Fisher Markets and Nash Social Welfare

Jugal Garg

jugal@illinois.edu

based on joint work with Martin Hoefer and Kurt Mehlhorn

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Assignment of Items to Agents with Valuations

- Set G of m indivisible items
- Set A of n agents or users

- Allocation $S = (S_1, \ldots, S_n)$ of items to agents
- Each item assigned to at most one agent

Assignment of Items to Agents with Valuations

- Set G of m indivisible items
- Set A of n agents or users

- Allocation $S = (S_1, \ldots, S_n)$ of items to agents
- Each item assigned to at most one agent

- Agent i has valuation function $v_i : 2^G \rightarrow \mathbb{R}_{\geq 0}$
- Non-negative: $v_i(S) \geq 0$ for every $S \subseteq G$
- Non-decreasing: $v_i(S) \leq v_i(T)$ for $S \subseteq T$
- Normalized: $v_i(\emptyset) = 0$

Objectives

Maximize the arithmetic mean of valuations
Utilitarian Social Welfare:

$$\mathsf{SW}(S) = \frac{1}{n}\sum_{i \in A} v_i(S_i)$$

Objectives

Maximize the arithmetic mean of valuations
Utilitarian Social Welfare:

$$\mathsf{SW}(S) = \frac{1}{n} \sum_{i \in A} v_i(S_i)$$

Maximize the minimum of valuations
Max-Min-Fairness, Egalitarian Welfare:

$$\mathsf{EW}(S) = \min_{i \in A} v_i(S_i)$$

Objectives

Maximize the arithmetic mean of valuations
Utilitarian Social Welfare:

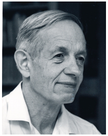
$$\text{SW}(S) = \frac{1}{n} \sum_{i \in A} v_i(S_i)$$

Maximize the minimum of valuations
Max-Min-Fairness, Egalitarian Welfare:

$$\text{EW}(S) = \min_{i \in A} v_i(S_i)$$

Maximize the geometric mean of valuations
Proportional Fairness, Nash Social Welfare:

$$\text{NSW}(S) = \left(\prod_{i \in A} v_i(S_i) \right)^{1/n}$$



Algorithms for Approximating Nash Social Welfare

Algorithm ALG computes a ρ-approximation if for every problem instance I

$$\mathsf{NSW}(ALG(I)) \geq \frac{\mathsf{NSW}(S^*)}{\rho} \ .$$

General Valuations:

- In general, if there is a finite ρ for arbitrary non-negative, non-decreasing functions, then P = NP. [Nguyen, Nguyen, Roos, Rothe, JAAMAS'14]

Algorithms for Approximating Nash Social Welfare

Additive Valuations:

$$v_i(S_i) = \sum_{j \in S_i} v_{ij}$$

- APX-hard, no 1.00008-approximation unless P $=$ NP [Lee, IPL'17]
- 2.889-approximation via markets [Cole, Gkatzelis, STOC'15]
- e-approximation via stable polynomials [Anari, Gharan, Singh, Saberi, ITCS'17]
- 2-approximation via markets [Cole, Devanur, Gkatzelis, Jain, Mai, Vazirani, Yazdanbod, EC'17]
- 1.45-approximation via limited envy [Barman, Krishnamurthy, Vaish, 2017]

Algorithms for Approximating Nash Social Welfare

Additive Valuations:

$$v_i(S_i) = \sum_{j \in S_i} v_{ij}$$

- APX-hard, no 1.00008-approximation unless P $=$ NP [Lee, IPL'17]
- 2.889-approximation via markets [Cole, Gkatzelis, STOC'15]
- e-approximation via stable polynomials [Anari, Gharan, Singh, Saberi, ITCS'17]
- 2-approximation via markets [Cole, Devanur, Gkatzelis, Jain, Mai, Vazirani, Yazdanbod, EC'17]
- 1.45-approximation via limited envy [Barman, Krishnamurthy, Vaish, 2017]

Extensions of the 2-approximation algorithm:

- Additive-separable concave valuations [Anari, Mai, Oveis Gharan, Vazirani, SODA'18]
- Multiple copies of each item [Bei, G., Hoefer, Mehlhorn, SAGT'17]

Budget-Additive Valuations

Budget-Additive Valuations:

$$v_i(S_i) = \min\left(c_i, \sum_{j \in S_i} v_{ij}\right)$$

Budget-Additive Valuations

Budget-Additive Valuations:

$$v_i(S_i) = \min\left(c_i, \sum_{j \in S_i} v_{ij}\right)$$

- Simple class of non-separable, submodular valuations

Budget-Additive Valuations

Budget-Additive Valuations:

$$v_i(S_i) = \min\left(c_i, \sum_{j \in S_i} v_{ij}\right)$$

- Simple class of non-separable, submodular valuations
- Applications in online advertising [Mehta, 2012]

Budget-Additive Valuations

Budget-Additive Valuations:

$$v_i(S_i) = \min\left(c_i, \sum_{j \in S_i} v_{ij}\right)$$

- Simple class of non-separable, submodular valuations
- Applications in online advertising [Mehta, 2012]
- Social welfare approximation, Walrasian equilibrium, Online algorithms [Andelman, Mansour, SWAT'04], [Buchbinder, Jain, Naor, ESA'07], [Srinivasan, APPROX'08], [Azar, Birnbaum, Karlin, Mathieu, Thach Nguyen, ICALP'08], [Chakrabarty, Goel, SICOMP'10], [Buchfuhrer, Dughmi, Fu, Kleinberg, Mossel, Papadimitriou, Schapira, Singer, Umans. SODA'10], [Feldman, Gravin, Lucier, STOC'13], [Roughgarden, Talgam-Cohen EC'15], etc.

Budget-Additive Valuations

Budget-Additive Valuations:

$$v_i(S_i) = \min\left(c_i, \sum_{j \in S_i} v_{ij}\right)$$

- Simple class of non-separable, submodular valuations
- Applications in online advertising [Mehta, 2012]
- Social welfare approximation, Walrasian equilibrium, Online algorithms [Andelman, Mansour, SWAT'04], [Buchbinder, Jain, Naor, ESA'07], [Srinivasan, APPROX'08], [Azar, Birnbaum, Karlin, Mathieu, Thach Nguyen, ICALP'08], [Chakrabarty, Goel, SICOMP'10], [Buchfuhrer, Dughmi, Fu, Kleinberg, Mossel, Papadimitriou, Schapira, Singer, Umans. SODA'10], [Feldman, Gravin, Lucier, STOC'13], [Roughgarden, Talgam-Cohen EC'15], etc.

Theorem [G., Hoefer, Mehlhorn, SODA'18]

For every constant $\varepsilon > 0$ there is a polynomial-time algorithm to compute a $(2e^{1/2e} + \varepsilon)$-approximation for maximum NSW with budget-additive valuations.

Budget-Additive Valuations

Budget-Additive Valuations:

$$v_i(S_i) = \min\left(c_i, \sum_{j \in S_i} v_{ij}\right)$$

- Simple class of non-separable, submodular valuations
- Applications in online advertising [Mehta, 2012]
- Social welfare approximation, Walrasian equilibrium, Online algorithms [Andelman, Mansour, SWAT'04], [Buchbinder, Jain, Naor, ESA'07], [Srinivasan, APPROX'08], [Azar, Birnbaum, Karlin, Mathieu, Thach Nguyen, ICALP'08], [Chakrabarty, Goel, SICOMP'10], [Buchfuhrer, Dughmi, Fu, Kleinberg, Mossel, Papadimitriou, Schapira, Singer, Umans. SODA'10], [Feldman, Gravin, Lucier, STOC'13], [Roughgarden, Talgam-Cohen EC'15], etc.

Theorem [G., Hoefer, Mehlhorn, SODA'18]

For every constant $\varepsilon > 0$ there is a polynomial-time algorithm to compute a $(2.404 + \varepsilon)$-approximation for maximum NSW with budget-additive valuations.

Relaxation – First Attempt

Optimization Problem as (Non-Linear) Integer Program:

$$\begin{array}{lrcll} \text{Max.} & \multicolumn{4}{l}{\left(\prod_{i \in A} \min\left(c_i, \sum_{j \in G} u_{ij} x_{ij}\right)\right)^{1/n}} \\ \text{s.t.} & \sum_{i \in A} x_{ij} & \leq & 1 & j \in G \\ & x_{ij} & \in & \{0,1\} & i \in A,\ j \in G \end{array}$$

Relaxation – First Attempt

Optimization Problem as (Non-Linear) Integer Program:

$$\begin{array}{llll} \text{Max.} & \frac{1}{n}\sum_{i \in A} \log\left(\min\left(c_i, \sum_{j \in G} u_{ij} x_{ij}\right)\right) & & \\ \text{s.t.} & \sum_{i \in A} x_{ij} \leq 1 & & j \in G \\ & x_{ij} \in \{0,1\} & & i \in A,\ j \in G \end{array}$$

Relaxation – First Attempt

Relaxation to Eisenberg-Gale Convex Program: [Eisenberg, Gale, Ann Math Stat'59] [Gale 1960], [Eisenberg, Mgmt Sci'61]

$$\text{Max.} \quad \frac{1}{n} \sum_{i \in A} \log \left(\min \left(c_i, \sum_{j \in G} u_{ij} x_{ij} \right) \right)$$

$$\text{s.t.} \quad \sum_{i \in A} x_{ij} \leq 1 \quad j \in G$$

$$x_{ij} \geq 0 \quad i \in A,\ j \in G$$

Relaxation – First Attempt

Relaxation to Eisenberg-Gale-Type Convex Program:

$$
\begin{aligned}
\text{Max.} \quad & \frac{1}{n} \sum_{i \in A} \log \sum_{j \in G} u_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j \in G} u_{ij} x_{ij} \quad \leq \quad c_i \quad i \in A \\
& \sum_{i \in A} x_{ij} \quad \leq \quad 1 \quad j \in G \\
& x_{ij} \quad \geq \quad 0 \quad i \in A,\ j \in G
\end{aligned}
$$

Optimal solutions: Competitive Equilibria with Equal Incomes

Interpretation as Competitive Equilibrium

Fisher Market with Equal Incomes:

- m divisible goods, n buyers
- Each good comes in unit supply
- Each buyer has same budget of money $m_i =$ ₹1

Interpretation as Competitive Equilibrium

Fisher Market with Equal Incomes:

- m divisible goods, n buyers
- Each good comes in unit supply
- Each buyer has same budget of money $m_i =$ ₹1
- Allocation $\mathbf{x} = (x_{ij})_{i \in A, j \in G}$
- Utility of buyer i is $u_i(\mathbf{x}) = \min\left(c_i, \sum_{j \in G} u_{ij} x_{ij}\right)$

Interpretation as Competitive Equilibrium

Fisher Market with Equal Incomes:

- m divisible goods, n buyers
- Each good comes in unit supply
- Each buyer has same budget of money $m_i =$ ₹1
- Allocation $\mathbf{x} = (x_{ij})_{i \in A, j \in G}$
- Utility of buyer i is $u_i(\mathbf{x}) = \min\left(c_i, \sum_{j \in G} u_{ij} x_{ij}\right)$

Demand bundle:

- For a set of prices $\mathbf{p}$, a demand bundle costs ₹1 and has best utility for i

Interpretation as Competitive Equilibrium

Fisher Market with Equal Incomes:

- m divisible goods, n buyers
- Each good comes in unit supply
- Each buyer has same budget of money $m_i = ₹1$
- Allocation $\mathbf{x} = (x_{ij})_{i \in A, j \in G}$
- Utility of buyer i is $u_i(\mathbf{x}) = \min\left(c_i, \sum_{j \in G} u_{ij} x_{ij}\right)$

Demand bundle:

- For a set of prices $\mathbf{p}$, a demand bundle costs ₹1 and has best utility for i

Competitive Equilibrium with Equal Incomes (CEEI): [Varian, JET'74]

- Pair $(\mathbf{x}, \mathbf{p})$ of allocation and prices
- $\mathbf{x}_i$ is demand bundle for i under $\mathbf{p}$, for every agent $i \in A$
- Market clears, i.e., total demand equals supply for every good $j \in G$

Example: Equilibrium for Additive Utilities

Markets with additive utilities:

$$u_i(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$$

Demand bundle has only goods with maximum bang-per-buck (MBB):

$$\alpha_i = \max_j u_{ij}/p_j$$

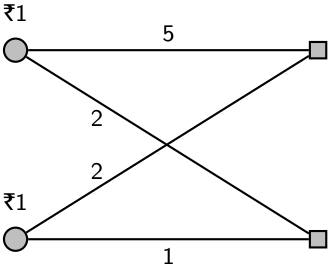
Example: Equilibrium for Additive Utilities

Markets with additive utilities:

$$u_i(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$$

Demand bundle has only goods with maximum bang-per-buck (MBB):

$$\alpha_i = \max_j u_{ij}/p_j$$



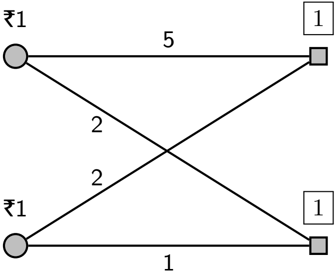
Example: Equilibrium for Additive Utilities

Markets with additive utilities:

$$u_i(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$$

Demand bundle has only goods with maximum bang-per-buck (MBB):

$$\alpha_i = \max_j u_{ij}/p_j$$



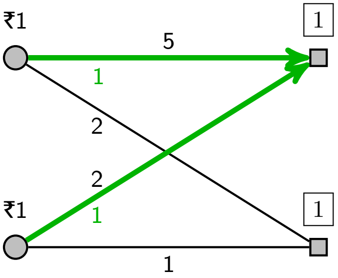
Example: Equilibrium for Additive Utilities

Markets with additive utilities:

$$u_i(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$$

Demand bundle has only goods with maximum bang-per-buck (MBB):

$$\alpha_i = \max_j u_{ij}/p_j$$



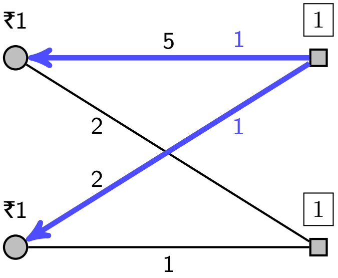
Example: Equilibrium for Additive Utilities

Markets with additive utilities:

$$u_i(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$$

Demand bundle has only goods with maximum bang-per-buck (MBB):

$$\alpha_i = \max_j u_{ij}/p_j$$



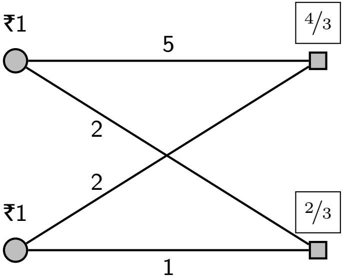
Example: Equilibrium for Additive Utilities

Markets with additive utilities:

$$u_i(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$$

Demand bundle has only goods with maximum bang-per-buck (MBB):

$$\alpha_i = \max_j u_{ij}/p_j$$



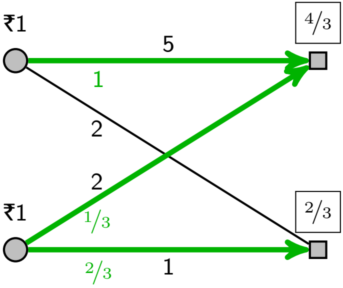
Example: Equilibrium for Additive Utilities

Markets with additive utilities:

$$u_i(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$$

Demand bundle has only goods with maximum bang-per-buck (MBB):

$$\alpha_i = \max_j u_{ij}/p_j$$



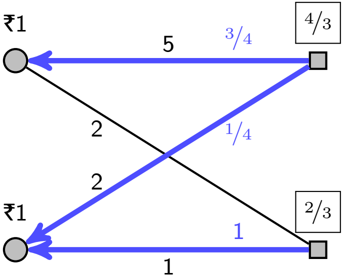
Example: Equilibrium for Additive Utilities

Markets with additive utilities:

$$u_i(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$$

Demand bundle has only goods with maximum bang-per-buck (MBB):

$$\alpha_i = \max_j u_{ij}/p_j$$



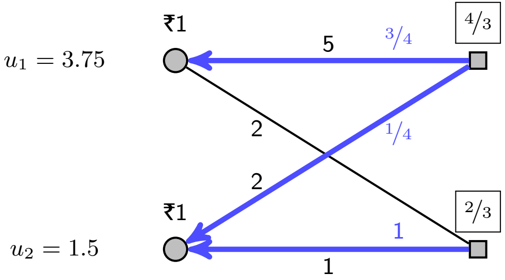
Example: Equilibrium for Additive Utilities

Markets with additive utilities:

$$u_i(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$$

Demand bundle has only goods with maximum bang-per-buck (MBB):

$$\alpha_i = \max_j u_{ij}/p_j$$

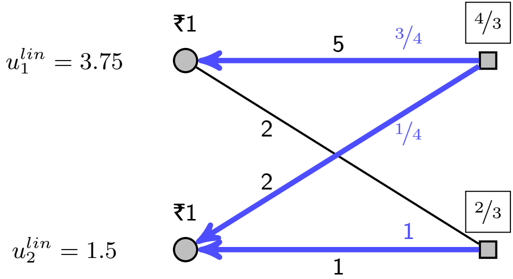


Equilibrium for Budget-Additive Utilities

- Let $(\mathbf{x}, \mathbf{p})$ be CEEI for an additive market with $u_i^{lin}(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$.
  Market clears, $\mathbf{x}_i$ is demand bundle also for u_i
- $(\mathbf{x}, \mathbf{p})$ is CEEI for the budget-additive market.
- However, suppose caps are $c_1 = 2$, $c_2 = \infty$:

Equilibrium for Budget-Additive Utilities

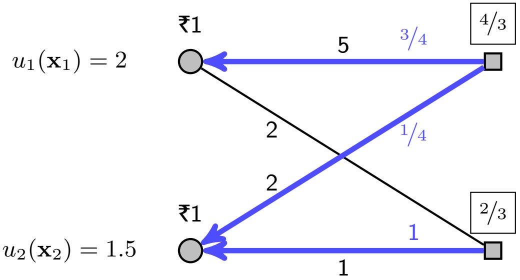
- Let $(\mathbf{x}, \mathbf{p})$ be CEEI for an additive market with $u_i^{lin}(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$. Market clears, $\mathbf{x}_i$ is demand bundle also for u_i
- $(\mathbf{x}, \mathbf{p})$ is CEEI for the budget-additive market.
- However, suppose caps are $c_1 = 2$, $c_2 = \infty$:



Problem with this equilibrium:

Equilibrium for Budget-Additive Utilities

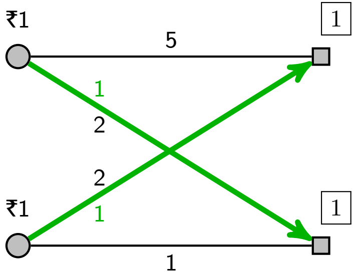
- Let $(\mathbf{x}, \mathbf{p})$ be CEEI for an additive market with $u_i^{lin}(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$. Market clears, $\mathbf{x}_i$ is demand bundle also for u_i
- $(\mathbf{x}, \mathbf{p})$ is CEEI for the budget-additive market.
- However, suppose caps are $c_1 = 2$, $c_2 = \infty$:



Problem with this equilibrium: Buyer 1 gets stuff that he does not value.

Equilibrium for Budget-Additive Utilities

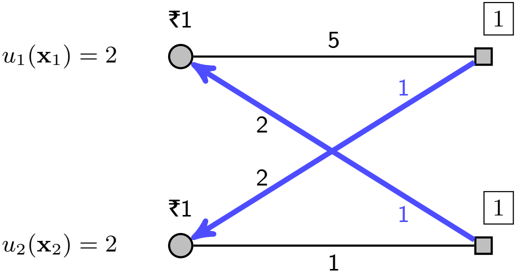
- Let $(\mathbf{x}, \mathbf{p})$ be CEEI for an additive market with $u_i^{lin}(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$. Market clears, $\mathbf{x}_i$ is demand bundle also for u_i
- $(\mathbf{x}, \mathbf{p})$ is CEEI for the budget-additive market.
- However, suppose caps are $c_1 = 2$, $c_2 = \infty$:



Problem with this equilibrium:

Equilibrium for Budget-Additive Utilities

- Let $(\mathbf{x}, \mathbf{p})$ be CEEI for an additive market with $u_i^{lin}(\mathbf{x}_i) = \sum_j u_{ij} x_{ij}$.
  Market clears, $\mathbf{x}_i$ is demand bundle also for u_i
- $(\mathbf{x}, \mathbf{p})$ is CEEI for the budget-additive market.
- However, suppose caps are $c_1 = 2$, $c_2 = \infty$:



Problem with this equilibrium: Buyer 1 can satisfy demand with less money.

Thrifty and Modest

Assumption

Each buyer spends least amount of money to reach optimal utility

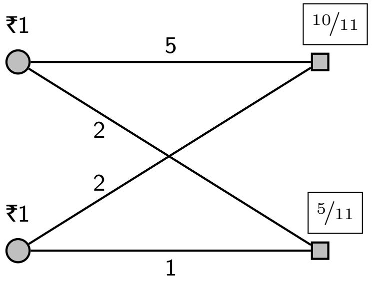
- $\mathbf{x}_i$ is modest: $\sum_j u_{ij} x_{ij} \leq c_i$.
- $\mathbf{x}_i$ is thrifty or MBB: $x_{ij} > 0 \Rightarrow u_{ij}/p_j = \alpha_i$.
- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$

Thrifty and Modest

Assumption

Each buyer spends least amount of money to reach optimal utility

- $\mathbf{x}_i$ is modest: $\sum_j u_{ij} x_{ij} \leq c_i$.
- $\mathbf{x}_i$ is thrifty or MBB: $x_{ij} > 0 \Rightarrow u_{ij}/p_j = \alpha_i$.
- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$

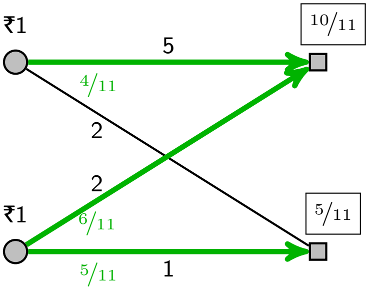


Thrifty and Modest

Assumption

Each buyer spends least amount of money to reach optimal utility

- $\mathbf{x}_i$ is modest: $\sum_j u_{ij} x_{ij} \leq c_i$.
- $\mathbf{x}_i$ is thrifty or MBB: $x_{ij} > 0 \Rightarrow u_{ij}/p_j = \alpha_i$.
- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$

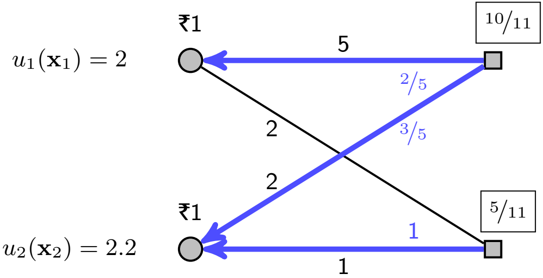


Thrifty and Modest

Assumption

Each buyer spends least amount of money to reach optimal utility

- $\mathbf{x}_i$ is modest: $\sum_j u_{ij} x_{ij} \leq c_i$.
- $\mathbf{x}_i$ is thrifty or MBB: $x_{ij} > 0 \Rightarrow u_{ij}/p_j = \alpha_i$.
- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$



Fisher Markets with Budget-Additive Utilities

Structure:

- $(\mathbf{x}, \mathbf{p})$ thrifty & modest CEEI $\Leftrightarrow$
  $\mathbf{x}$ optimal solution to EG-type convex program

  [Bei, G., Hoefer, Mehlhorn, ESA'16]

  [Cole, Devanur, Gkatzelis, Jain, Mai, Vazirani, Yazdanbod, EC'17]
- Price vectors of thrifty & modest CEEI form a lattice

  [Bei, G., Hoefer, Mehlhorn, ESA'16]

Fisher Markets with Budget-Additive Utilities

Structure:

- $(\mathbf{x}, \mathbf{p})$ thrifty & modest CEEI $\Leftrightarrow$
  $\mathbf{x}$ optimal solution to EG-type convex program

  [Bei, G., Hoefer, Mehlhorn, ESA'16]

  [Cole, Devanur, Gkatzelis, Jain, Mai, Vazirani, Yazdanbod, EC'17]
- Price vectors of thrifty & modest CEEI form a lattice

  [Bei, G., Hoefer, Mehlhorn, ESA'16]

Computing Thrifty & Modest CEEI:

- Arbitrary one in weakly polynomial time [Vegh, MOR'14]
- With maximum and minimum prices [Bei, G., Hoefer, Mehlhorn ESA'16]

So Far

NSW Problem

$\downarrow$ Relaxation

EG-type Convex Program

$\parallel\!\!|$

Fisher Markets with Budget-additive Utilties

So Far

NSW Problem

$\downarrow$ Relaxation

EG-type Convex Program

$|||$

Fisher Markets with Budget-additive Utilties

However, no meaningful approximation guarantee for NSW by rounding:

EG-type convex program has exponential integrality gap!

[Cole, Gkatzelis, STOC'15]

Strategy

NSW Problem

$\downarrow$

Fisher Markets with Budget-additive Utilties and Earning Limits

Additional Constraints: Earning Limits

Agents with Utility Limits:

- Buyer i brings money and tries to reach utility limit c_i
- $\sum_j u_{ij} x_{ij} < c_i$: Spends all money on MBB goods
- $\sum_j u_{ij} x_{ij} = c_i$: Reach cap with minimum spending (on MBB goods)
- Adjusts total spending to prices

Goods with Earning Limits:

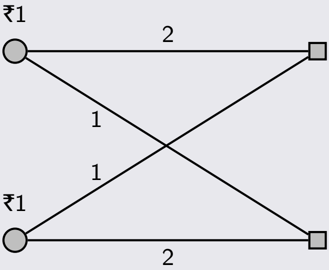
- Seller j sells good j and tries to reach earning limit d_j
- $p_j < d_j$: Sells all supply in proportion to money flow
- $p_j \geq d_j$: Reach cap with minimum supply
- Adjusts total supply to prices

Caps, Supply and Allocation

- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$
- Seller j is capped: $p_j \geq d_j$

Example

Buyer caps: $c_1 = c_2 = \infty$
Seller caps: $d_1 = d_2 = 1$

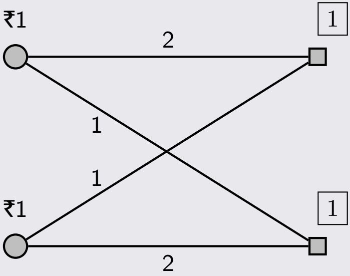


Caps, Supply and Allocation

- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$
- Seller j is capped: $p_j \geq d_j$

Example

Buyer caps: $c_1 = c_2 = \infty$
Seller caps: $d_1 = d_2 = 1$

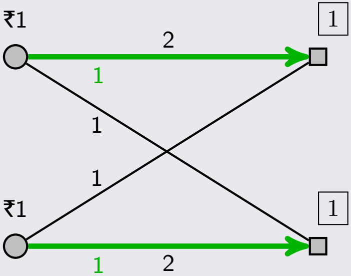


Caps, Supply and Allocation

- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$
- Seller j is capped: $p_j \geq d_j$

Example

Buyer caps: $c_1 = c_2 = \infty$
Seller caps: $d_1 = d_2 = 1$

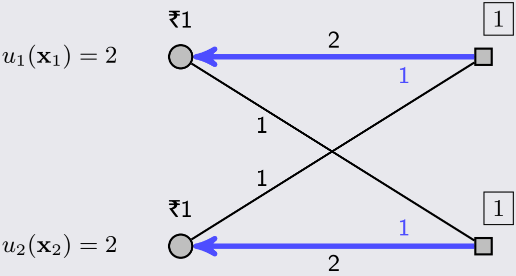


Caps, Supply and Allocation

- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$
- Seller j is capped: $p_j \geq d_j$

Example

Buyer caps: $c_1 = c_2 = \infty$
Seller caps: $d_1 = d_2 = 1$

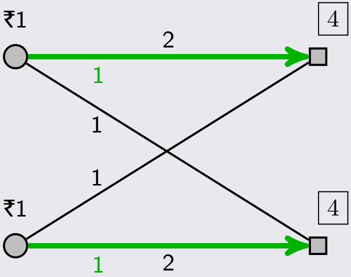


Caps, Supply and Allocation

- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$
- Seller j is capped: $p_j \geq d_j$

Example

Buyer caps: $c_1 = c_2 = \infty$
Seller caps: $d_1 = d_2 = 1$

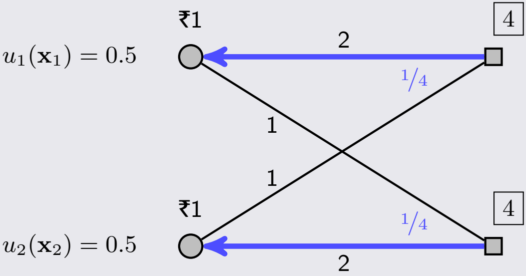


Caps, Supply and Allocation

- Buyer i is capped: $u_i(\mathbf{x}_i) = c_i$
- Seller j is capped: $p_j \geq d_j$

Example

Buyer caps: $c_1 = c_2 = \infty$
Seller caps: $d_1 = d_2 = 1$



Non-convexity

The set of equilibria is non-convex and disconnected.

There can be no convex program such that optimal solutions are the set of allocations and/or price vectors of thrifty & modest CEEI.

Fully Polynomial-Time Approximation Scheme

For fixed $\varepsilon > 0$ consider perturbed utility $\tilde{u}$:

- Get $\tilde{u}_{ij}$ by rounding up u_{ij} to next power of $1+\varepsilon$
- Perturbed utility is $\tilde{u}(\mathbf{x}) = \min\left(c_i, \sum_j \tilde{u}_{ij} x_{ij}\right)$

Fully Polynomial-Time Approximation Scheme

For fixed $\varepsilon > 0$ consider perturbed utility $\tilde{u}$:

- Get $\tilde{u}_{ij}$ by rounding up u_{ij} to next power of $1 + \varepsilon$
- Perturbed utility is $\tilde{u}(\mathbf{x}) = \min\left(c_i, \sum_j \tilde{u}_{ij} x_{ij}\right)$

Theorem [G., Hoefer, Mehlhorn, SODA'18]

There is an algorithm to compute an exact equilibrium for perturbed utilities in time polynomial in n, m, $\log \max_{i,j}(u_{ij}, c_i, d_j)$ and $\frac{1}{\varepsilon}$.

Fully Polynomial-Time Approximation Scheme

For fixed $\varepsilon > 0$ consider perturbed utility $\tilde{u}$:

- Get $\tilde{u}_{ij}$ by rounding up u_{ij} to next power of $1 + \varepsilon$
- Perturbed utility is $\tilde{u}(\mathbf{x}) = \min\left(c_i, \sum_j \tilde{u}_{ij} x_{ij}\right)$

Theorem [G., Hoefer, Mehlhorn, SODA'18]

There is an algorithm to compute an exact equilibrium for perturbed utilities in time polynomial in n, m, $\log \max_{i,j}(u_{ij}, c_i, d_j)$ and $\frac{1}{\varepsilon}$.

The set of equilibria remains non-convex and disjoint even when all u_{ij}s are integer powers of a real number.

Fully Polynomial-Time Approximation Scheme

Surplus ...

- ... of an agent:

$$s(i) = \sum_{j \in G} f_{ij} - m_i^a$$

- ... of a good:

$$s(j) = p_i^a - \sum_{i \in A} f_{ij}$$

Fully Polynomial-Time Approximation Scheme

Surplus ...

- ... of an agent:

$$s(i) = \sum_{j \in G} f_{ij} - m_i^a$$

- ... of a good:

$$s(j) = p_i^a - \sum_{i \in A} f_{ij}$$

Maintaining and adjusting a money flow f:

- Compute thrifty CEEI when ignoring utility caps

  [Bei, G., Hoefer, Mehlhorn, SAGT'17]

- Surplus of all goods 0, surplus of buyers may be positive.

Fully Polynomial-Time Approximation Scheme

Surplus ...

- ... of an agent:

$$s(i) = \sum_{j \in G} f_{ij} - m_i^a$$

- ... of a good:

$$s(j) = p_i^a - \sum_{i \in A} f_{ij}$$

Maintaining and adjusting a money flow f:

- Compute thrifty CEEI when ignoring utility caps

[Bei, G., Hoefer, Mehlhorn, SAGT'17]

- Surplus of all goods 0, surplus of buyers may be positive.

- Prices decrease monotonically during the algorithm
- Iteratively bring surplus to 0 for some buyer, while...
- ... keeping surplus of all goods 0, and
- ... keeping 0-surplus-buyers at surplus 0.

FPTAS Outline

Enlarging the set of buyers with 0 surplus

- Pick a buyer k with $s(k) > 0$

FPTAS Outline

Enlarging the set of buyers with 0 surplus

- Pick a buyer k with $s(k) > 0$
- Consider buyers and goods reachable from k wrt. residual money flow

FPTAS Outline

Enlarging the set of buyers with 0 surplus

- Pick a buyer k with $s(k) > 0$
- Consider buyers and goods reachable from k wrt. residual money flow
- Decrease prices of reachable goods by common factor until:

  Event 1: New MBB edge

  Event 2: $s(k)$ becomes zero.

FPTAS Outline

Enlarging the set of buyers with 0 surplus

- Pick a buyer k with $s(k) > 0$
- Consider buyers and goods reachable from k wrt. residual money flow
- Decrease prices of reachable goods by common factor until:

  Event 1: New MBB edge

  Event 2: $s(k)$ becomes zero.

Key Lemma: After a polynomially many iterations of this loop, $s(k) = 0$.

Nash Social Welfare

$$\text{NSW}(\mathbf{x}) = \left(\prod_{i \in A} \min \left(c_i, \sum_{j \in G} v_{ij} x_{ij} \right) \right)^{1/n}$$

Observations:

- Every integral $\mathbf{x}$ has same NSW($\mathbf{x}$) for v_{ij} and $v'_{ij} = \min(c_i, v_{ij})$
- If we scale v_{ij} and c_i by arbitrary positive number $k_i > 0$, it cancels out in the approximation ratio NSW($\mathbf{x}^*$)/NSW($\mathbf{x}^{alg}$).

⇒ Assume $v_{ij} \leq c_i \; \forall i, j$.
⇒ By scaling normalize to $\max_{j : p_j > 0} v_{ij}/p_j = 1$.

Nash Social Welfare

$$\text{NSW}(\mathbf{x}) = \left(\prod_{i \in A} \min \left(c_i, \sum_{j \in G} v_{ij} x_{ij} \right) \right)^{1/n}$$

Observations:

- Every integral $\mathbf{x}$ has same NSW($\mathbf{x}$) for v_{ij} and $v'_{ij} = \min(c_i, v_{ij})$
- If we scale v_{ij} and c_i by arbitrary positive number $k_i > 0$, it cancels out in the approximation ratio NSW($\mathbf{x}^*$)/NSW($\mathbf{x}^{alg}$).

$\Rightarrow$ Assume $v_{ij} \leq c_i \; \forall i, j$.
$\Rightarrow$ By scaling normalize to $\max\limits_{j: p_j > 0} v_{ij}/p_j = 1$.

Approximation Algorithm:

1. Compute a thrifty & modest CEEI wrt. perturbed valuations, where all agent budgets are ₹1, and all earning limits are ₹1.
2. Round the fractional allocation to an integral one.

Rounding the Equilibrium

Step 1: Allocation graph forms a forest. For each tree component, assign some agent to be the root. If good j has no child-agent, assign it to its parent agent.

Rounding the Equilibrium

Step 1: Allocation graph forms a forest. For each tree component, assign some agent to be the root. If good j has no child-agent, assign it to its parent agent.

No agent suffers from this assignment.

Rounding the Equilibrium

Step 1: Allocation graph forms a forest. For each tree component, assign some agent to be the root. If good j has no child-agent, assign it to its parent agent.

Step 2: If some good j has several child agents, keep only one that buys largest amount of j. For other child agents i delete edge (i, j) and make i root of a new tree component.

Rounding the Equilibrium

Step 1: Allocation graph forms a forest. For each tree component, assign some agent to be the root. If good j has no child-agent, assign it to its parent agent.

Step 2: If some good j has several child agents, keep only one that buys largest amount of j. For other child agents i delete edge (i, j) and make i root of a new tree component.

Agent loses allocation, then becomes a new root. Hence, only roots lose allocation. A root loses at most half of its value (by normalization of v).

Rounding the Equilibrium

Step 1: Allocation graph forms a forest. For each tree component, assign some agent to be the root. If good j has no child-agent, assign it to its parent agent.

Step 2: If some good j has several child agents, keep only one that buys largest amount of j. For other child agents i delete edge (i, j) and make i root of a new tree component.

Step 3: If for some good j and child agent i we have $p_j \leq m_i^a/2$, then assign j to its parent agent and make i the root node of a new tree component.

Rounding the Equilibrium

Step 1: Allocation graph forms a forest. For each tree component, assign some agent to be the root. If good j has no child-agent, assign it to its parent agent.

Step 2: If some good j has several child agents, keep only one that buys largest amount of j. For other child agents i delete edge (i, j) and make i root of a new tree component.

Step 3: If for some good j and child agent i we have $p_j \leq m_i^a/2$, then assign j to its parent agent and make i the root node of a new tree component.

Agent loses allocation, then becomes a new root. Hence, only roots lose allocation. A root loses at most half of its value (by normalization of v).

Rounding the Equilibrium

Structure of the forest:

- Every good has exactly one child agent.
- Every agent is root or gets at least half of value in CEEI from parent good.

Rounding the Equilibrium

Structure of the forest:

- Every good has exactly one child agent.
- Every agent is root or gets at least half of value in CEEI from parent good.

Step 4: For each tree component, do the following recursively:
Assign the root agent a child-good j that gives max-value among children goods in the fractional solution.
Except subtree rooted at j, assign each good to its child agent in the remaining tree.
Make child-agent of good j the new root node of the tree.

Rounding the Equilibrium

Structure of the forest:

- Every good has exactly one child agent.
- Every agent is root or gets at least half of value in CEEI from parent good.

Step 4: For each tree component, do the following recursively:
Assign the root agent a child-good j that gives max-value among children goods in the fractional solution.
Except subtree rooted at j, assign each good to its child agent in the remaining tree.
Make child-agent of good j the new root node of the tree.

Recursion path starting at root agent and continuing with max-value child good. Aside the recursion path, good goes to child agent, gives a 2-approximation. On the recursion path, approximation is captured by a telescopic product term.

Bounds on the Optimum

The rounding algorithm returns integral $\mathbf{x}$ with value of the allocation:

$$\mathsf{NSW}(\mathbf{x}) \quad \geq \quad \frac{1}{2e^{1/(2e)}} \cdot \prod_{i \in A_c} c_i \cdot \prod_{j : p_j > 1} p_j \ ,$$

where A_c is the set of capped agents and p_j the price of good j in the CEEI.

The optimal integral allocation $\mathbf{x}^*$ has value at most

$$\mathsf{NSW}(\mathbf{x}^*) \quad \leq \quad \prod_{i \in A_c} c_i \cdot \prod_{j : p_j > 1} p_j \ .$$

Overall ratio for FPTAS and rounding is $2e^{1/(2e)} + \varepsilon < 2.404 + \varepsilon$.

Bounds on the Optimum

The rounding algorithm returns integral $\mathbf{x}$ with value of the allocation:

$$\mathsf{NSW}(\mathbf{x}) \quad \geq \quad \frac{1}{2e^{1/(2e)}} \cdot \prod_{i \in A_c} c_i \cdot \prod_{j : p_j > 1} p_j \ ,$$

where A_c is the set of capped agents and p_j the price of good j in the CEEI.

The optimal integral allocation $\mathbf{x}^*$ has value at most

$$\mathsf{NSW}(\mathbf{x}^*) \quad \leq \quad \prod_{i \in A_c} c_i \cdot \prod_{j : p_j > 1} p_j \ .$$

Overall ratio for FPTAS and rounding is $2e^{1/(2e)} + \varepsilon < 2.404 + \varepsilon$.

Proposition [G., Hoefer, Mehlhorn, SODA'18]

It is NP-hard to approximate NSW with budget-additive valuations to within a factor of $\sqrt{8/7}$.

Bounds on the Optimum

The rounding algorithm returns integral $\mathbf{x}$ with value of the allocation:

$$\mathsf{NSW}(\mathbf{x}) \quad \geq \quad \frac{1}{2e^{1/(2e)}} \cdot \prod_{i \in A_c} c_i \cdot \prod_{j: p_j > 1} p_j \enspace,$$

where A_c is the set of capped agents and p_j the price of good j in the CEEI.

The optimal integral allocation $\mathbf{x}^*$ has value at most

$$\mathsf{NSW}(\mathbf{x}^*) \quad \leq \quad \prod_{i \in A_c} c_i \cdot \prod_{j: p_j > 1} p_j \enspace.$$

Overall ratio for FPTAS and rounding is $2e^{1/(2e)} + \varepsilon < 2.404 + \varepsilon$.

Proposition [G., Hoefer, Mehlhorn, SODA'18]

It is NP-hard to approximate NSW with budget-additive valuations to within a factor of 1.069.

Open Problems

Algorithms for Nash Social Welfare with provable performance guarantees are poorly understood!

- Submodular Valuations?
- (Fractionally) Subadditive, Complementary Valuations?
- Hardness of Approximation?
- Truthfulness vs. Approximation?

Markets with Caps

- Linear Markets with Earning and Utility Limits
- Equilibrium computation in PPAD $\cap$ PLS [G., Hoefer, Mehlhorn, 2017b]
- Poly-time for constant number of buyers or sellers [G., Hoefer, Mehlhorn, 2017b]
- Exchange Markets with Utility Limits?
- Additional Applications?
- etc.

Thank you!