Approximating Nash Social Welfare for submodular valuations

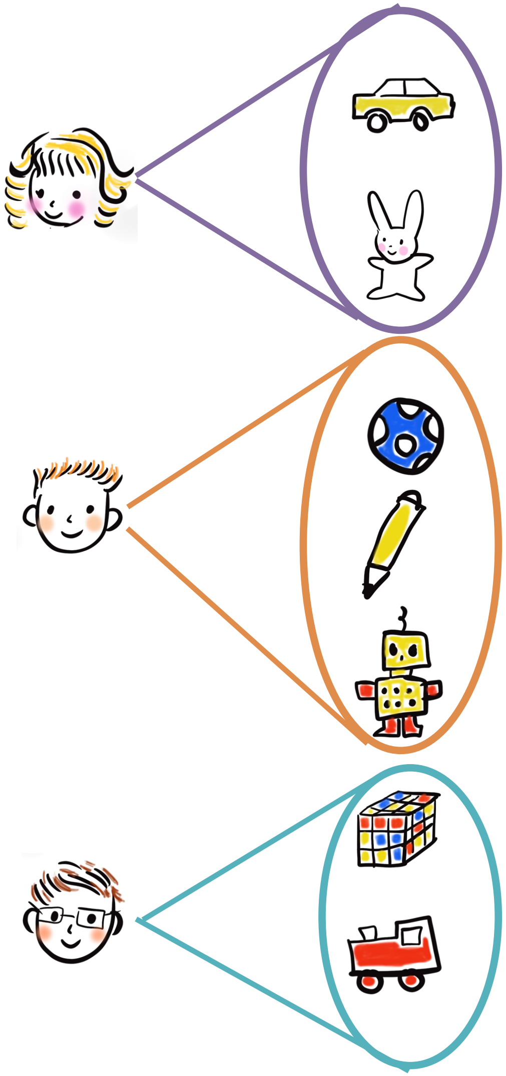
László Végh

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

Joint work with
Jugal Garg (UIUC), Edin Husić (IDSIA),
Wenzheng Li and Jan Vondrák (Stanford)

ICERM, 31 Mar 2023

Allocation problems



- A: set of n agents
- G: set of m indivisible items
- $v_i: 2^G \to \mathbb{R}_+$ valuation functions
  $v_i(\emptyset) = 0$, monotone
- Goal: find a partition $G = S_1 \cup S_2 \cup \cdots \cup S_n$ that maximizes a certain welfare function
  - Utilitarian social welfare: $\max \sum_{i=1}^{n} v_i(S_i)$
  - Max-min (Santa Claus): $\max \min_i v_i(S_i)$
  - Nash Social Welfare: $\max \left(\prod_i^n v_i(S_i)\right)^{\frac{1}{n}}$
  - ...

Nash Social Welfare

$$\max \left(\prod_{i}^{n} v_i(S_i) \right)^{\frac{1}{n}}$$

- Smooth tradeoff between efficiency and fairness
- Motivated by Nash bargaining

Nash Social Welfare

- Scale-free: invariant under replacing $v_i(S)$ by $\alpha v_i(S)$ for $\alpha > 0$
- Pareto optimal
- EF1: Envy-free up to 1 good

The Unreasonable Fairness of Maximum Nash Welfare

IOANNIS CARAGIANNIS, University of Patras
DAVID KUROKAWA, Carnegie Mellon University
HERVÉ MOULIN, University of Glasgow and Higher School of Economics, St Petersburg
ARIEL D. PROCACCIA, Carnegie Mellon University
NISARG SHAH, Carnegie Mellon University
JUNXING WANG, Carnegie Mellon University

The *maximum Nash welfare (MNW)* solution — which selects an allocation that maximizes the product of utilities — is known to provide outstanding fairness guarantees when allocating divisible goods. And while it seems to lose its luster when applied to indivisible goods, we show that, in fact, the MNW solution is unexpectedly, strikingly fair even in that setting. In particular, we prove that it selects allocations that are envy free up to one good — a compelling notion that is quite elusive when coupled with economic efficiency. We also establish that the MNW solution provides a good approximation to another popular (yet possibly infeasible) fairness property, the maximin share guarantee, in theory and — even more so — in practice. While finding the MNW solution is computationally hard, we develop a nontrivial implementation, and demonstrate that it scales well on real data. These results lead us to believe that MNW is the ultimate solution for allocating indivisible goods, and underlie its deployment on a popular fair division website.

spliddit.org

Valuation functions

- Additive

$$v_i(S) = \sum_{i \in S} v_{ij}$$

  - Restricted additive: $v_{ij} \in \{0, t_j\}$ for all $i \in A, j \in G$
- Gross substitutes/M♮-concave valuations
- Submodular:

  $v_i(S+v) - v_i(S) \geq v_i(T+v) - v_i(T) \quad \forall S \subseteq T \subseteq G$

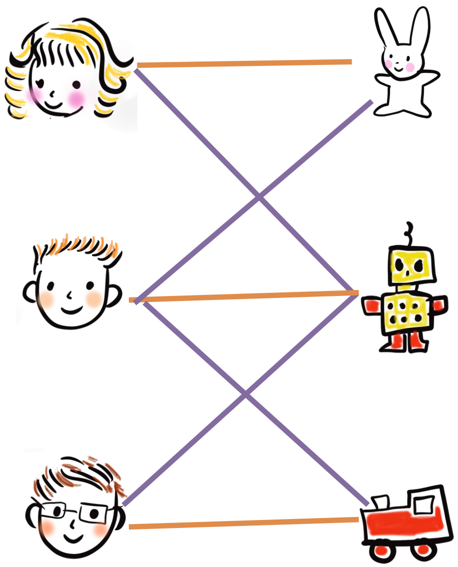
  decreasing marginal utilities
- Subadditive

$$v_i(S) + v_i(T) \geq v_i(S \cup T) \quad \forall S, T \subseteq G$$

Hardness

- Additive is in P for $\max \sum_{i=1}^{n} v_i(S_i)$
- Restricted additive is NP-hard
  for $\max \min_i v_i(S_i)$ and for $\max \left(\prod_i^n v_i(S_i)\right)^{\frac{1}{n}}$
- Reduction from SubsetSum:
  - Given $a_1, a_2, \dots, a_m \in \mathbb{N}$, decide whether there exists a partition
    $[m] = S_1 \cup S_2$ with
    $$\sum_{i \in S_1} a_i = \sum_{i \in S_2} a_j$$
- NSW is APX-hard for additive valuations [Lee 2015]
  NP-hard to approximate within a factor c for some $c > 1$

Connection to matchings



If $n = m$, optimal solution using max weight perfect matchings. $v_{ij} \coloneqq v_i(\{j\})$

- $\min_i v_i(S_i)$: find largest t such that $E_t = \{(i,j) \colon v_{ij} \geq t\}$ admits a perfect matching
- $\left(\prod_i^n v_i(S_i)\right)^{\frac{1}{n}}$: find a max weight perfect matching for

$$w_{ij} = \log v_{ij}$$

Approximation algorithms

α-approximation algorithm for fairness objective $f(S_1, S_2, \dots, S_n)$:

Find an allocation $S_1, S_2, \dots, S_n$ such that

$$f(S_1, S_2, \dots, S_n) \geq \frac{\mathrm{OPT}}{\alpha}$$

where OPT is the value of the best possible solution

	$\sum_{i=1}^{n} v_i(S_i)$	$\min_i v_i(S_i)$	$\left(\prod_i^n v_i(S_i)\right)^{\frac{1}{n}}$
Restricted additive	P	$4 + \varepsilon$ DRZ ’20	
Additive	P	$O(\sqrt{n}\log^3 n)$ AS ’10 polylog in quasipoly time CCK ’09	1.44 BKV ’18
Gross substitutes	P Gül & Stacchetti ’99		Rado: 6.8 est. LV ’21 772 GHV ’21
Submodular	$\frac{e}{e-1}$ Vondrák ’08	$O(n)$ KP ’07	380 LV ’21 4 GHLVV ’22+
Subadditive	VQ $\Omega\left(\frac{m}{\log^{1+\varepsilon} m}\right)$ MVS ’08 DQ 2 Feige ’06		VQ $\Theta(n)$ BBKS ’20, CGM ’20

	$\sum_{i=1}^{n} v_i(S_i)$	$\min_i v_i(S_i)$	$\left(\prod_i^{n} v_i(S_i)\right)^{\frac{1}{n}}$
Restricted additive	P	$4+\varepsilon$ DRZ ’20	
Additive	P	$O(\sqrt{n}\log^3 n)$ AS ’10 polylog in quasipoly time CCK ’09	1.44 BKV ’18
Gross substitutes	P Gül & Stacchetti ’99		Rado: 6.8 est. LV ’21 772 GHV ’21
Submodular	$\frac{e}{e-1}$ Vondrák ’08	$O(n)$ KP ’07	380 LV ’21 4 GHLVV ’22+
Subadditive	VQ $\Omega\left(\frac{m}{\log^{1+\varepsilon} m}\right)$ MVS ’08 DQ 2 Feige ’06		VQ $\Theta(n)$ BBKS ’20, CGM ’20

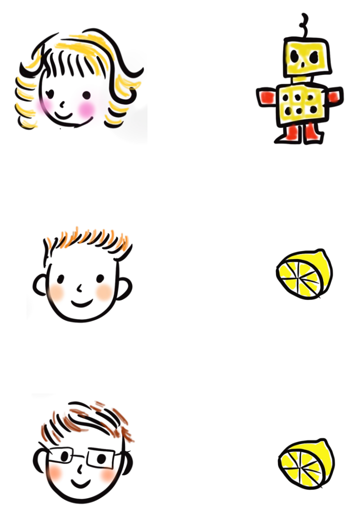
	$\sum_{i=1}^{n} v_i(S_i)$	$\min_i v_i(S_i)$	$\left(\prod_i^n v_i(S_i)\right)^{\frac{1}{n}}$
Restricted additive	P	$4+\varepsilon$ DRZ '20	
Additive	P	$O(\sqrt{n}\log^3 n)$ AS '10 polylog in quasipoly time CCK '09	1.44 BKV '18
Gross substitutes	P Gül & Stacchetti '99		Rado: 6.8 est. LV '21 772 GHV '21
Submodular	$\frac{e}{e-1}$ Vondrák '08	$O(n)$ KP '07	380 LV '21 4 GHLVV '22+
Subadditive	VQ $\Omega\left(\frac{m}{\log^{1+\varepsilon} m}\right)$ MVS '08 DQ 2 Feige '06		VQ $\Theta(n)$ BBKS '20, CGM '20

Nash Social Welfare for additive valuations

Natural convex relaxation:

$$\max \prod_i \left(\sum_j v_{ij} x_{ij} \right)^{1/n} \qquad \max \sum_i \log \sum_j v_{ij} x_{ij}$$

$$\sum_i x_{ij} \leq 1 \quad \forall j \in G\,, \qquad x \geq 0$$



- Eisenberg—Gale convex program for linear Fisher markets
- For NSW the gap is unbounded:

$$v_{ij} = \begin{cases} M \text{ if } j = 1 \\ 1 \;\text{ if } j > 1 \end{cases}$$

- Integer optimum: $M^{1/n}$
  Fractional optimum: $(M + n - 1)/n$

Nash Social Welfare for additive valuations

- Cole & Gkatzelis 2015:
  $2e^{1/e} \approx 2.89$-approximation
  spending restricted market equilibrium

- Anari, Oveis Gharan, Saberi & Singh 2017:
  e-approximation
  real stable polynomials

- Barman, Krishnamurthy, Vaish 2018:
  $e^{1/e} + \varepsilon \approx 1.44$-approximation
  connection to EF1 (envy-free up to one item) fairness

Nash Social Welfare
recent developments

Rado valuations: subclass of gross substitutes defined by matroids and matchings

- Li & Vondrák 2021: estimation algorithm for NSW with conic combination of Rado that guesses the optimum value within a factor 6.8
  real stable polynomial approach
- Garg, Husić, V. 2021: 772-approximation for NSW with Rado valuations
- Li & Vondrák 2021: 380-approximation for NSW with submodular valuations
- Garg, Husić, Li, V. & Vondrák 2022: 4-approximation for NSW with submodular valuations.

The rise and fall of techniques

- Mixed integer relaxation via concave extension of Rado GHV ‘21
- Exact convex solutions, support sparsification GHV ’21
- Multilinear relaxation, iterated continuous greedy LV ’21
- Randomized rounding LV ’21
- Multilinear relaxation, Frank-Wolfe algorithm GHLVV ’22

The rise and fall of techniques

- ~~Mixed integer relaxation via concave extension of Rado~~ GHV ‘21
- ~~Exact convex solutions, support sparsification~~ GHV ’21
- ~~Multilinear relaxation, iterated continuous greedy~~ LV ’21
- ~~Randomized rounding~~ LV ’21
- ~~Multilinear relaxation, Frank-Wolfe algorithm~~ GHLVV ’22

A simple three-phase algorithm

THM (Garg, Husić, Li, V. & Vondrák 2022): There exists a polynomial-time $4 + \varepsilon$ approximation algorithm for NSW with submodular valuations

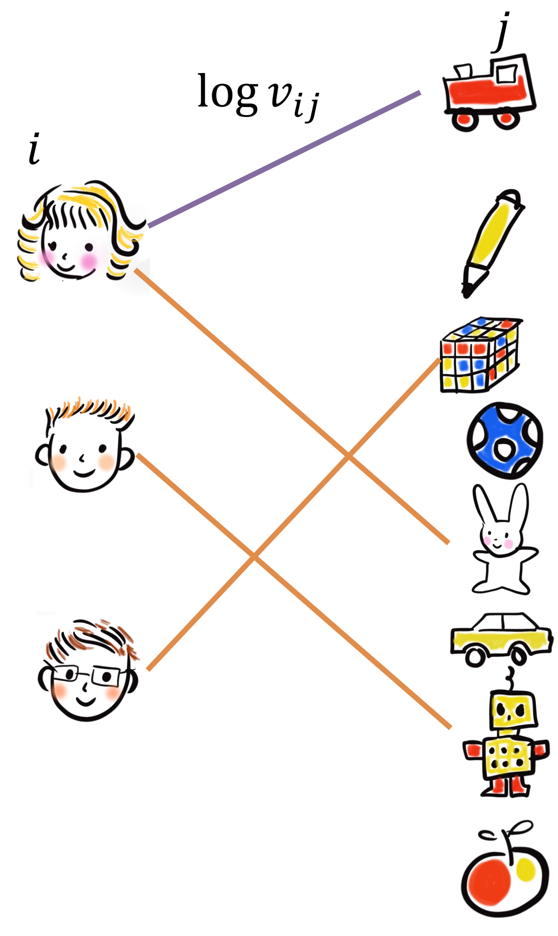
I. Solve the optimal matching problem when every agent is allowed to get exactly one good.
H: set of allocated goods.

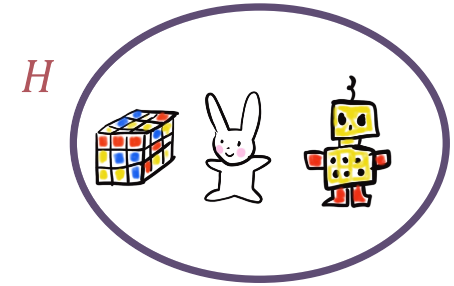
II. Allocate the remaining goods in $G \setminus H$ using local search with respect to modified valuation functions.

III. Reallocate the goods in H optimally.

Phase I: Find the set H

- $v_{ij} := v_i(\{j\})$
- Assume $v_{ij} > 0\ \forall i \forall j$
- Find a matching $\tau: A \to G$ maximizing
$$\prod_i v_{i\tau(i)}$$
- Set $H := \tau(A)$
- Warning: we cannot commit to allocating $\tau(i)$ to i in the final allocation!





Phase II: Local Search

- Favourite item in $G \setminus H$

$$\ell(i) \in \arg\max_{j \in G \setminus H} v_{ij}$$

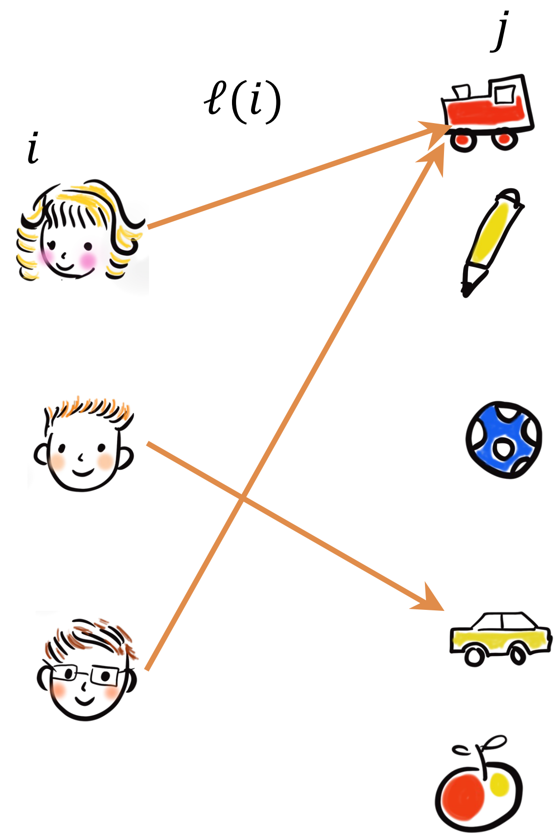
Can be the same for multiple agents!

- Modified utility function:

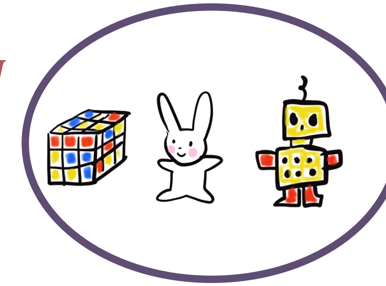
$$\bar{v}_i(S) := v_i(S) + v_{i\ell(i)}$$

Add the value of the favourite item

Properties: $\bar{v}_i(\emptyset) = v_{i\ell(i)}$ and $\bar{v}_i(j) \leq 2\bar{v}_i(\emptyset)$



H

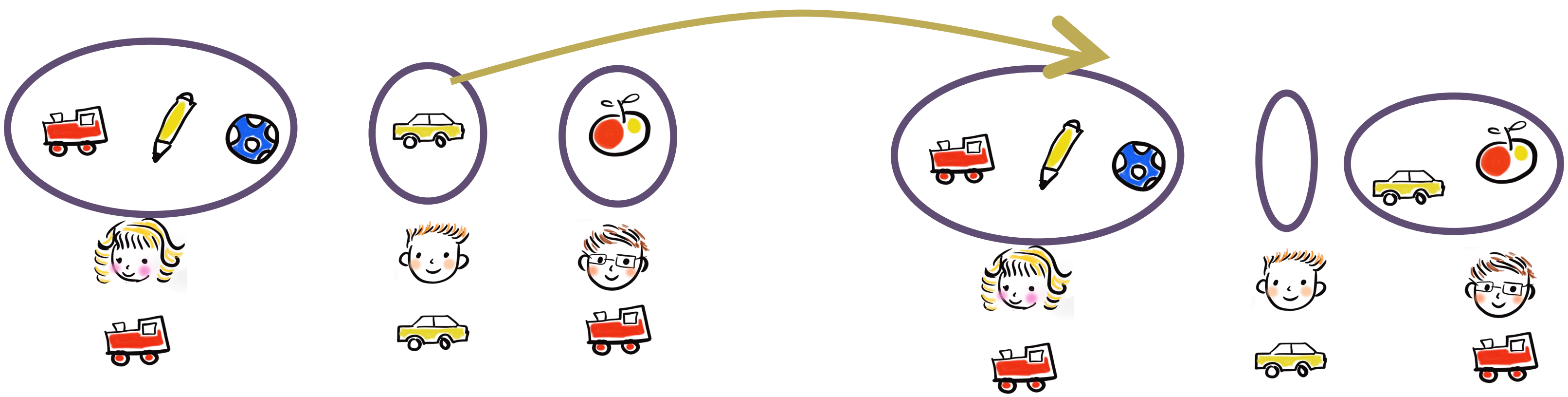


Phase II: Local Search

- Start with any allocation $(R_1, R_2, \ldots, R_n)$ of $G \setminus H$
- Modify the allocation by reassigning a single item as long as this increases the modified welfare by at least $(1+\varepsilon)$:

$$\prod_{i \in A} \bar{v}_i(R_i)$$

- Ignore items in H completely

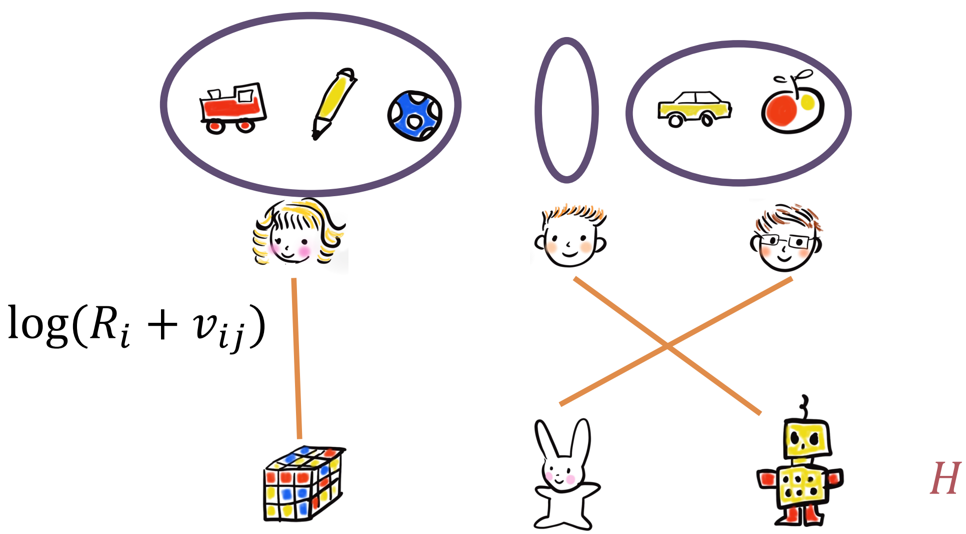


Phase III: Matching recombination

- Allocation $(R_1, R_2, \ldots, R_n)$ of $G \setminus H$ from Phase II
- We now ignore the $\ell(i)$ items!
- Find matching $\rho: A \to H$ that maximizes

$$\prod_{i \in A} v(R_i + \rho(i))$$

- Return final allocation $(R_1 + \rho(1), R_2 + \rho(2), \ldots, R_n + \rho(n))$



Analysis: local search

LEMMA I If $(R_1, R_2, \dots, R_n)$ is a local maximizer of $\prod_{i=1}^{n} \bar{v}_i(R_i)$, then

$$\prod_{i=1}^{n} \bar{v}_i(R_i) \geq \frac{1}{\mathrm{e}^n} \prod_{i=1}^{n} \bar{v}_i(S_i)$$

for any partition $(S_1, S_2, \dots, S_n)$ of $G \setminus H$

- At local optimum, for any $j \in R_i$, $k \neq i$:

$$\frac{\bar{v}_k(R_k + j)}{\bar{v}_k(R_k)} \leq \frac{\bar{v}_i(R_i)}{\bar{v}_i(R_i - j)}$$

- Define price of $j \in R_i$ as

$$p_j \coloneqq \log \frac{\bar{v}_i(R_i)}{\bar{v}_i(R_i - j)}$$

Analysis: local search

- Define price of $j \in R_i$ as

$$p_j := \log \frac{\bar{v}_i(R_i)}{\bar{v}_i(R_i - j)}$$

CLAIM: For every agent i, we have $\sum_{j \in R_i} p_j \leq 1$

PROOF

$$\sum_{j \in R_i} p_j = \sum_{j \in R_i} \log\left(1 + \frac{\bar{v}_i(R_i) - \bar{v}_i(R_i - j)}{\bar{v}_i(R_i - j)}\right)$$

$$\leq \sum_{j \in R_i} \frac{\bar{v}_i(R_i) - \bar{v}_i(R_i - j)}{\bar{v}_i(R_i - j)}$$

$$\leq \sum_{j \in R_i} \frac{v_i(R_i) - v_i(R_i - j)}{v_i(R_i)} \leq 1$$

$$\bar{v}_i(R_i - j) = v_i(R_i - j) + v_{i\ell(i)} \geq v_i(R_i - j) + v_{ij} \geq v_i(R_i)$$

Analysis: local search

LEMMA I If $(R_1, R_2, \ldots, R_n)$ is a local maximizer of $\prod_{i=1}^{n} \bar{v}_i(R_i)$, then

$$\prod_{i=1}^{n} \bar{v}_i(R_i) \geq \frac{1}{\mathrm{e}^n} \prod_{i=1}^{n} \bar{v}_i(S_i)$$

for any partition $(S_1, S_2, \ldots, S_n)$ of $G \setminus H$

Proof: For any $j \in R_i$, $k \neq i$:

$$\frac{\bar{v}_k(R_k + j)}{\bar{v}_k(R_k)} \leq \frac{\bar{v}_i(R_i)}{\bar{v}_i(R_i - j)} = \mathrm{e}^{p_j}$$

$$\prod_{i=1}^{n} \frac{\bar{v}_i(S_i)}{\bar{v}_i(R_i)} \leq \prod_{i=1}^{n} \frac{\bar{v}_i(R_i \cup S_i)}{\bar{v}_i(R_i)} \leq$$

$$\leq \prod_{i=1}^{n} \prod_{j \in S_i} \mathrm{e}^{p_j} = \prod_{i=1}^{n} \prod_{j \in R_i} \mathrm{e}^{p_j} \leq \mathrm{e}^n$$

Analysis overview

LEMMA II For the locally optimal allocation $(R_1, R_2, \ldots, R_n)$ of $G \setminus H$ from Phase II and the favourite items $\ell(i)$, there exists a matching $\pi: A \to H$ such that

$$\left(\prod_{i}^{n} v_i(R_i + \ell(i) + \pi(i))\right)^{\frac{1}{n}} \geq \frac{1}{3\mathrm{e}}\,\mathrm{OPT}$$

Warning: Allocation infeasible due to the $\ell(i)'s$

Proof sketch:

- In the optimal solution, letting every agent keep their allocation from $G \setminus H$ and the best item from H looses only a constant factor. The matching π is formed by these items in H.
- The allocation $(R_i + \ell(i))$ is approximately as good as the best possible solution on $G \setminus H$.

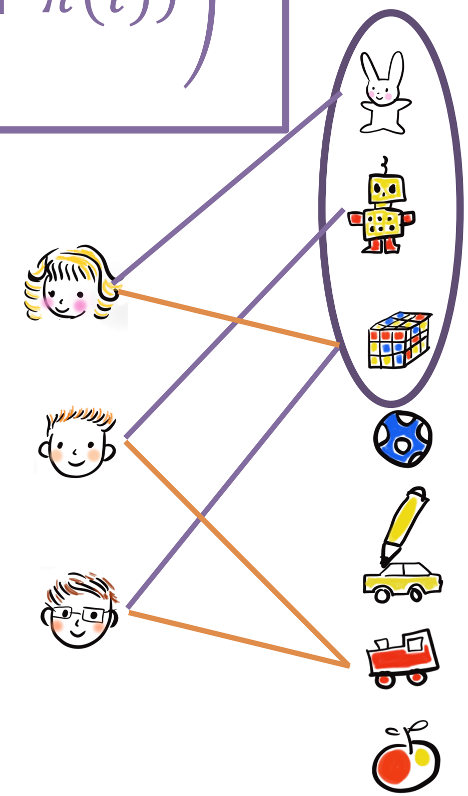
Analysis overview

LEMMA III There exists a matching $\rho: A \to H$ such that

$$\left(\prod_{i}^{n} v_i(R_i + \rho(i))\right)^{\frac{1}{n}} \geq \frac{1}{3}\left(\prod_{i}^{n} v_i(R_i + \ell(i) + \pi(i))\right)^{\frac{1}{n}}$$

Proof sketch:

- Start with infeasible allocation $R_i + \ell(i) + \pi(i)$
- For each agent getting only one of R_i, $\ell(i)$, and $\pi(i)$ preserves at least 1/3 of the allocation
- If everyone gets R_i or $\pi(i)$ we are done: return $\rho = \pi$



Analysis overview

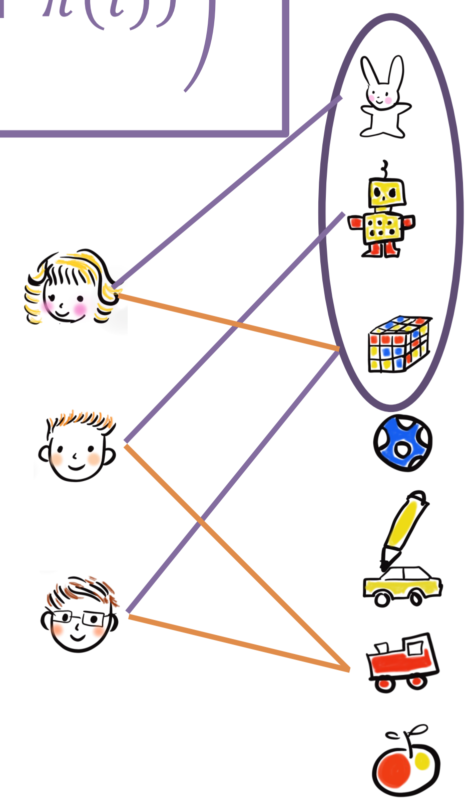
LEMMA III There exists a matching $\rho: A \to H$ such that

$$\left(\prod_i^n v_i(R_i + \rho(i))\right)^{\frac{1}{n}} \geq \frac{1}{3}\left(\prod_i^n v_i(R_i + \ell(i) + \pi(i))\right)^{\frac{1}{n}}$$

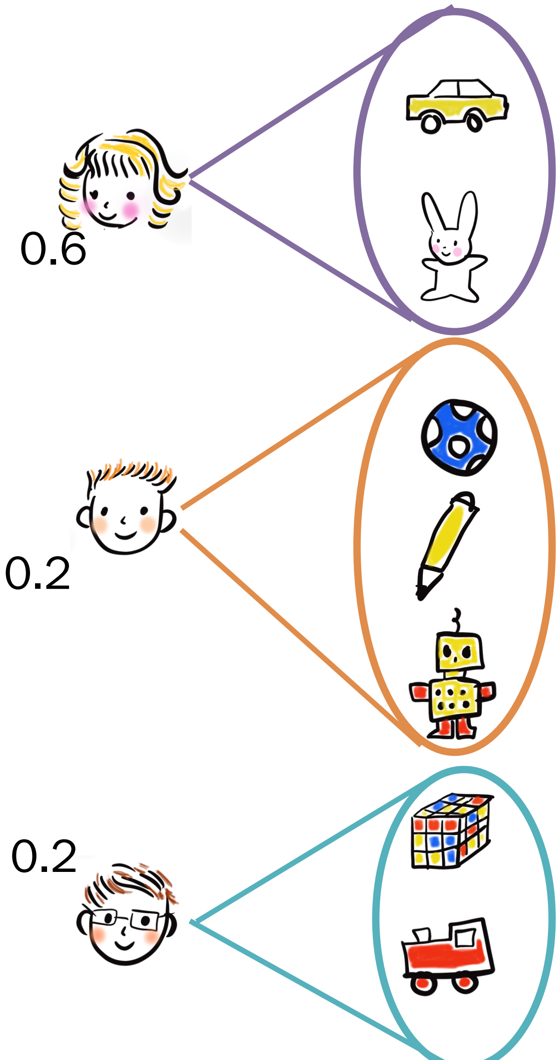
Proof sketch:

- For each agent getting only one of R_i, $\ell(i)$, and $\pi(i)$ preserves at least 1/3 of the allocation
- Starting from the $\ell(i)$'s, swap the $\ell(i)$ and $\pi(i)$ items to the initial max matching $\tau(i)$ along alternating paths.
- Key fact: for every agent i, $v_i(\tau(i)) \geq v_i(\ell(i))$

  Proof: initial matching selected $\tau(i)$ over $\ell(i)$



Generalization
Asymmetric Nash Social Welfare

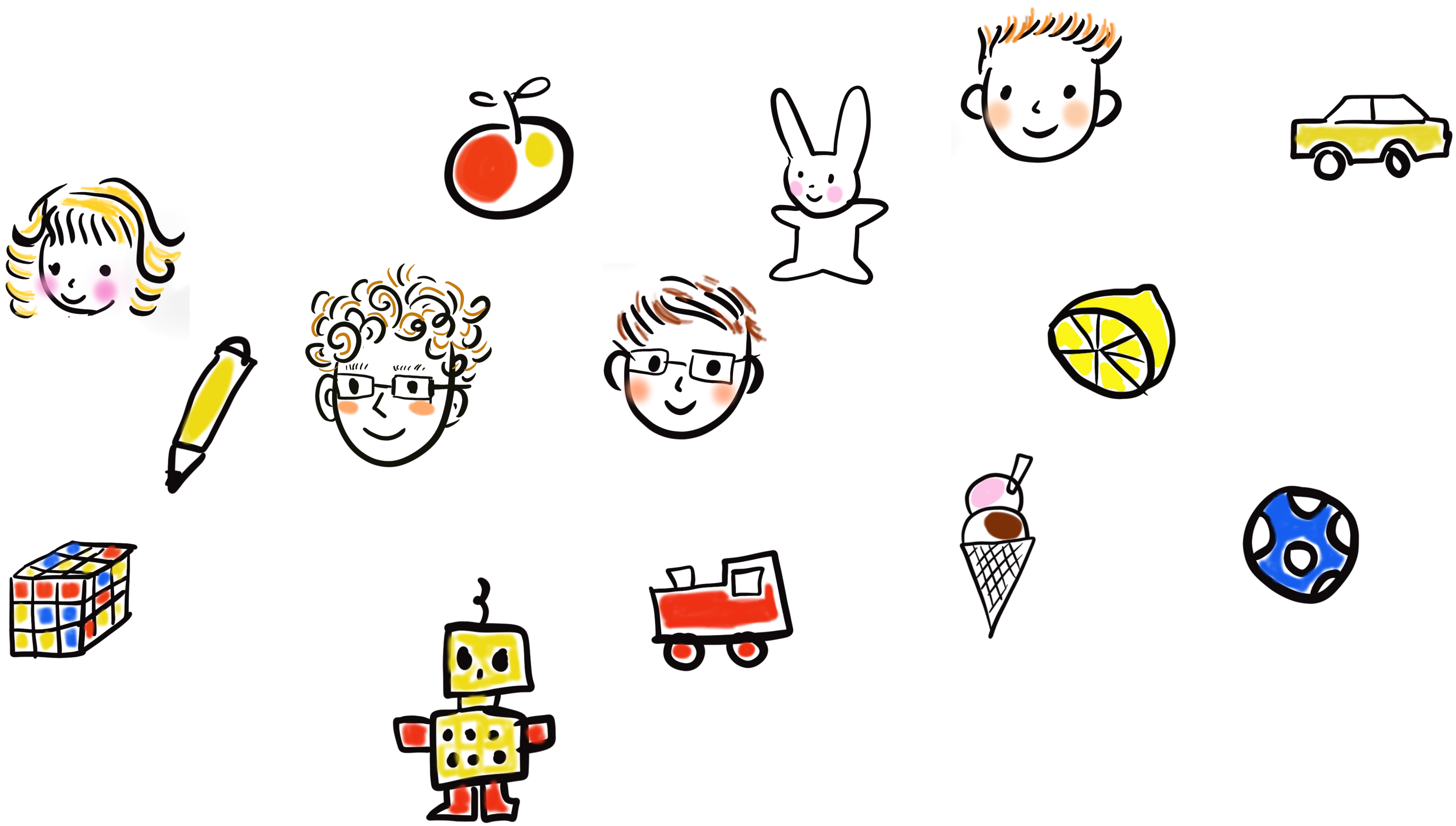


- w_i : weight for agent i, $\sum w_i = 1$

$$\text{Max} \prod_{i}^{n} v_i(S_i)^{w_i}$$

- Our algorithm gives $(2 + nw_{\max} + \varepsilon)$e-approximation
- BBKS 2020: $O(n)$ for subadditive valuations
- If $w_{\max} \gg \frac{1}{n}$, no constant factor approximation known even for linear valuations!

Thank you!



Drawings by Ágota Szuly