Combinatorial Auctions with Restricted Complements

Shaddin Dughmi

STANFORD UNIVERSITY → Microsoft Research → USC UNIVERSITY OF SOUTHERN CALIFORNIA

Joint work with:
Ittai Abraham
Moshe Babaioff
Tim Roughgarden

April 12, 2012

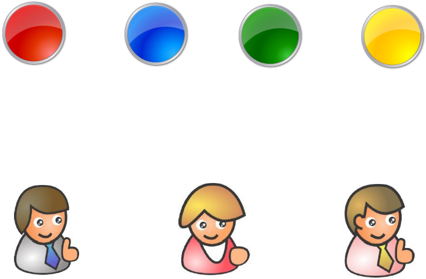
Outline

Outline

Combinatorial Auctions

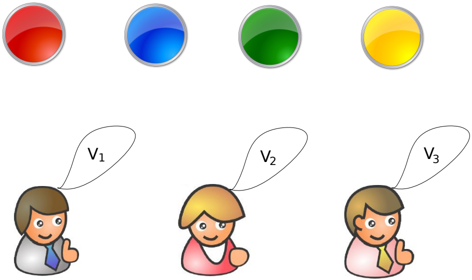
- n players

Combinatorial Auctions



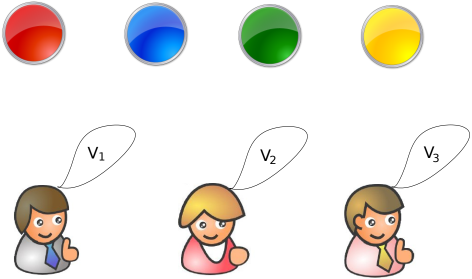
- n players
- m items

Combinatorial Auctions



- n players
- m items
- Valuation $v_i : 2^{[m]} \to \mathbb{R}$.
  - $v_i(B)$ is player i's value for bundle B.

Combinatorial Auctions

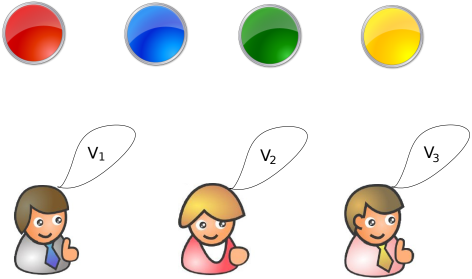


- n players
- m items
- Valuation $v_i : 2^{[m]} \to \mathbb{R}$.
  - $v_i(B)$ is player i's value for bundle B.

Goal

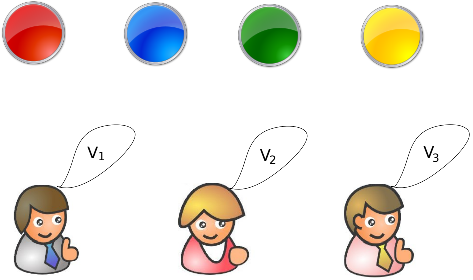
Partition items into bundles $B_1, B_2, \ldots, B_n$ to maximize welfare:
$v_1(B_1) + v_2(B_2) + \ldots + v_n(B_n)$

Combinatorial Auctions



Highly in-approximable if $P \neq NP$, unless we assume structure on valuations.

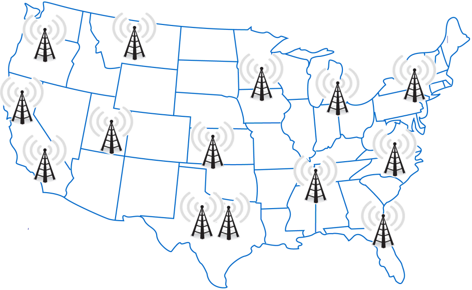
Combinatorial Auctions



Highly in-approximable if $P \neq NP$, unless we assume structure on valuations.

We will consider a classes of valuations allowing constant factor approximation algorithms.

Example: Spectrum Auctions



- Each telecom has a private value in $$ for each bundle of licenses
- Dependencies: Some of the licenses are substitutes/complements

Importance of Combinatorial Auctions

- Paradigmatic problem in algorithmic mechanism design.
- Many applications.
- Theoretically clean and expressive.

Importance of Combinatorial Auctions

- Paradigmatic problem in algorithmic mechanism design.
- Many applications.
- Theoretically clean and expressive.

Goal

Assuming valuations come from a class that naturally models some application, we want a mechanism that is:

1. (Dominant strategy) incentive compatible (truthful in expectation)
2. Polynomial time
3. Guarantees a “good” approximation to the social welfare
   - Constant factor
   - Close to best of a polynomial time approximation algorithm

Importance of Combinatorial Auctions

- Paradigmatic problem in algorithmic mechanism design.
- Many applications.
- Theoretically clean and expressive.

Goal

Assuming valuations come from a class that naturally models some application, we want a mechanism that is:

1. (Dominant strategy) incentive compatible (truthful in expectation)
2. Polynomial time
3. Guarantees a “good” approximation to the social welfare
   - Constant factor
   - Close to best of a polynomial time approximation algorithm

For which valuation classes is this possible?

Previous Work

Positive Results:

- $O(\log m \log\log m)$ for subadditive, demand oracle. [Dobzinski ’07]
- $O(\log m / \log\log m)$ for submodular, communication. [Dobzinski, Fu, Kleinberg ’10]
- $1 - 1/e$ for coverage valuations, computational. [Dughmi, Roughgarden, Yan ’11]

Negative Results:

- $\Omega(m^{\alpha})$ for submodular, value oracle. [Dughmi, Vondrak ’11]
- $\Omega(n^{\alpha})$ for submodular, computational. [Dobzinski, Vondrak ’12]

This Sucks

1. Only “good” upperbounds are for complement free valuations
   - In practice complements are present, and are the main obstacle.

This Sucks

1. Only “good” upperbounds are for complement free valuations
   - In practice complements are present, and are the main obstacle.
2. Lower bounds are fragile
   - Rely on hardness of single-player utility maximization.
   - Fall apart when we assume access to a demand oracle.

Modeling Complements

- In general, nothing is possible
  - Even for “single-minded bidders”, computational lower-bound of $\Omega(\sqrt{m})$.

Modeling Complements

- In general, nothing is possible
  - Even for “single-minded bidders”, computational lower-bound of $\Omega(\sqrt{m})$.
- Ideally, want a model that is:
  1. Parametrized by the “size” of complements
  2. Admits approximation algorithms with guarantees that degrade gracefully with size of complements.

Modeling Complements

- In general, nothing is possible
  - Even for "single-minded bidders", computational lower-bound of $\Omega(\sqrt{m})$.
- Ideally, want a model that is:
  1. Parametrized by the "size" of complements
  2. Admits approximation algorithms with guarantees that degrade gracefully with size of complements.
  3. Succinct

Modeling Complements

- In general, nothing is possible
  - Even for “single-minded bidders”, computational lower-bound of $\Omega(\sqrt{m})$.
- Ideally, want a model that is:
  1. Parametrized by the “size” of complements
  2. Admits approximation algorithms with guarantees that degrade gracefully with size of complements.
  3. Succinct
  4. Admits a polynomial-time demand oracle

Modeling Complements

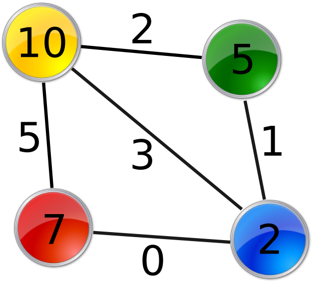
- In general, nothing is possible
  - Even for “single-minded bidders”, computational lower-bound of $\Omega(\sqrt{m})$.
- Ideally, want a model that is:
  1. Parametrized by the “size” of complements
  2. Admits approximation algorithms with guarantees that degrade gracefully with size of complements.
  3. Succinct
  4. Admits a polynomial-time demand oracle

This Paper

We consider a such a natural model for combinatorial auctions with complements.

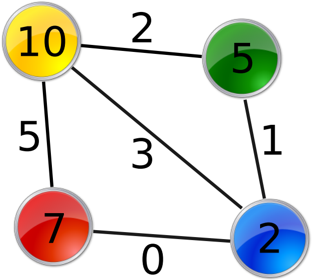
(Hyper) graph valuations

- Valuation of a player i described by a graph on the items.
- Weights $v_i(j) \geq 0$ on nodes j
- Weights $v_i(j,k) \geq 0$ on edges (j,k)
- $v_i(S) = \sum_{j \in S} v_i(j) + \sum_{j,k \in S} v_i(j,k)$



(Hyper) graph valuations

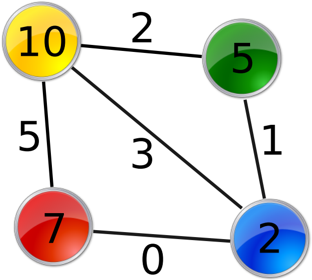
- Valuation of a player i described by a graph on the items.
- Weights $v_i(j) \geq 0$ on nodes j
- Weights $v_i(j,k) \geq 0$ on edges (j,k)
- $v_i(S) = \sum_{j \in S} v_i(j) + \sum_{j,k \in S} v_i(j,k)$



- Generalizing to hypergraphs, we model k-complements as a k-hypergraph valuation.

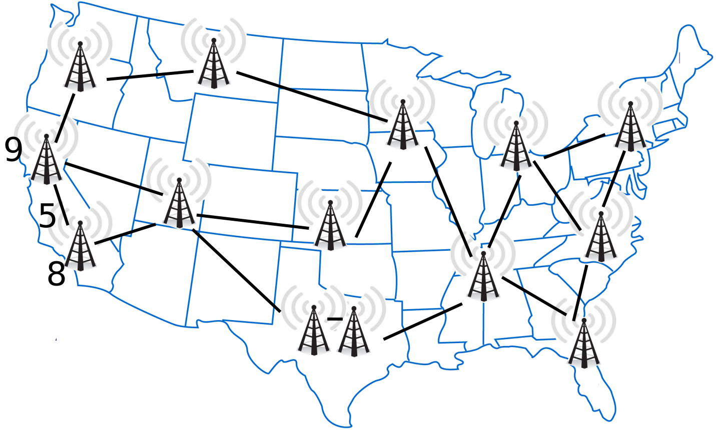
(Hyper) graph valuations

- Valuation of a player i described by a graph on the items.
- Weights $v_i(j) \geq 0$ on nodes j
- Weights $v_i(j,k) \geq 0$ on edges (j,k)
- $v_i(S) = \sum_{j \in S} v_i(j) + \sum_{j,k \in S} v_i(j,k)$



- Generalizing to hypergraphs, we model k-complements as a k-hypergraph valuation.
- Similar to models proposed earlier in the literature [Conitzer, Sandholm, and Santi '05, Chevaleyre et al '08].

Example: Spectrum Auctions



Results

1. Polynomial-time k-approximation algorithm for k-complements.
2. Polynomial-time and Truthful PTAS for 2-complements when valuation graphs exclude a fixed minor.
3. Polynomial-time and Truthful-in-expectation $O(\log^k(m))$ approximation in general.

Results

1. Polynomial-time k-approximation algorithm for k-complements.
2. Polynomial-time and Truthful PTAS for 2-complements when valuation graphs exclude a fixed minor.
3. Polynomial-time and Truthful-in-expectation $O(\log^k(m))$ approximation in general.

Results

1. Polynomial-time k-approximation algorithm for k-complements.
2. Polynomial-time and Truthful PTAS for 2-complements when valuation graphs exclude a fixed minor.
3. Polynomial-time and Truthful-in-expectation $O(\log^k(m))$ approximation in general.

Techniques: Proxy bidders approach of Dobzinski, Fu, and Kleinberg ’10, LP approach of Lavi and Swamy ’05.

Outline

Mechanisms and Truthfulness

Mechanism

1. Bidding: Solicit valuations $v_1, \ldots, v_n : 2^{[m]} \to \mathbb{R}$
2. Allocation: Compute "good" allocation $B_1, \ldots, B_n$
3. Payment: Charge payments $p_0, \ldots, p_n$

Mechanisms and Truthfulness

Mechanism

1. Bidding: Solicit valuations $v_1, \ldots, v_n : 2^{[m]} \to \mathbb{R}$
2. Allocation: Compute "good" allocation $B_1, \ldots, B_n$
3. Payment: Charge payments $p_0, \ldots, p_n$

Truthfulness in Expectation

A mechanism is truthful in expectation if a player maximizes his expected utility by reporting his true valuation, regardless of reports of others.

- $utility(i) = v_i(B_i) - p_i$

VCG Mechanism

Vickrey Clarke Groves (VCG) Mechanism for CA

1. Solicit purported valuations $v_1, \ldots, v_n : 2^{[m]} \to \mathbb{R}$
2. Find allocation $(B_1^*, \ldots, B_n^*)$ maximizing (purported) welfare: $\sum_i v_i(B_i^*)$
3. Charge each player his externality
   - The increase in (purported) welfare of other players if he drops out

VCG Mechanism

Vickrey Clarke Groves (VCG) Mechanism for CA

1. Solicit purported valuations $v_1, \ldots, v_n : 2^{[m]} \to \mathbb{R}$
2. Find allocation $(B_1^*, \ldots, B_n^*)$ maximizing (purported) welfare: $\sum_i v_i(B_i^*)$
3. Charge each player his externality
   - The increase in (purported) welfare of other players if he drops out

Theorem (Vickrey, Clarke, Groves)

VCG is truthful

VCG Mechanism

Vickrey Clarke Groves (VCG) Mechanism for CA

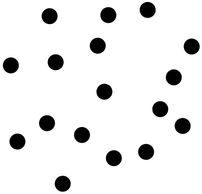
1. Solicit purported valuations $v_1, \ldots, v_n : 2^{[m]} \to \mathbb{R}$
2. Find allocation $(B_1^*, \ldots, B_n^*)$ maximizing (purported) welfare: $\sum_i v_i(B_i^*)$
3. Charge each player his externality
   - The increase in (purported) welfare of other players if he drops out

Problem

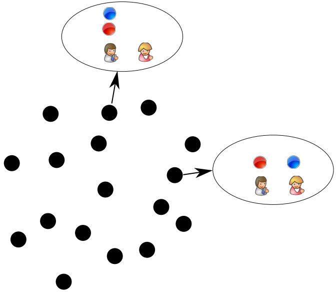
When the allocation problem is NP-hard, VCG cannot be implemented in polynomial time.

Some "special" approximation algorithms, when plugged into VCG, preserve truthfulness and recover polytime.

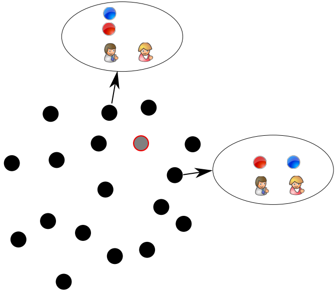
Maximal in Distributional Range Algorithms



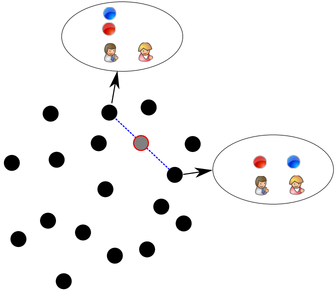
Maximal in Distributional Range Algorithms



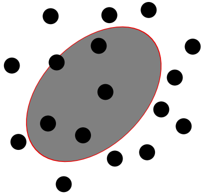
Maximal in Distributional Range Algorithms



Maximal in Distributional Range Algorithms



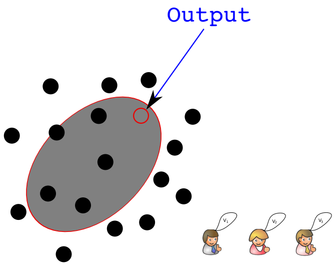
Maximal in Distributional Range Algorithms



Maximal in Distributional Range

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

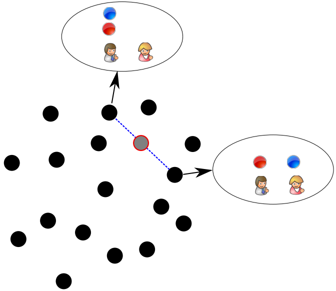
Maximal in Distributional Range Algorithms



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.

Maximal in Distributional Range Algorithms



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

Maximal in Distributional Range and Truthfulness

Plugging an MIDR algorithm into VCG yields a truthful-in-expectation mechanism

- Simply VCG applied to the “smaller” problem of finding the best lottery in $\mathcal{R}$, which we solve optimally.

Maximal in Distributional Range and Truthfulness

Plugging an MIDR algorithm into VCG yields a truthful-in-expectation mechanism

- Simply VCG applied to the "smaller" problem of finding the best lottery in $\mathcal{R}$, which we solve optimally.

Upshot

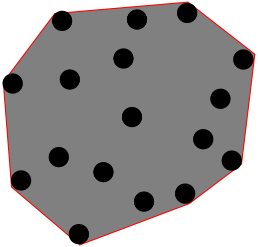
Reduced designing a truthful mechanism to designing an approximation algorithm of this MIDR variety.

Designing MIDR Algorithms

- A good MIDR Algorithm achieves a good “trade-off” between approximation ratio, and runtime

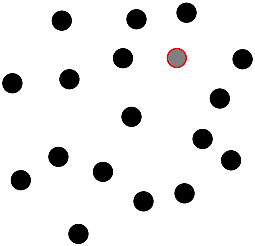
Designing MIDR Algorithms

- A good MIDR Algorithm achieves a good "trade-off" between approximation ratio, and runtime
- At one extreme: $\mathcal{R} =$ all distributions
  - Approximation ratio = 1
  - NP-hard if the problem is NP-hard



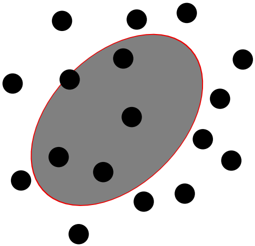
Designing MIDR Algorithms

- A good MIDR Algorithm achieves a good “trade-off” between approximation ratio, and runtime
- At one extreme: $\mathcal{R} =$ all distributions
  - Approximation ratio = 1
  - NP-hard if the problem is NP-hard
- At another extreme: $\mathcal{R} = \{x\}$ a singleton
  - Definitely polytime
  - Approximation ratio is terrible



Designing MIDR Algorithms

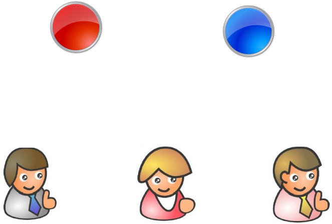
- A good MIDR Algorithm achieves a good "trade-off" between approximation ratio, and runtime
- At one extreme: $\mathcal{R} =$ all distributions
  - Approximation ratio = 1
  - NP-hard if the problem is NP-hard
- At another extreme: $\mathcal{R} = \{x\}$ a singleton
  - Definitely polytime
  - Approximation ratio is terrible



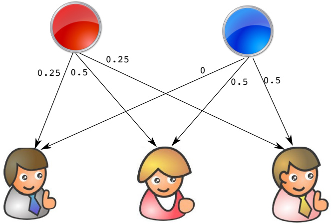
Is there a "sweet spot"?

- Large enough for good approximation
- Small/well-structured enough for polytime optimization

Example of MIDR

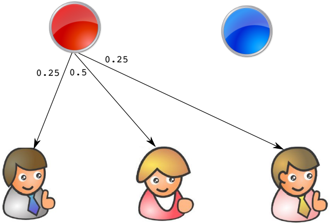


Example of MIDR



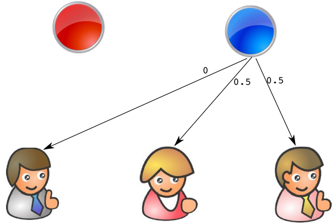
- Independent lottery:
  - Associates with each player i and item j probability x_{ij} that i gets j
  - Each item j assigned independently with those probabilities.

Example of MIDR



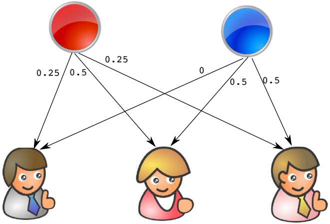
- Independent lottery:
  - Associates with each player i and item j probability x_{ij} that i gets j
  - Each item j assigned independently with those probabilities.

Example of MIDR



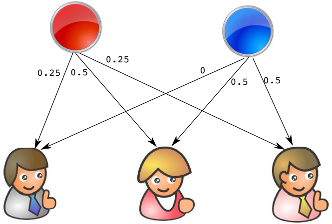
- Independent lottery:
  - Associates with each player i and item j probability x_{ij} that i gets j
  - Each item j assigned independently with those probabilities.

Example of MIDR



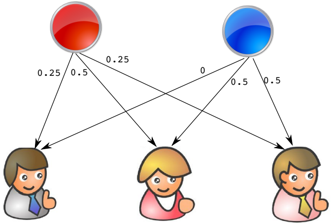
- Independent lottery:
  - Associates with each player i and item j probability x_{ij} that i gets j
  - Each item j assigned independently with those probabilities.
- Each set of fractions x_{ij} defines a different independent lottery
- The set of independent lotteries is a distributional range.

Example of MIDR



- Independent lottery:
  - Associates with each player i and item j probability x_{ij} that i gets j
  - Each item j assigned independently with those probabilities.
- Each set of fractions x_{ij} defines a different independent lottery
- The set of independent lotteries is a distributional range.
- Fact: MIDR over all independent lotteries is NP-hard.

Example of MIDR

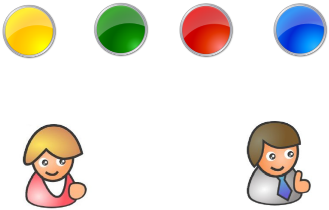


- Independent lottery:
  - Associates with each player i and item j probability x_{ij} that i gets j
  - Each item j assigned independently with those probabilities.
- Each set of fractions x_{ij} defines a different independent lottery
- The set of independent lotteries is a distributional range.
- Fact: MIDR over all independent lotteries is NP-hard.

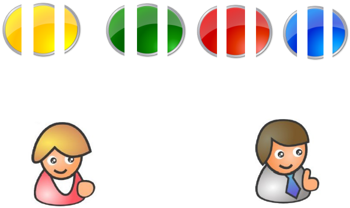
Ours will be the range of independent lotteries where each $x_{ij} \in \{0, 1/\log m\}$.

Outline

The Distributional Range



The Distributional Range



- Divide each item into $\log m$ raffle tickets
  - Each ticket is a $1/\log m$ chance of winning the item.

The Distributional Range

- Divide each item into $\log m$ raffle tickets
  - Each ticket is a $1/\log m$ chance of winning the item.
- Consider allocations of raffle tickets s.t. no player getting more than one ticket per item.

The Distributional Range

- Divide each item into $\log m$ raffle tickets
  - Each ticket is a $1/\log m$ chance of winning the item.
- Consider allocations of raffle tickets s.t. no player getting more than one ticket per item.
- Each such allocation of raffle tickets maps to a random allocation of original items
  - Independently for each item, randomly choose one of the raffle tickets as the winner.

The Distributional Range

- Divide each item into $\log m$ raffle tickets
  - Each ticket is a $1/\log m$ chance of winning the item.
- Consider allocations of raffle tickets s.t. no player getting more than one ticket per item.
- Each such allocation of raffle tickets maps to a random allocation of original items
  - Independently for each item, randomly choose one of the raffle tickets as the winner.
- The family of resulting random allocations is our range $\mathcal{R}$.

The Approximation Guarantee

Lemma

For every allocation $(S_1, \ldots, S_n)$ *of items, there is a lottery in the range with* $1/\log^2 m$ *fraction of the welfare.*

The Approximation Guarantee

Lemma

For every allocation $(S_1, \ldots, S_n)$ of items, there is a lottery in the range with $1/\log^2 m$ fraction of the welfare.

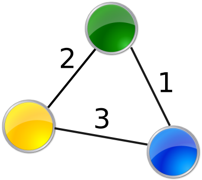
Since we plan to optimize over the range, we will get a $\log^2 m$ approximation.

The Approximation Guarantee

Proof.

- Replace each item in each S_i with a raffle ticket for that item.
  - Corresponds to a lottery in the range, by running raffles.

□

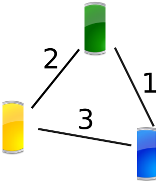


The Approximation Guarantee

Proof.

- Replace each item in each S_i with a raffle ticket for that item.
  - Corresponds to a lottery in the range, by running raffles.

□

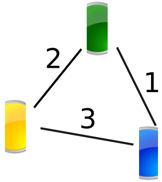


The Approximation Guarantee

Proof.

- Replace each item in each S_i with a raffle ticket for that item.
  - Corresponds to a lottery in the range, by running raffles.
- Consider an edge (j, k) with both endpoints in S_i.

□

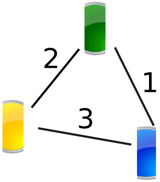


The Approximation Guarantee

Proof.

- Replace each item in each S_i with a raffle ticket for that item.
  - Corresponds to a lottery in the range, by running raffles.
- Consider an edge (j, k) with both endpoints in S_i.
- He gets each of j,k with probability $1/\log m$ in the raffle.
  - Gets both with probability $1/\log^2 m$.

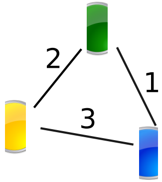
□



Proof.

- Replace each item in each S_i with a raffle ticket for that item.
  - Corresponds to a lottery in the range, by running raffles.
- Consider an edge (j,k) with both endpoints in S_i.
- He gets each of j,k with probability $1/\log m$ in the raffle.
  - Gets both with probability $1/\log^2 m$.
- Since his utility is additive over edges, done.

□



Optimizing over the Distributional Range

Observation

- Value of player i for raffle tickets for S_i is simply:

$$v_i'(S_i) = v_i(S_i)/\log^2 m$$

- Therefore: Graph valuation with weights scaled down by $\log^2 m$.
- Optimization problem for our range is combinatorial auctions with graph valuations and $\log m$ supply of each item.

Optimizing over the Distributional Range

Observation

- Value of player i for raffle tickets for S_i is simply:

$$v_i'(S_i) = v_i(S_i)/\log^2 m$$

- Therefore: Graph valuation with weights scaled down by $\log^2 m$.
- Optimization problem for our range is combinatorial auctions with graph valuations and $\log m$ supply of each item.

Fact (Folklore)

A large linear programming relaxation, called the configuration LP, is solvable in polytime when valuations admit a demand oracle.

Optimizing over the Distributional Range

Observation

- Value of player i for raffle tickets for S_i is simply:

$$v'_i(S_i) = v_i(S_i)/\log^2 m$$

- Therefore: Graph valuation with weights scaled down by $\log^2 m$.
- Optimization problem for our range is combinatorial auctions with graph valuations and $\log m$ supply of each item.

Fact (Folklore)

A large linear programming relaxation, called the configuration LP, is solvable in polytime when valuations admit a demand oracle.

Fact (Informal)

The configuration LP "essentially" has integrality gap 1 when we have $\log m$ *supply.*

Optimizing over the Distributional Range

Observation

- Value of player i for raffle tickets for S_i is simply:

$$v_i'(S_i) = v_i(S_i) / \log^2 m$$

- Therefore: Graph valuation with weights scaled down by $\log^2 m$.
- Optimization problem for our range is combinatorial auctions with graph valuations and $\log m$ supply of each item.

Fact (Folklore)

A large linear programming relaxation, called the configuration LP, is solvable in polytime when valuations admit a demand oracle.

Fact (Informal)

The configuration LP "essentially" has integrality gap 1 when we have $\log m$ *supply.*

Optimizing over the Distributional Range

Lemma

(Hyper)graph valuations admit a polynomial-time demand oracle.

Optimizing over the Distributional Range

Lemma

(Hyper)graph valuations admit a polynomial-time demand oracle.

Proof.

- Recall: Given prices p_j for each $j \in [m]$, want to find set $S \subseteq [m]$ maximizing
$$v(S) - \sum_{j \in S} p_j$$

Optimizing over the Distributional Range

Lemma

(Hyper)graph valuations admit a polynomial-time demand oracle.

Proof.

- Recall: Given prices p_j for each $j \in [m]$, want to find set $S \subseteq [m]$ maximizing
$$v(S) - \sum_{j \in S} p_j$$
- Rewriting, want to maximize $\sum_{j \in S} v(j) + \sum_{j,k \in S} v(j,k) - \sum_{j \in S} p_j$.

Optimizing over the Distributional Range

Lemma

(Hyper)graph valuations admit a polynomial-time demand oracle.

Proof.

- Recall: Given prices p_j for each $j \in [m]$, want to find set $S \subseteq [m]$ maximizing
$$v(S) - \sum_{j \in S} p_j$$
- Rewriting, want to maximize $\sum_{j \in S} v(j) + \sum_{j,k \in S} v(j,k) - \sum_{j \in S} p_j$.
- Clearly supermodular, so can be maximized in polynomial time.

□

Outline

Open Questions

1. Main open question: Is there a constant factor polytime truthful mechanism for CA with restricted complements?

Open Questions

1. Main open question: Is there a constant factor polytime truthful mechanism for CA with restricted complements?
   - Existing results rule out deterministic VCG based.
   - Existing techniques, such as Lavi/Swamy, proxy valuations, and Convex Rounding, don't seem to work.

Open Questions

1. Main open question: Is there a constant factor polytime truthful mechanism for CA with restricted complements?
   - Existing results rule out deterministic VCG based.
   - Existing techniques, such as Lavi/Swamy, proxy valuations, and Convex Rounding, don't seem to work.
2. What about other classes of valuations, such as submodular, if equipped with a demand oracle?

Open Questions

1. Main open question: Is there a constant factor polytime truthful mechanism for CA with restricted complements?
   - Existing results rule out deterministic VCG based.
   - Existing techniques, such as Lavi/Swamy, proxy valuations, and Convex Rounding, don't seem to work.
2. What about other classes of valuations, such as submodular, if equipped with a demand oracle?
3. More generally: assuming individual agents can maximize their own utility, does truthfulness + polytime follow?

Open Questions

1. Main open question: Is there a constant factor polytime truthful mechanism for CA with restricted complements?
   - Existing results rule out deterministic VCG based.
   - Existing techniques, such as Lavi/Swamy, proxy valuations, and Convex Rounding, don't seem to work.
2. What about other classes of valuations, such as submodular, if equipped with a demand oracle?
3. More generally: assuming individual agents can maximize their own utility, does truthfulness + polytime follow?
4. Other ways of modeling complements?

Thank You for Listening