Seminar Approximation Algorithms

Approximating Nash Social Welfare under Submodular Valuations through (Un)Matchings

Based on a paper of the same name by Garg, Kulkarni and Kulkarni

Zeno Adrian Weil

Supervised by Dr Giovanna Varricchio

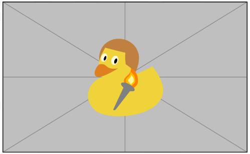
10th July 2023 · Algorithms and Complexity (Prof. Dr Martin Hoefer)

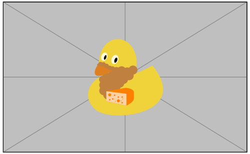
What is the issue?

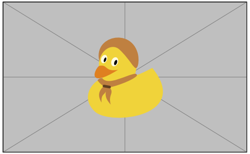
We need to distribute goods amongst recipients *fast*, *efficient* and *fairly*.

Where is this encountered?

- industrial procurement
- mobile edge computing
- satellites
- water withdrawal







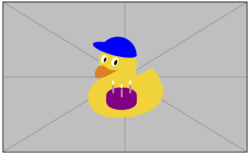


Table of Contents

Preliminaries

Allocations

Setting:

- goods: set $\mathcal{G}$ of m items
  - unsharable
  - indivisible
- recipients: set $\mathcal{A}$ of n agents

Definition (1)

An *allocation* is a tuple $\boldsymbol{x} = (\boldsymbol{x}_i)_{i \in \mathcal{A}}$ of bundles $\boldsymbol{x}_i \subset \mathcal{G}$ such that each item is element of precisely one bundle.
Item j is *assigned* to agent i if $j \in \boldsymbol{x}_i$.

But how to measure its efficiency and fairness?

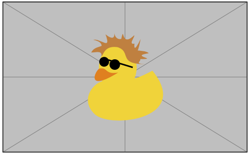
Valuation Functions

Requirements:

- monotonically non-decreasing: $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_2) \quad \forall \mathcal{S}_1 \subset \mathcal{S}_2 \subset \mathcal{G}$
- normalised: $v_i(\emptyset) = 0$
- non-negative: $v_i(\mathcal{S}) \geq 0 \quad \forall \mathcal{S} \subset \mathcal{G}$

Types:

- additive: $v_i(\mathcal{S}) := \sum_{j \in \mathcal{S}} v_i(j) \quad \forall \mathcal{S} \subset \mathcal{G}$
- submodular: $v_i(\mathcal{S}_1 \mid \mathcal{S}_2) := v_i(\mathcal{S}_1 \cup \mathcal{S}_2) - v_i(\mathcal{S}_2) \quad \forall \mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{G}$ with $\mathcal{S}_1, \mathcal{S}_2$ disjoint
  - more general (encompasses additivity)
  - diminishing returns



Asymmetric Maximum Nash Social Welfare Problem

Problem (2)

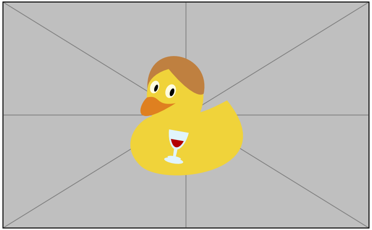
$$x^* \stackrel{!}{=} \arg\max_{x \in X_{\mathscr{A}}(\mathscr{G})} \{\mathrm{NSW}(x)\} \quad \text{with } \mathrm{NSW}(x) := \Big(\prod_{i \in \mathscr{A}} v_i(x_i)^{\eta_i}\Big)^{1/\sum_{i\in\mathscr{A}} \eta_i}$$

- $X_{\mathscr{A}}(\mathscr{G})$: all possible allocations
- η_i: agent weight

- middle ground between efficiency and fairness

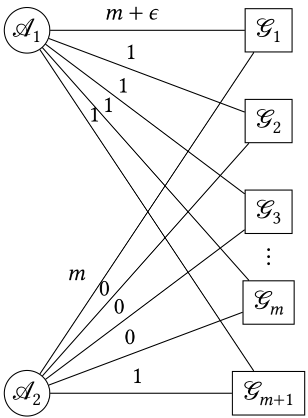
Challenge

Algorithm with approximation factor *independent from m*!



RepReMatch

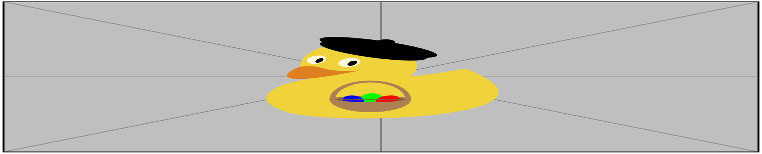
Naïve Approach



What are the low-value items?

Looking into the Future

- sort items by valuation in descending order
  - low-value items on the left



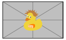
- use their valuations for edge weights in early matchings

A $2n$-approximation is possible … using SMatch.

This only works for additive valuation functions.

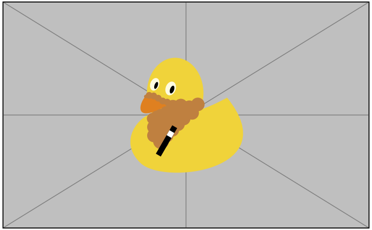
Changing the Past

Under submodular valuation, the set of lowest valuation is approximable only by $\Omega(\sqrt{m/\ln m})$.



We can change the past in three phases:

- **Phase I** Assign enough high-value items temporarily.
- **Phase II** Assign the remaining items definitely.
- **Phase III** Re-assign the items of phase I definitely.



A $2n(\log_2 n + 3)$-approximation is possible!

The Algorithm

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Phase I

1. repeat $\lceil \log_2 n \rceil + 1$ times or until $\mathscr{G} = \emptyset$
   1. create bipartite graph $G = (\mathscr{A}, \mathscr{G}, E)$ with edge weights $w(i,j) = \eta_i \log v_i(j)$
   2. compute maximum weight matching $\mathscr{M}$
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$ according to matching $\mathscr{M}$ and remove assigned items

Phase II

2. repeat until $\mathscr{G} = \emptyset$
   1. create bipartite graph $G = (\mathscr{A}, \mathscr{G}, E)$ with edge weights $w(i,j) = \eta_i \log\big(v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})\big)$
   2. compute maximum weight matching $\mathscr{M}$
   3. update bundles $\boldsymbol{x}_i^{\mathrm{II}}$ according to matching $\mathscr{M}$ and remove assigned items

Phase III

3. create bipartite graph $G = (\mathscr{A}, \bigcup_{i \in \mathscr{A}} \boldsymbol{x}_i^{\mathrm{I}}, E)$ with edge weights $w(i,j) = \eta_i \log\big(v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})\big)$
4. compute maximum weight matching $\mathscr{M}$
5. create bundles $\boldsymbol{x}_i^{\mathrm{III}}$ according to matching $\mathscr{M}$ and previous bundles $\boldsymbol{x}_i^{\mathrm{II}}$

Analysing Phase II

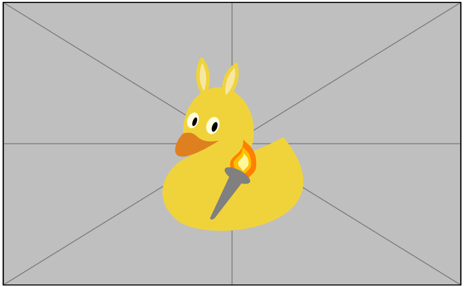
Definition (9)

The set $\mathscr{L}_{i,r}$ of *lost items* is the set of all optimal items $j \in \boldsymbol{x}_i^*$ assigned to other agents $i' \neq i$ in round r.

Definition (10)

The set of *optimal and attainable items* is defined as

$$\overline{\boldsymbol{x}}_{i,r}^* := \begin{cases} \boldsymbol{x}_i^* \setminus \bigcup_{i' \in \mathscr{A}} \boldsymbol{x}_{i'}^{\mathrm{I}} & \text{in round } r = 0, \\ \overline{\boldsymbol{x}}_{i,0}^* \setminus \mathscr{L}_{i,1} & \text{in round } r = 1, \\ \overline{\boldsymbol{x}}_{i,r-1}^* \setminus (\mathscr{L}_{i,r} \cup \{a_i^{r-1}\}) & \text{in round } r = 2, \ldots, \tau_i^{\mathrm{II}}. \end{cases}$$



$\Rightarrow$ What is the valuation of the remaining items?

Analysing Phase II — Valuation of Unassigned Items (1/4)

Lemma (11)

$$v_i(\bar{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}) \geq v_i(\bar{\boldsymbol{x}}^*_{i,1}) - \sum_{r'=1}^{r-1} |\mathscr{L}_{i,r'+1}| \cdot v_i(a_i^{r'} \mid a_i^1, \dots, a_i^{r'-1}) - v_i(a_i^1, \dots, a_i^{r-1}) \qquad \forall r \geq 2$$

- $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, \dots, a_i^{\tau_i^{\mathrm{II}}}\}$

Proof

- definition of marginal valuation

$$v_i(\bar{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}) = v_i(\bar{\boldsymbol{x}}^*_{i,r} \cup \{a_i^1, \dots, a_i^{r-1}\}) - v_i(a_i^1, \dots, a_i^{r-1})$$

$\Rightarrow$ We need a lower bound on $v_i(\bar{\boldsymbol{x}}^*_{i,r} \cup \{a_i^1, \dots, a_i^{r-1}\})$!

Analysing Phase II — Valuation of Unassigned Items (2/4)

Lemma (11)

$$v_i(\overline{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}) \geq v_i(\overline{\boldsymbol{x}}^*_{i,1}) - \sum_{r'=1}^{r-1} |\mathscr{L}_{i,r'+1}| \cdot v_i(a_i^{r'} \mid a_i^1, \dots, a_i^{r'-1}) - v_i(a_i^1, \dots, a_i^{r-1}) \qquad \forall r \geq 2$$

Proof

- $\overline{\boldsymbol{x}}^*_{i,r} = \overline{\boldsymbol{x}}^*_{i,r-1} \setminus (\mathscr{L}_{i,r} \cup \{a_i^{r-1}\}) \implies (\overline{\boldsymbol{x}}^*_{i,r} \cup \{a_i^{r-1}\}) \supset (\overline{\boldsymbol{x}}^*_{i,r-1} \setminus \mathscr{L}_{i,r})$
  - item a_i^{r-1} perhaps not element of $\overline{\boldsymbol{x}}^*_{i,r-1}$
- diminishing returns $\implies v_i(\mathcal{S}_1 \mid \mathcal{S}_2 \cup \mathcal{S}_3) \leq v_i(\mathcal{S}_1 \mid \mathcal{S}_2)$

$$\begin{aligned} v_i(\overline{\boldsymbol{x}}^*_{i,r} \cup \{a_i^1, \dots, a_i^{r-1}\}) &\geq v_i(\overline{\boldsymbol{x}}^*_{i,r-1} \setminus \mathscr{L}_{i,r} \cup \{a_i^1, \dots, a_i^{r-2}\}) \\ &= v_i(\overline{\boldsymbol{x}}^*_{i,r-1} \cup \{a_i^1, \dots, a_i^{r-2}\}) - v_i(\mathscr{L}_{i,r} \mid \overline{\boldsymbol{x}}^*_{i,r-1} \setminus \mathscr{L}_{i,r} \cup \{a_i^1, \dots, a_i^{r-2}\}) \\ &\geq v_i(\overline{\boldsymbol{x}}^*_{i,r-1} \cup \{a_i^1, \dots, a_i^{r-2}\}) - v_i(\mathscr{L}_{i,r} \mid a_i^1, \dots, a_i^{r-2}) \end{aligned}$$

Analysing Phase II — Valuation of Unassigned Items (3/4)

Lemma (11)

$$v_i(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}) \geq v_i(\overline{x}^*_{i,1}) - \sum_{r'=1}^{r-1} |\mathscr{L}_{i,r'+1}| \cdot v_i(a_i^{r'} \mid a_i^1, \dots, a_i^{r'-1}) - v_i(a_i^1, \dots, a_i^{r-1}) \qquad \forall r \geq 2$$

Proof

- apply inequality recursively

$$\begin{aligned} v_i(\overline{x}^*_{i,r} \cup \{a_i^1, \dots, a_i^{r-1}\}) &\geq v_i(\overline{x}^*_{i,r-1} \cup \{a_i^1, \dots, a_i^{r-2}\}) - v_i(\mathscr{L}_{i,r} \mid a_i^1, \dots, a_i^{r-2}) \\ &\geq v_i(\overline{x}^*_{i,1}) - \sum_{r'=1}^{r-1} v_i(\mathscr{L}_{i,r'+1} \mid a_i^1, \dots, a_i^{r'-2}) \end{aligned}$$

$\Rightarrow$ We need an upper bound on $v_i(\mathscr{L}_{i,r'+1} \mid a_i^1, \dots, a_i^{r'-2})$!

Analysing Phase II — Valuation of Unassigned Items (4/4)

Lemma (11)

$$v_i(\bar{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}) \geq v_i(\bar{x}^*_{i,1}) - \sum_{r'=1}^{r-1} |\mathcal{L}_{i,r'+1}| \cdot v_i(a_i^{r'} \mid a_i^1, \dots, a_i^{r'-1}) - v_i(a_i^1, \dots, a_i^{r-1}) \qquad \forall r \geq 2$$

Proof

- diminishing returns $\implies v_i(\mathcal{S}) \leq \sum_{j \in \mathcal{S}} v_i(j)$ for all $\mathcal{S}$
- item $a_i^{r'}$ assigned before any item $j \in \mathcal{L}_{i,r'+1} \implies v_i(a_i^{r'} \mid a_i^1, \dots, a_i^{r'-1}) \geq v_i(j \mid a_i^1, \dots, a_i^{r'-1})$

$$\begin{aligned} v_i(\mathcal{L}_{i,r'+1} \mid a_i^1, \dots, a_i^{r'-2}) &\leq \sum_{j \in \mathcal{L}_{i,r'+1}} v_i(j \mid a_i^1, \dots, a_i^{r'-2}) \\ &\leq |\mathcal{L}_{i,r'+1}| \cdot v_i(a_i^{r'} \mid a_i^1, \dots, a_i^{r'-1}) \end{aligned}$$

$\square$

Analysing Phase II — Valuation of Assigned Items

Lemma (13)

$$v_i(a_i^1, \ldots, a_i^{\tau_i^{\mathrm{II}}}) \geq v_i(\overline{\boldsymbol{x}}_{i,1}^*)/n$$

Proof

Left as exercise to the listeners.

Hint: $|\mathscr{L}_{i,r}| \leq n - 1$ $\square$

Analysing Phases I & III

Reminder

Phase I Temporary assignments. Valuations of single items as edge weights.

Phase III Items of phase I released. Valuations of items and bundles from phase II as edge weights.

Phase I reserves 'high-value' items. But what qualifies as 'high-value'?

Definition (14)

Let $\boldsymbol{x}_i^* = \{o_i^1, o_i^2, \dots\}$ be an optimal bundle. An item $j \in \mathcal{G}$ is *outstanding* if $v_i(j) \geq v_i(o_i^1)$.

$\Rightarrow$ Are enough outstanding items reserved?

Analysing Phases I & III — todo

Lemma (15)

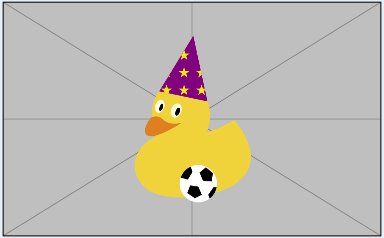
Each agent can be matched with an outstanding item in phase III.

Sketch Proof

- number of agents not matched with an outstanding item in phase III halved with each round of phase I
  - induction on number of rounds in phase I
- $\lceil \log_2 n \rceil + 1$ rounds in phase I are enough

Base Case: In round 1 of phase I, either

- $\geq n/2$ many agents matched with an outstanding item
- $< n/2$ many agents matched with an outstanding item
  - $> n/2$ many items o_i^1 assigned to someone else
  - $> n/2$ many agents matched upon release in phase III



$\square$

The Approximation Factor

Theorem (16)

The approximation factor is $2n(\log_2 n + 3)$.