Algorithms and Complexity Group
Institute for Computer Science
Goethe University Frankfurt

Summer Semester 2023

Seminar Approximation Algorithms

ANSWuSVþ(U)M

Zeno Adrian Weil

19th May 2023

Supervisor: Dr Giovanna Varricchio

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Todo list

1 Introduction

- problem introduction, motivation, applications
- formal problem definition
- short literature review: What is known, what not? New findings?
- content & structure of paper

Definition 1. Let $\mathcal{G} := \{1, \dots, m\}$ be a set of indivisible *items* and $\mathcal{A} := \{1, \dots, n\}$ be a set of *agents*. An *allocation* is a tuple $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \in \mathcal{P}(G)^n$ such that each item is element of exactly one $\boldsymbol{x}_i$, that is $\bigcup_{i \in \mathcal{A}} \boldsymbol{x}_i = \mathcal{G}$ and $\boldsymbol{x}_i \cap \boldsymbol{x}_{i'} = \emptyset$ for all $i \neq i'$. An item $j \in \mathcal{G}$ is *assigned* to agent $i \in \mathcal{A}$ if $j \in \boldsymbol{x}_i$ holds.

or rather 'alloc-ated'?

⋮

Definition 2. Given a set $\mathcal{G}$ of items and a set $\mathcal{A}$ of agents with *valuations* $v_i : \mathcal{P}(\mathcal{G}) \to \mathbb{R}$ and *agent weights* η_i for all agents $i \in \mathcal{A}$, the *Nash Social Welfare problem* (NSW) is to find an allocation maximising the weighted geometric mean of valuations, that is

i: less strict def of valu-ations; restriction for our case later on

$$\underset{\boldsymbol{x} \in \Pi_n(\mathcal{G})}{\arg\max} \left\{ \left(\prod_{i \in \mathcal{A}} v_i(\boldsymbol{x}_i)^{\eta_i} \right)^{1/\sum_{i \in \mathcal{A}} \eta_i} \right\}$$

where $\Pi_n(\mathcal{G})$ is the set of all possible allocations of the items in $\mathcal{G}$ amongst n agents. The problem is called *symmetric* if all agent weights η_i are equal, and *asymmetric* otherwise.

⋮

In a slight abuse of notation, we omit the brackets in a valuation function if the set of items contains only one item, that is $v_i(j) = v_i(\{j\})$.

⋮

definition of approximation factor [def environment or in-text?]

⋮

Garg, Kulkarni and Kulkarni consider five different types of non-negative monotonically non-decreasing valuation functions of which we are going to consider only the following two due to space constraints:

Additive The valuation $v_i(\mathcal{S})$ of an agent i for a set $\mathcal{S} \subset \mathcal{G}$ of items j is the sum of individual valuations $v_i(j)$, that is $v_i(\mathcal{S}) = \sum_{j \in \mathcal{S}} v_i(j)$.

Submodular Let $v_i(\mathcal{S}_1 \mid \mathcal{S}_2) := v_i(\mathcal{S}_1 \cup \mathcal{S}_2) - v_i(\mathcal{S}_2)$ denote the marginal utility of agent i for a set $\mathcal{S}_1 \subset \mathcal{G}$ of items over the disjoint set $\mathcal{S}_2 \subset \mathcal{G}$. This valuation functions satisfies the submodularity constraint $v_i(j \mid \mathcal{S}_1 \cup \mathcal{S}_2) \leq v_i(j \mid \mathcal{S}_1)$ for all agents $i \in \mathcal{A}$, items $j \in \mathcal{G}$ and sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{G}$ of items.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2 SMatch

- giving intuition why naïve repeated maximum matching is not independent of m (cf. Figure 1)
- textual walk-through of algorithm
- giving intuition of lemmas used and theorem proof

Proofs are short so this section might not be much shorter than the original one.

On second thought, this section could possibly be shortened even more by omitting the theory and getting merged with the introduction, depending on the length of the rest of the document.

For a fixed agent i, order the items in descending order of the valuations by agent i and denote the j-th most liked item by $\mathcal{G}_i^{(j)}$.

probably to be mentionend earlier because of u_i

Lemma 1. $v_i(h_i^t) \geq v_i(\mathcal{G}_i^{(tn)})$.

Proof. At the start of the t-th iteration, at most $(t-1)n$ items out of the tn most highly valued items $\mathcal{G}_i^{(1)}, \dots, \mathcal{G}_i^{(tn)}$ have been assigned in previous iterations since at most n items are assigned in each iteration. During the t-th iteration, at most $n-1$ more of those highly valued items could be assigned to all other agents $i' \neq i$, leaving at least one item in $\mathcal{G}_i^{(1)}, \dots, \mathcal{G}_i^{(tn)}$ unassigned. Since $v_i(\mathcal{G}_i^{(k)}) \geq v_i(\mathcal{G}_i^{(tn)})$ for all $k \leq i$ by definition, the lemma follows. □

Lemma 2. $v_i(h_i^2, \dots, h_i^{\tau_i}) \geq \frac{u_i}{n}$.

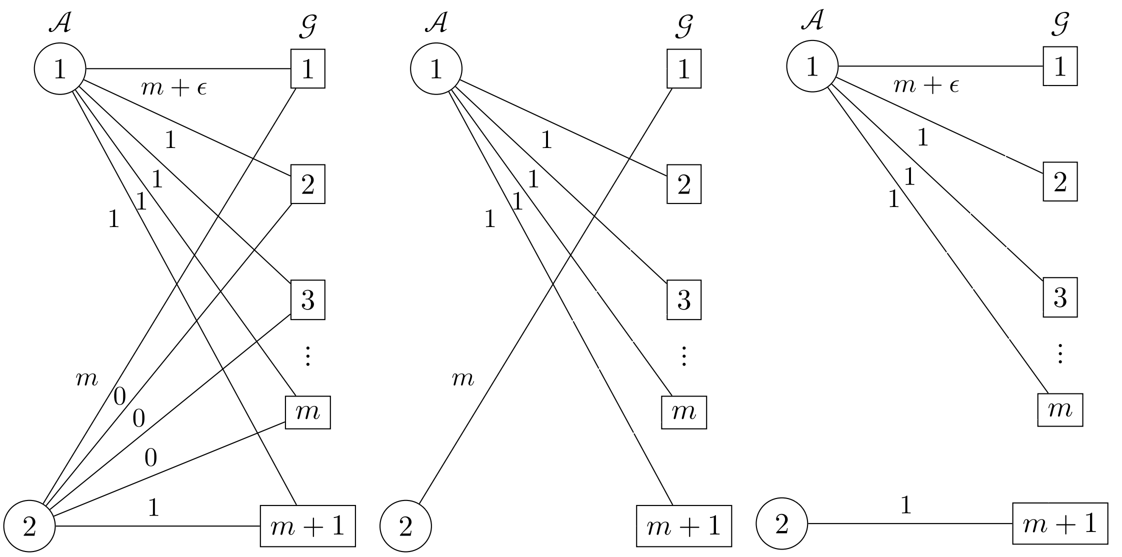
Proof Sketch. □

Algorithm 1: SMatch for the Asymmetric Additive NSW problem

Input: set $\mathcal{A} = \{1, \dots, n\}$ of agents with weights $\eta_i \forall i \in \mathcal{A}$, set $\mathcal{G} = \{1, \dots, m\}$ indivisible items, additive valuations $v_i \colon \mathcal{P}(\mathcal{G}) \to \mathbb{R}^+_{>0}$ where $v_i(\mathcal{S})$ is the valuation of agent $i \in \mathcal{A}$ for each item set $\mathcal{S} \subset \mathcal{G}$

Output: $\frac{1}{2n}$-approximation $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ of an optimal allocation

$\boldsymbol{x}_i \leftarrow \emptyset \quad \forall i \in \mathcal{A}$
$u_i \leftarrow v_i(\mathcal{G}_{i,[2n+1:m]}) \quad \forall i \in \mathcal{A}$
$\mathcal{W} \leftarrow \{\, \eta_i \cdot \log(v_i(j) + \frac{u_i}{n}) \mid i \in \mathcal{A}, j \in \mathcal{G} \,\}$ ▷ *edge weights*
$G \leftarrow (\mathcal{A}, \mathcal{G}, \mathcal{W})$ ▷ *bipartite graph*
$\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$
$\boldsymbol{x}_i \leftarrow \{\, j \mid (i,j) \in \mathcal{M} \,\} \quad \forall i \in \mathcal{A}$ ▷ *allocate according to matching*
$\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G} \setminus \{\, j \mid (i,j) \in \mathcal{M} \,\}$ ▷ *remove allocated goods*
while $\mathcal{G}^{\text{rem}} \neq \emptyset$ **do**
    $\mathcal{W} \leftarrow \{\, \eta_i \cdot \log(v_i(j) + v_i(\boldsymbol{x}_i)) \mid i \in \mathcal{A}, j \in \mathcal{G}^{\text{rem}} \,\}$
    $G \leftarrow (\mathcal{A}, \mathcal{G}^{\text{rem}}, \mathcal{W})$
    $\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$
    $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i \cup \{\, j \mid (i,j) \in \mathcal{M} \,\} \quad \forall i \in \mathcal{A}$
    $\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G}^{\text{rem}} \setminus \{\, j \mid (i,j) \in \mathcal{M} \,\}$
end while
return $\boldsymbol{x}$



(a) problem instance **(b)** OPT $= \sqrt{m \cdot m}$ **(c)** ALG $\leq \sqrt{(2m + \epsilon - 1) \cdot 1}$

might delete if document in two columns → unused edges in grey

Figure 1: Agent 1 values item 1 at $m + \epsilon$, and all other items at 1. Agent 2 values item 1 at m, item $m+1$ at 1, and all other items at 0. In an optimal allocation, item 1 would be assigned to agent 2 and all other items to agent 1, resulting in a NSW of $\sqrt{m \cdot m} = m$. A repeated maximum matching algorithm would greedily assign item 1 to agent 1 and item $m+1$ to agent 2 in the first round. Even if all remaining items were going to be assigned to agent 1, the NSW will never surpass $\sqrt{(2m + \epsilon - 1) \cdot 1} < \sqrt{2m}$. The approximation factor $\alpha \approx \sqrt{m/2}$ therefore depends on the number of items.

Algorithm 2: RepReMatch for the Asymmetric Submodular NSW problem

Input: set $\mathcal{A} = \{1, \dots, n\}$ of agents with weights $\eta_i \forall i \in \mathcal{A}$, set $\mathcal{G} = \{1, \dots, m\}$ indivisible items, additive valuations $v_i \colon \mathcal{P}(\mathcal{G}) \to \mathbb{R}^+_{>0}$ where $v_i(\mathcal{S})$ is the valuation of agent $i \in \mathcal{A}$ for each item set $\mathcal{S} \subset \mathcal{G}$

Output: $\frac{1}{2n \log n}$-approximation $\boldsymbol{x}^{\mathrm{III}} = (\boldsymbol{x}_1^{\mathrm{III}}, \dots, \boldsymbol{x}_n^{\mathrm{III}})$ of an optimal allocation

Phase I:

$\boldsymbol{x}_i^{\mathrm{I}} \leftarrow \emptyset \quad \forall i \in \mathcal{A}$

$\mathcal{G}^{\mathrm{rem}} \leftarrow \mathcal{G}$

for $t = 0, \dots, \lceil \log n \rceil - 1$ **do**

  if $\mathcal{G}^{\mathrm{rem}} \neq \emptyset$ **then**

    $\mathcal{W} \leftarrow \{ \eta_i \cdot \log(v_i(j)) \mid i \in \mathcal{A}, j \in \mathcal{G} \}$

    $G \leftarrow (\mathcal{A}, \mathcal{G}, \mathcal{W})$

    $\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$

    $\boldsymbol{x}_i^{\mathrm{I}} \leftarrow \boldsymbol{x}_i^{\mathrm{I}} \cup \{j\} \quad \forall (i,j) \in \mathcal{M}$

    $\mathcal{G}^{\mathrm{rem}} \leftarrow \mathcal{G}^{\mathrm{rem}} \setminus \{ j \mid (i,j) \in \mathcal{M} \}$

  end if

end for

Phase II:

$\boldsymbol{x}_i^{\mathrm{II}} \leftarrow \emptyset \quad \forall i \in \mathcal{A}$

while $\mathcal{G}^{\mathrm{rem}} \neq \emptyset$ **do**

  $\mathcal{W} \leftarrow \{ \eta_i \cdot \log(v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})) \mid i \in \mathcal{A}, j \in \mathcal{G} \}$

  $G \leftarrow (\mathcal{A}, \mathcal{G}, \mathcal{W})$

  $\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$

  $\boldsymbol{x}_i^{\mathrm{II}} \leftarrow \boldsymbol{x}_i^{\mathrm{II}} \cup \{j\} \quad \forall (i,j) \in \mathcal{M}$

  $\mathcal{G}^{\mathrm{rem}} \leftarrow \mathcal{G}^{\mathrm{rem}} \setminus \{ j \mid (i,j) \in \mathcal{M} \}$

end while

Phase III:

$\mathcal{G}^{\mathrm{rem}} \leftarrow \bigcup_{i \in \mathcal{A}} \boldsymbol{x}_i^{\mathrm{I}}$ ▷ *release items allocated in first phase*

$\mathcal{W} \leftarrow \{ \eta_i \cdot \log(v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})) \mid i \in \mathcal{A}, j \in \mathcal{G} \}$

$G \leftarrow (\mathcal{A}, \mathcal{G}, \mathcal{W})$

$\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$

$\boldsymbol{x}_i^{\mathrm{III}} \leftarrow \boldsymbol{x}_i^{\mathrm{II}} \cup \{j\} \quad \forall (i,j) \in \mathcal{M}$

$\mathcal{G}^{\mathrm{rem}} \leftarrow \mathcal{G}^{\mathrm{rem}} \setminus \{ j \mid (i,j) \in \mathcal{M} \}$

$\boldsymbol{x}_i^{\mathrm{III}} \leftarrow \text{arbitrary\_allocation}(\mathcal{A}, \mathcal{G}^{\mathrm{rem}}, \boldsymbol{x}_i^{\mathrm{III}}, (v_i)_{i \in \mathcal{A}})$

decide on $\boldsymbol{x}$ or $\boldsymbol{x}^{\mathrm{III}}$; discrepancies in orig paper in def of $\boldsymbol{x}_{\mathrm{III}}$!

return $\boldsymbol{x}^{\mathrm{III}}$