf{Pr}$[Cover B] $+$ $\mathbf{Pr}$[Cover C]
- Suffices to show each term concave

Proof: Polynomial-time Solvability

Fraction: x_1 x_2

Probability: $1 - e^{-x_1}$ $1 - e^{-x_2}$



- Value= $\mathbf{Pr}[\text{Cover A}] + \mathbf{Pr}[\text{Cover B}] + \mathbf{Pr}[\text{Cover C}]$
- Suffices to show each term concave

$$\mathbf{Pr}[\text{Cover A}] = 1 - e^{-x_1}$$

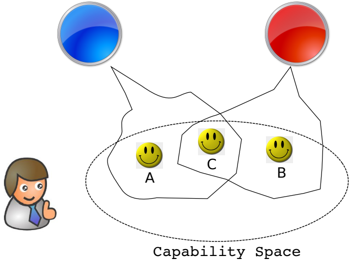
$$\mathbf{Pr}[\text{Cover B}] = 1 - e^{-x_2}$$

$$\mathbf{Pr}[\text{Cover C}] = 1 - e^{-(x_1+x_2)}$$

Proof: Polynomial-time Solvability

Fraction: x_1 x_2

Probability: $1 - e^{-x_1}$ $1 - e^{-x_2}$



- Value= $\mathbf{Pr}[\text{Cover A}] + \mathbf{Pr}[\text{Cover B}] + \mathbf{Pr}[\text{Cover C}]$
- Suffices to show each term concave

In general,

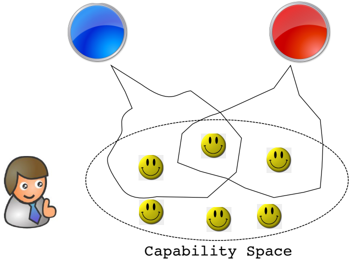
$$Pr[\text{cover D}] = 1 - \prod_{\text{j covers D}} e^{-x_j} = 1 - exp\left(- \sum_{\text{j covers D}} x_j\right)$$

which is a concave function of x.

Proof: Approximation

Fraction: y_1 y_2

Probability: $1 - e^{-y_1}$ $1 - e^{-y_2}$

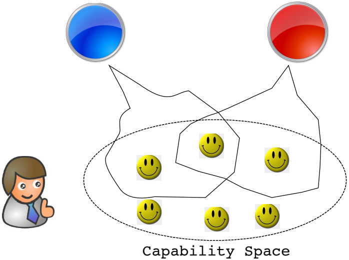


- Fix player i, and integer solution y

Proof: Approximation

Fraction: y_1 y_2

Probability: $1 - e^{-y_1}$ $1 - e^{-y_2}$

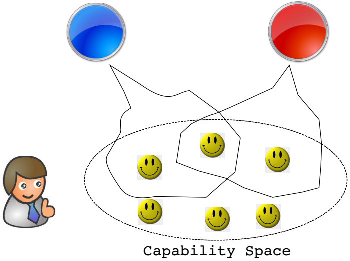


- Fix player i, and integer solution y
- Suffices to show that each capability A covered in y is covered with with probability at least $(1 - 1/e)$ in $r(y)$

Proof: Approximation

Fraction: y_1 y_2

Probability: $1 - e^{-y_1}$ $1 - e^{-y_2}$

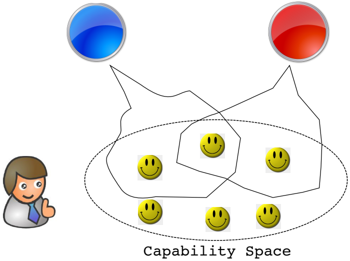


- Fix player i, and integer solution y
- Suffices to show that each capability A covered in y is covered with with probability at least $(1 - 1/e)$ in $r(y)$
- There is an item j covering A with $y_{ij} = 1$

Proof: Approximation

Fraction: y_1 y_2

Probability: $1 - e^{-y_1}$ $1 - e^{-y_2}$



- Fix player i, and integer solution y
- Suffices to show that each capability A covered in y is covered with with probability at least $(1 - 1/e)$ in $r(y)$
- There is an item j covering A with $y_{ij} = 1$
- Player i gets j with probability $1 - 1/e$ in $r(y)$

Proof Overview

Theorem (Dughmi, Roughgarden, and Yan '11)

There is a polynomial time, $1 - \frac{1}{e}$ approximate, MIDR algorithm for combinatorial auctions with coverage valuations.

Lemma (Polynomial-time solvability)

The expected welfare of rounding $x \in \mathcal{P}$ is a concave function of x.

Implies that finding the rounding-optimal fractional solution is a convex optimization problem, solvable in polynomial time*.

Lemma (Approximation)

For every set of coverage valuations and integer solution $y \in \mathcal{P}$,

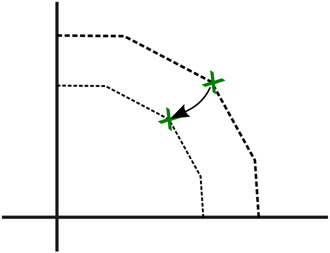
$$welfare(r(y)) \geq (1 - \frac{1}{e}) welfare(y)$$

Implies that optimizing welfare of rounded solution over $\mathcal{P}$ gives a $(1 - \frac{1}{e})$-approximation algorithm.

Relation to Lavi/Swamy

Lavi-Swamy can be interpreted as rounding anticipation for a “simple” convex rounding algorithm

- Rounding algorithm r rounds fractional point x of LP to distribution D_x with expectation $\frac{x}{\alpha}$.
- By linearity, the LP objective $v^T x$ and the welfare of the rounded solution $v^T r(x) = \frac{v^T x}{\alpha}$ are the same, up to a universal scaling factor α.
- Therefore, solving the LP optimizes over the range of distributions resulting from rounding algorithm r



Outline

1 Review

2 Rounding Anticipation

3 Characterizations of Incentive Comapatibility
- Direct Characterization
- Characterizing the Allocation rule

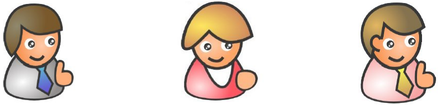
4 Lower Bounds in Prior Free AMD

Characterizing Incentive Compatible Mechanisms

- Recall: monotonicity characterization of truthful mechanisms for single parameter problems
- There are characterizations in general (non-SP) mechanism design problems
- However: more complex, and nuanced
- Nevertheless, useful for lower bounds

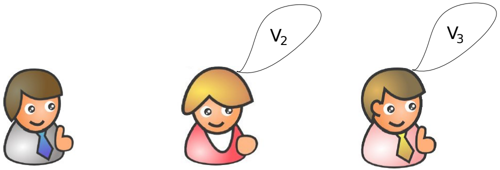
Taxation Principle

For each player i and fixed reports v_{-i} of others:



Taxation Principle

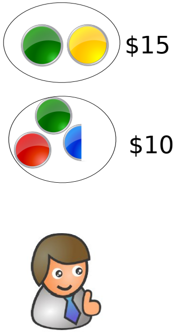
For each player i and fixed reports v_{-i} of others:



Taxation Principle

For each player i and fixed reports v_{-i} of others:

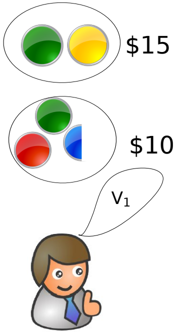
- Truthful mechanism fixes a menu of distributions over allocations, and associated prices



Taxation Principle

For each player i and fixed reports v_{-i} of others:

- Truthful mechanism fixes a menu of distributions over allocations, and associated prices
- When player i reports v_i, the mechanism:
  - Chooses the distribution/price pair (D, p) maximizing $E_{\omega \sim D}[v_i(\omega)] - p$.
  - Allocates a sample $\omega \sim D$, and charges player i p



Cycle Monotonicity

The most general characterization of dominant-strategy implementable allocation rules.

Cycle Monotonicity

An allocation rule f is cycle monotone if for every player i, every valuation profile $v_{-i} \in \mathcal{V}_{-i}$ of other players, every integer $k \geq 0$, and every sequence $v_i^1, \ldots, v_i^k \in \mathcal{V}_i$ of k valuations for player i, the following holds

$$\sum_{j=1}^{k} [v_i(\omega_j) - v_i(\omega_{j+1})] \geq 0$$

where ω_j denotes $f(v_i^j, v_{-i})$ for all $j \in \{1, \ldots, k\}$, and $\omega_{k+1} = \omega_1$.

Theorem

For every mechanism design problem, an allocation rule f is dominant-strategy implementable if and only if it is cycle monotone.

Weak Monotonicity

The special case of cycle monotonicity for cycles of length 2.

Weak Monotonicity

An allocation rule f is weakly monotone if for every player i, every valuation profile $v_{-i} \in \mathcal{V}_{-i}$ of other players, and every pair of valuations $v_i, v_i' \in \mathcal{V}_i$ of player i, the following holds

$$v_i(\omega) - v_i(\omega') \geq v_i'(\omega) - v_i'(\omega')$$

where $\omega = f(v_i, v_{-i})$ and $\omega' = f(v_i', v_{-i})$

This is necessary for all mechanism design problems. For problems with a convex domain, it is also sufficient.

Theorem [Saks,Yu]

For every mechanism design problem where each $\mathcal{V}_i \subseteq \mathbb{R}^\Omega$ is a convex set of functions, an allocation rule f is dominant-strategy implementable if and only if it is weakly monotone.

Roberts' Theorem

In the most general mechanism design problem imaginable, we can say more, at least about deterministic mechanisms.

Unrestricted Mechanism Design Problem

Each player's valuation is an arbitrary function $v_i : \Omega \to \mathbb{R}$. Formally, $\mathcal{V}_i = \mathbb{R}^\Omega$.

Roberts' Theorem

In the most general mechanism design problem imaginable, we can say more, at least about deterministic mechanisms.

Unrestricted Mechanism Design Problem

Each player's valuation is an arbitrary function $v_i : \Omega \to \mathbb{R}$. Formally, $\mathcal{V}_i = \mathbb{R}^\Omega$.

Here, cycle monotonicity and weak monotonicity are equivalent to maximization of a weighted variant of welfare

Theorem (Roberts)

For the unrestricted mechanism design problem, when $|\Omega \geq 3|$, the allocation rule of every deterministic and dominant-strategy truthful mechanism is an affine maximizer over some range $\mathcal{R} \subseteq \Omega$.

f is an affine maximizer over $\mathcal{R}$ if

$$f(v_1, \ldots, v_n) \in \underset{\omega \in \mathcal{R}}{\operatorname{argmax}} \left(\beta_\omega + \sum_i \alpha_i v_i(\omega) \right)$$

Restricted Valuations and/or Randomization

Problems we have seen are special cases of the unrestricted mechanism design problem

- Single-parameter problems: linearity in a single variable
- Combinatorial Auctions: No externality, submodularity, etc
- GAP: no externality

Restricted Valuations and/or Randomization

Problems we have seen are special cases of the unrestricted mechanism design problem

- Single-parameter problems: linearity in a single variable
- Combinatorial Auctions: No externality, submodularity, etc
- GAP: no externality

Even so, all mechanisms we have seen had allocation rules that were affine maximizers (though some randomized).

Restricted Valuations and/or Randomization

Question

Does Roberts' theorem still hold with restricted valuations? What about when randomization is allowed?

- Restricted valuations: No in general.
- Randomization: poorly understood.

Restricted Valuations and/or Randomization

Question

Does Roberts' theorem still hold with restricted valuations? What about when randomization is allowed?

- Restricted valuations: No in general.
- Randomization: poorly understood.
- Space of non-VCG-based mechanisms poorly understood. . .

Restricted Valuations and/or Randomization

Question

Does Roberts' theorem still hold with restricted valuations? What about when randomization is allowed?

- Restricted valuations: No in general.
- Randomization: poorly understood.
- Space of non-VCG-based mechanisms poorly understood...

Randomized analogue of Roberts seems to hold "in spirit" so far:

- Most mechanisms successfully employed are VCG-based (MIR, MIDR)
- Where VCG-based failed, a general LB usually followed.

Outline

1. Review
2. Rounding Anticipation
3. Characterizations of Incentive Comapatibility
   - Direct Characterization
   - Characterizing the Allocation rule
4. Lower Bounds in Prior Free AMD

Negative Results: First approach

Characterize/Embed Approach

1. Show Roberts-like characterization
   - Every truthful mechanism essentially optimizes welfare over a range $\mathcal{R}$
2. Show that if $\mathcal{R}$ is big enough to guarantee "good" approximation, then exact optimization over $\mathcal{R}$ embeds a hard problem.
   - Direct argument: multi-unit auctions [LMN '03].
   - VC-Dimension: combinatorial public projects. [PSS '08]

Negative Results: First approach

Characterize/Embed Approach

1. Show Roberts-like characterization
   - Every truthful mechanism essentially optimizes welfare over a range $\mathcal{R}$
2. Show that if $\mathcal{R}$ is big enough to guarantee “good” approximation, then exact optimization over $\mathcal{R}$ embeds a hard problem.
   - Direct argument: multi-unit auctions [LMN ’03].
   - VC-Dimension: combinatorial public projects. [PSS ’08]

- Successfully applied only to deterministic mechanisms.
- In some cases, such as combinatorial auctions, only embed part.
  - Applies only to maximal in range mechanisms.
  - [DN ’07], [BDFKMPSSU ’10]

Negative Results: Second Approach

Direct Approach [Dobzinski '11]

Using taxation principle, shows that a "good" mechanism must solve an intractable single-agent utility maximization problem, for some fixed reports of others.

Negative Results: Second Approach

Direct Approach [Dobzinski '11]

Using taxation principle, shows that a "good" mechanism must solve an intractable single-agent utility maximization problem, for some fixed reports of others.

Applied to combinatorial auctions and public projects [D11, DV11, DV12]