GOETHE UNIVERSITÄT FRANKFURT AM MAIN

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

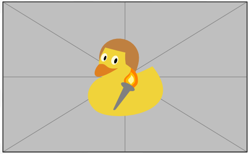
4th July 2023 · Algorithms and Complexity (Prof. Dr Martin Hoefer)

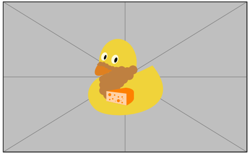
What is the issue?

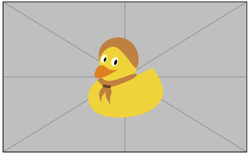
We need to distribute goods amongst recipients *fast*, *efficient* and *fairly*.

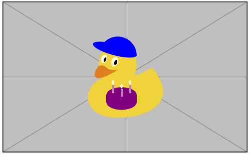
Where is this encountered?

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









Allocations

Setting:

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

Definition

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 the efficiency and fairness of the allocation?

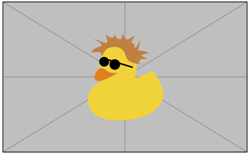
Valuation Functions

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



Asymmetric Maximum Nash Social Welfare Problem

Definition

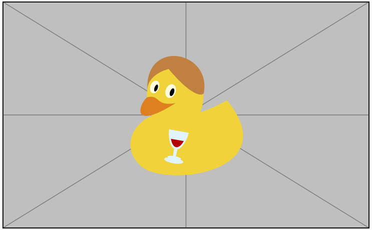
$$\boldsymbol{x}^* \stackrel{!}{=} \arg\max_{\boldsymbol{x} \in X_{\mathscr{A}}(\mathscr{G})} \{\mathrm{NSW}(\boldsymbol{x})\} \quad \text{with } \mathrm{NSW}(\boldsymbol{x}) := \Big(\prod_{i \in \mathscr{A}} v_i(\boldsymbol{x}_i)^{\eta_i}\Big)^{1/\sum_{i \in \mathscr{A}} \eta_i}$$

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

- middle ground between efficiency and fairness

Challenge

Algorithm with approximation factor *independent from m*!



GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Naiver Ansatz

Inhalt...