

Seminar Approximation Algorithms

ANSWuSVp(U)M

Zeno Adrian Weil

8th June 2023

Examiner: Prof Dr Martin Hoefer
Supervisor: Dr Giovanna Varricchio

Abstract

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Todo list

■	or rather ‘allocated’?	4
■	mention divisibility	4
■	agents without items assigned have valuation zero \rightarrow prevent	4
■	definition of approximation factor [def environment or in-text?]	4
■	or rather utility?	4
■	What is the motivation for submodular functions? [IRfCAwSUF]	4
■	fix pos of subcaptions & of agent 2	6
■	What is the factor?	6
■	<i>i</i> : I do not get the reason for the extra step in the original paper.	8
■	<i>i</i> : Would it be ‘dirty’ to include notation in the definitions?	11
■	Do that or, alternatively, find a paper showing the other def. [IRfCAwSUF]	11
■	ditto	12
■	include if space enough	12
■	change proof when intro is finished	12
■	wrong! the union contains it though	13
■	<i>i</i> : Error in paper? see also lemma 5	14
■	<i>i</i> : Should it not be $\frac{e-1}{e} - \epsilon$?	15

1 Introduction

- problem introduction, motivation, applications
- formal problem definition (incl. why geometric mean?)
- short literature review: What is known, what not? New findings?
- content & structure of paper

Definition 1. Let \mathcal{G} be a set of size m and \mathcal{A} be a set of size n . The elements of set \mathcal{G} are called *items*, and the elements of set \mathcal{A} are called *agents*. An *allocation* is a tuple $\mathbf{x} = (\mathbf{x}_i)_{i \in \mathcal{A}}$ of so-called *bundles* $\mathbf{x}_i \subset \mathcal{G}$ such that each item is element of exactly one bundle, that is $\bigcup_{i \in \mathcal{A}} \mathbf{x}_i = \mathcal{G}$ and $\mathbf{x}_i \cap \mathbf{x}_{i'} = \emptyset$ for all $i \neq i'$. An item $j \in \mathcal{G}$ is *assigned to* agent $i \in \mathcal{A}$ if $j \in \mathbf{x}_i$ holds.

or rather
'alloc-
ated'?

mention divisibility

:

Problem 1. Given a set \mathcal{G} of items and a set \mathcal{A} of agents with *valuation functions* $v_i: \mathcal{P}(\mathcal{G}) \rightarrow \mathbb{R}$ 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

$$\arg \max_{\mathbf{x} \in \mathbf{X}_n(\mathcal{G})} \left\{ \left(\prod_{i \in \mathcal{A}} v_i(\mathbf{x}_i)^{\eta_i} \right)^{1/\sum_{i \in \mathcal{A}} \eta_i} \right\}$$

where $\mathbf{X}_n(\mathcal{G})$ 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.

agents without items assigned have valuation zero \rightarrow prevent

:

In a slight abuse of notation, we omit curly braces delimiting a set in the arguments of a valuation function, so we write $v_i(j_1, j_2, \dots)$ but mean $v_i(\{j_1, j_2, \dots\})$ for example.

:

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

:

Garg, Kulkarni and Kulkarni [4] 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 valuation of agent i for a set $\mathcal{S}_1 \subset \mathcal{G}$ of items over a *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.

or rather
utility?

What is the motivation for submodular functions? [IRfCAwSUF]

We use *additive NSW* and *submodular NSW* as shorthands for the Nash social welfare problems with additive and submodular valuation functions, respectively.

- non-sharable indivisible resources -> items; opposed to divisible and sharable ones like money, electricity, network connections
- different types of efficiency: Pareto optimal, utilitarian, egalitarian, Nash
- additive: \mathcal{NP} -hard 1,069 (<https://www.semanticscholar.org/reader/1c5347490b006dc6c498131a9ee01281a31edbe7>) matroids?
- submodularity: discrete analogue of concavity and arises naturally in economic setting since it captures the property that marginal utilities are decreasing as we allocate more goods
 - alt def: <https://www.semanticscholar.org/reader/1c5347490b006dc6c498131a9ee01281a31edbe7>
- NSW [Lee17] [CKM⁺16]: \mathcal{NP} -hard 1,5819, EF1, PO, approximation recover most guarantees; <https://www.semanticscholar.org/reader/1c5347490b006dc6c498131a9ee01281a31edbe7> invariance under scaling
- mention Nash’s name in footnote
- especially asymmetric case less understood <https://www.semanticscholar.org/reader/bef4514574543720b>
- Utilitarian SW completely resolved [Von08] [KLMM08]: constant-factor $\frac{e}{e-1}$, hardness result
- maximin somewhat resolved [AS10] [KP07] [BD05]: additive $(\sqrt{n} \log^3 n)$ -factor; submodular (n) -factor; hardness of 2 for both
- symmetric additive(-like) NSW [BKV18]: $e^{1/e}$ -factor
- asymmetric submodular NSW [NR14] [CDG⁺17]: not much known; (m) -factor
- applications
 - industrial procurement: capture of preferences? in bundles or singular? capture of business rules (physical objects, configuration, synergy) [2]
 - earth observation satellite: co-founded -> weights reflecting investment -> utility must reflect that; sharable as data can be distributed multiple times [2]
 - manufacturing system: task scheduling across production resources -> network of dependencies; dynamic, real-time calculation (feasibility) more important than optimality because of disturbance -> distributed way by means of cooperation and coordination; rescheduling [2]
 - combinatorial auctions: FCC auction of spectrum licenses [5]
 - edge computing: ‘paradigm in which the resources for communication, computation, control and storage are placed at the edge of the Internet, in close proximity to mobile devices, sensors, actuators, connected things and end users.’; highly competitive environment; locality, latency, security, incentivisation [1, 3]
- B. R. Chaudhury, Y. K. Cheung, J. Garg, N. Garg, M. Hoefer, and K. Mehlhorn. Fair division of indivisible goods for a class of concave valuations. *J. Artif. Intell. Res.*, 74:111–142, 2022

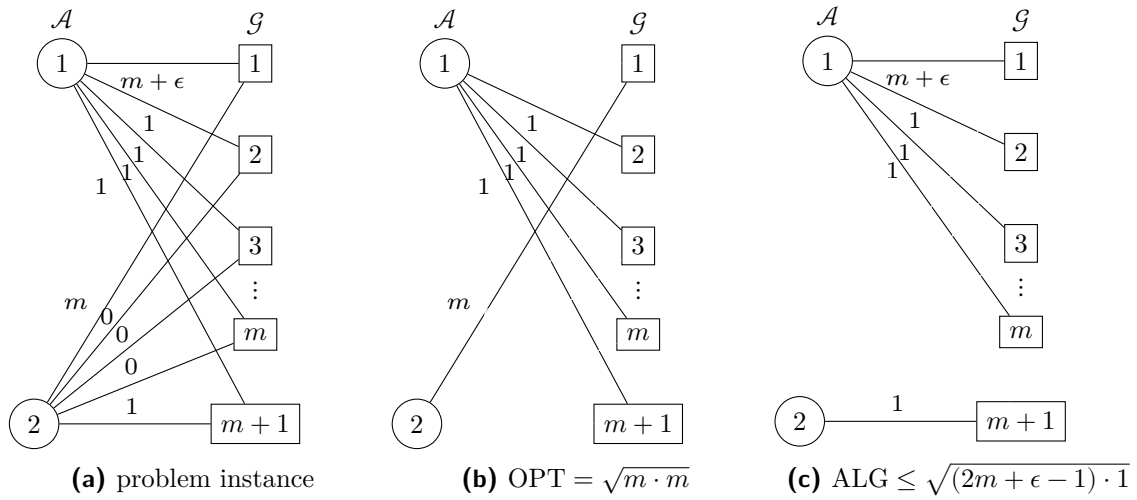


Figure 1: An example showing that simple repeated matching without consideration of the future leads to an approximation factor dependent on the number of items. Agent 1 values item 1 at $m + \epsilon$ and all other items at 1. Agent 2 values item 1 at m , item $m + 1$ at 1 and all other items at 0. 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.

fix pos of
subcap-
tions & of
agent 2

2 SMatch

In the case of an equal number of agents and items, i.e., $n = m$, the *additive* NSW can be solved exactly by finding a maximum matching on a bipartite graph with the sets of agents and of items as its parts; as weight of the edge between agent i and item j , use $\eta_i \log v_i(j)$, that is the weighted valuation of item j by agent i in the *logarithmic* Nash social welfare. Should there be more items than agents, then it would be obvious at first to just repeatedly find a maximum matching and assign the items accordingly until all items are assigned. The flaw of this idea is that such a greedy algorithm only considers the valuations of items in the current matching and perhaps the valuations of items already assigned. As the example in fig. 1 demonstrates, this leads to an algorithm with an approximation factor dependent on the number m of items. The geometric mean of the NSW favours allocations with similarly valued bundles, wherefore it may be beneficial to give items to agents who cannot expect many more valuable items in the future instead of to agents who value the item a bit more but do so for other items as well.

What is
the factor?

The algorithm SMatch, described in algorithm 1, eliminates the flaw by first gaining foresight of the valuations of items assigned after the first matching, achieving an approximation factor of $2n$ (cf. theorem 1 later on). For a fixed agent i , order the items in descending order of valuations and denote the j -th most liked item by \mathcal{G}_i^j . To obtain a well-defined order, items of equal rank are further ordered numerically. SMatch does in fact repeatedly match items. During the first matching, however, the edge weights are defined as $\eta_i \log(v_i(j) + u_i/n)$ for an edge between agent i and item j . The addend u_i serves as estimation of the valuation of items assigned after the first matching (cf. lemma 2) and is

Algorithm 1: SMatch for the Asymmetric Additive NSW problem

Input : set \mathcal{G} of m items, set \mathcal{A} of n agents, additive valuation functions $v_i: \mathcal{P}(\mathcal{G}) \rightarrow \mathbb{R}_0^+$ and weights η_i for all agents $i \in \mathcal{A}$

Output: $\frac{1}{2n}$ -approximation $\mathbf{x} = (\mathbf{x}_i)_{i \in \mathcal{A}}$ of an optimal allocation

```

1  $\mathbf{x}_i \leftarrow \emptyset \quad \forall i \in \mathcal{A}$ 
2  $u_i \leftarrow v_i(\mathcal{G}_i^{2n+1}, \dots, \mathcal{G}_i^m) \quad \forall i \in \mathcal{A}$   $\triangleright$  estimation of future valuations
3  $\mathcal{W} \leftarrow \{\eta_i \cdot \log(v_i(j) + u_i/n) \mid i \in \mathcal{A}, j \in \mathcal{G}\}$   $\triangleright$  edge weights
4  $G \leftarrow (\mathcal{A}, \mathcal{G}, \mathcal{W})$   $\triangleright$  bipartite graph
5  $\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$ 
6  $\mathbf{x}_i \leftarrow \{j \mid (i, j) \in \mathcal{M}\} \quad \forall i \in \mathcal{A}$   $\triangleright$  assign according to matching
7  $\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G} \setminus \{j \mid (i, j) \in \mathcal{M}\}$   $\triangleright$  remove assigned items
8 while  $\mathcal{G}^{\text{rem}} \neq \emptyset$  do
9    $\mathcal{W} \leftarrow \{\eta_i \cdot \log(v_i(j) + v_i(\mathbf{x}_i)) \mid i \in \mathcal{A}, j \in \mathcal{G}^{\text{rem}}\}$ 
10   $G \leftarrow (\mathcal{A}, \mathcal{G}^{\text{rem}}, \mathcal{W})$ 
11   $\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$ 
12   $\mathbf{x}_i \leftarrow \mathbf{x}_i \cup \{j \mid (i, j) \in \mathcal{M}\} \quad \forall i \in \mathcal{A}$ 
13   $\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G}^{\text{rem}} \setminus \{j \mid (i, j) \in \mathcal{M}\}$ 
14 end while
15 return  $\mathbf{x}$ 

```

defined as

$$u_i := \min_{\substack{\mathcal{S} \subseteq \mathcal{G} \\ |\mathcal{S}| \leq 2n}} \{v_i(\mathcal{G} \setminus \mathcal{S})\} = v_i(\mathcal{G}_i^{2n+1}, \dots, \mathcal{G}_i^m). \quad (1)$$

The set \mathcal{S} has less than $2n$ elements only if there are less than $2n$ items in total. From the second matching onwards, the edge weights are defined as $\eta_i \log(v_i(j) + v_i(\mathbf{x}_i))$, where \mathbf{x}_i is the continuously updated bundle of agent i . The addend $v_i(\mathbf{x}_i)$ could lead to better allocations in applications, but does not improve the approximation factor asymptotically.

To calculate the approximation factor of SMatch, we first need to establish a lower bound on the valuation of single items. For convenience, we order the items in the final bundle $\mathbf{x}_i = \{h_i^1, \dots, h_i^{\tau_i}\}$ of agent i by the order in which they were assigned, so that item h_i^t is assigned according to the t -th matching. Note that it holds $v_i(h_i^t) \geq v_i(h_i^{t'})$ for all $t' \geq t$.

Lemma 1. *For each agent $i \in \mathcal{A}$, her final bundle $\mathbf{x}_i = \{h_i^1, \dots, h_i^{\tau_i}\}$, and her tn -th most highly valued item \mathcal{G}_i^{tn} , it holds $v_i(h_i^t) \geq v_i(\mathcal{G}_i^{tn})$ for all $t = 1, \dots, \tau_i$.*

Proof. Before the t -th matching, no more than $(t-1)n$ items out of the tn most highly valued items $\mathcal{G}_i^1, \dots, \mathcal{G}_i^{tn}$ have been assigned in previous matchings since at most n many out of those items are assigned each time. Because of the t -th matching, at most $n-1$ more could be assigned to all other agents $i' \neq i$, leaving at least one item of $\mathcal{G}_i^1, \dots, \mathcal{G}_i^{tn}$ unassigned. Since $v_i(\mathcal{G}_i^k) \geq v_i(\mathcal{G}_i^{tn})$ for all $k \leq tn$ by definition of \mathcal{G}_i , the lemma follows. \square

We can now establish $u_i/n = v_i(\mathcal{G}_i^{2n+1}, \dots, \mathcal{G}_i^m)/n$ as lower bound on the valuations of items assigned after the first matching.

Lemma 2. *For each agent $i \in \mathcal{A}$ and her final bundle $\mathbf{x}_i = \{h_i^1, \dots, h_i^{\tau_i}\}$, it holds $v_i(h_i^2, \dots, h_i^{\tau_i}) \geq u_i/n$.*

Proof. By lemma 1 and definition of \mathcal{G}_i , every item h_i^t is worth at least as much as each item \mathcal{G}_i^{tn+k} with $k \in \{1, \dots, n\}$ and, consequently, its valuation $v_i(h_i^t)$ is at least as high

as the mean valuation $\frac{1}{n}v_i(\mathcal{G}_i^{tn+1}, \dots, \mathcal{G}_i^{tn+n})$. Further, it holds $\tau_i n + n \geq m$ since agent i gets assigned $\tau_i \geq \lfloor \frac{m}{n} \rfloor \geq \frac{m}{n} - 1$ many items. Together, this yields

$$v_i(h_i^2, \dots, h_i^{\tau_i}) = \sum_{t=2}^{\tau_i} v_i(h_i^t) \geq \sum_{t=2}^{\tau_i} \frac{1}{n} v_i(\mathcal{G}_i^{tn+1}, \dots, \mathcal{G}_i^{tn+n}) \geq \frac{1}{n} v_i(\mathcal{G}_i^{2n+1}, \dots, \mathcal{G}_i^m) = \frac{u_i}{n}. \quad (2)$$

i: I do not get the reason for the extra step in the original paper.

□

Remark 1. In lemma 2, we assumed non-zero valuations for all items, hence the bundle lengths of $\tau_i \geq \lfloor \frac{m}{n} \rfloor$. Of course in an actual program, one would not assign items to agents who value them at zero. Nevertheless, lemma 2 still holds inasmuch as additional zero valuations in eq. (2) do not change the sum.

This allows us to calculate an approximation factor for SMatch by comparing its output with an optimal allocation \mathbf{x}^* .

Theorem 1. *SMatch has an approximation factor of $2n$.*

Proof. Lemma 2 can be plugged into the logarithmic NSW:

$$\log \text{NSW}(\mathbf{x}) = \frac{1}{\sum_{i \in \mathcal{A}} \eta_i} \cdot \sum_{i \in \mathcal{A}} \eta_i \log v_i(h_i^1, \dots, h_i^{\tau_i}) \quad (3)$$

$$= \frac{1}{\sum_{i \in \mathcal{A}} \eta_i} \cdot \sum_{i \in \mathcal{A}} \eta_i \log(v_i(h_i^1) + v_i(h_i^2, \dots, h_i^{\tau_i})) \quad (4)$$

$$\geq \frac{1}{\sum_{i \in \mathcal{A}} \eta_i} \cdot \sum_{i \in \mathcal{A}} \eta_i \log(v_i(h_i^1) + u_i/n) \quad (5)$$

Notice that the first matching of SMatch maximises the sum in eq. (5). Thus, assigning all agents i their respective most highly valued item g_i^1 in an optimal bundle $\mathbf{x}_i^* = \{g_i^1, \dots, g_i^{\tau_i^*}\}$ yields the even lower bound

$$\log \text{NSW}(\mathbf{x}) \geq \frac{1}{\sum_{i \in \mathcal{A}} \eta_i} \cdot \sum_{i \in \mathcal{A}} \eta_i \log(v_i(g_i^1) + u_i/n). \quad (6)$$

Recall the definition of u_i from eq. (1). Consider a slightly modified variant:

$$u_i = v_i(\mathcal{G} \setminus \mathcal{S}_i) \text{ with } \mathcal{S}_i := \arg \min_{\substack{\mathcal{S} \subset \mathcal{G} \\ |\mathcal{S}| \leq 2n}} \{v_i(\mathcal{G} \setminus \mathcal{S})\} \quad (7)$$

Moreover, consider the set \mathcal{S}_i^* of the (at most) $2n$ most highly valued items in the optimal bundle \mathbf{x}_i^* , i. e.

$$\mathcal{S}_i^* := \arg \min_{\substack{\mathcal{S} \subset \mathcal{G} \\ |\mathcal{S}| \leq 2n}} \{v_i(\mathbf{x}_i^* \setminus \mathcal{S})\}. \quad (8)$$

It holds $v_i(g_i^1) \geq \frac{1}{2n} v_i(\mathcal{S}_i^*)$ because of $v_i(g_i^1) \geq v_i(j)$ for all $j \in \mathcal{S}_i^*$. Furthermore, it holds $u_i = v_i(\mathcal{G} \setminus \mathcal{S}_i) \geq v_i(\mathbf{x}_i^* \setminus \mathcal{S}_i) \geq v_i(\mathbf{x}_i^* \setminus \mathcal{S}_i^*)$. We can insert these two inequalities into eq. (6) and prove the theorem thereby:

$$\log \text{NSW}(\mathbf{x}) \geq \frac{1}{\sum_{i \in \mathcal{A}} \eta_i} \cdot \sum_{i \in \mathcal{A}} \eta_i \log \left(\frac{v_i(\mathcal{S}_i^*)}{2n} + \frac{v_i(\mathbf{x}_i^* \setminus \mathcal{S}_i^*)}{n} \right) \quad (9)$$

$$\geq \frac{1}{\sum_{i \in \mathcal{A}} \eta_i} \cdot \sum_{i \in \mathcal{A}} \eta_i \log \left(\frac{v_i(\mathbf{x}_i^*)}{2n} \right) \quad (10)$$

$$= \log \left(\frac{\text{NSW}(\mathbf{x}^*)}{2n} \right) \quad (11)$$

□

The analysis is asymptotically tight. It is possible to design an instance for the asymmetric NSW such that SMatch achieves an approximation ratio approaching $2/n$. It remains to be shown whether the symmetric NSW is equally hard. [4, Section 6.3]

Remark 2. SMatch produces *fair* allocations which are *envy-free up to one item* (EF1). An allocation \mathbf{x} is EF1 if, for every pair $(i_1, i_2) \in \mathcal{A}^2$ of agents, one needs to remove at most one item from the bundle \mathbf{x}_{i_2} of agent i_2 so that agent i_1 does not want to swap bundles. In other words, either it holds $v_{i_1}(\mathbf{x}_{i_1}) \geq v_{i_2}(\mathbf{x}_{i_2})$ or there is an item $j \in \mathbf{x}_{i_2}$ such that $v_{i_1}(\mathbf{x}_{i_1}) \geq v_{i_2}(\mathbf{x}_{i_2} \setminus \{j\})$. [4, Section 5.2]

3 RepReMatch

The algorithm SMatch estimates the valuation of the lowest-value items by determining the set of highest-value items and then valuing the remaining items. Unfortunately, this approach does not work for general submodular valuation functions because taking the set of highest-value items away does not necessarily leave a set of lowest-value items. In fact, it can be shown [6] that determining the set of lowest-value items is approximable only within a factor of $\Omega(\sqrt{m/\ln m})$.

For this reason, the algorithm RepReMatch, described in algorithm 2, relies on an approach with three phases, achieving an approximation factor of $2n(\log_2 n + 3)$ (cf. theorem 2). In phase I, a sufficiently big set of high-value items is determined through repeated matchings. This phase serves merely to determine this set, so items are assigned

Algorithm 2: RepReMatch for the Asymmetric Submodular NSW problem

Input : set \mathcal{G} of m items, set \mathcal{A} of n agents, additive valuation functions

$v_i: \mathcal{P}(\mathcal{G}) \rightarrow \mathbb{R}_0^+$ and weights η_i for all agents $i \in \mathcal{A}$

Output: $\frac{1}{2n(\log_2 n + 3)}$ -approximation $\mathbf{x}^{\text{III}} = (\mathbf{x}_i^{\text{III}})_{i \in \mathcal{A}}$ of an optimal allocation

Phase I:

```

1  $\mathbf{x}_i^{\text{I}} \leftarrow \emptyset$ 
2  $\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G}$ 
3 for  $t \leftarrow 1, \dots, \lceil \log_2 n \rceil + 1$  do
4   if  $\mathcal{G}^{\text{rem}} \neq \emptyset$  then
5      $\mathcal{W} \leftarrow \{ \eta_i \cdot \log(v_i(j)) \mid i \in \mathcal{A}, j \in \mathcal{G}^{\text{rem}} \}$   $\triangleright$  valuation of single item
6      $G \leftarrow (\mathcal{A}, \mathcal{G}^{\text{rem}}, \mathcal{W})$ 
7      $\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$ 
8      $\mathbf{x}_i^{\text{I}} \leftarrow \mathbf{x}_i^{\text{I}} \cup \{j\} \quad \forall (i, j) \in \mathcal{M}$ 
9      $\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G}^{\text{rem}} \setminus \{j \mid (i, j) \in \mathcal{M}\}$ 
10  end if
11 end for
```

Phase II:

```

12  $\mathbf{x}_i^{\text{II}} \leftarrow \emptyset \quad \forall i \in \mathcal{A}$   $\triangleright$  put allocation  $\mathbf{x}^{\text{I}}$  away and start a new one
13 while  $\mathcal{G}^{\text{rem}} \neq \emptyset$  do
14    $\mathcal{W} \leftarrow \{ \eta_i \cdot \log(v_i(\mathbf{x}_i^{\text{II}} \cup \{j\})) \mid i \in \mathcal{A}, j \in \mathcal{G}^{\text{rem}} \}$   $\triangleright$  val. of item & cur. bundle
15    $G \leftarrow (\mathcal{A}, \mathcal{G}^{\text{rem}}, \mathcal{W})$ 
16    $\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$ 
17    $\mathbf{x}_i^{\text{II}} \leftarrow \mathbf{x}_i^{\text{II}} \cup \{j\} \quad \forall (i, j) \in \mathcal{M}$ 
18    $\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G}^{\text{rem}} \setminus \{j \mid (i, j) \in \mathcal{M}\}$ 
19 end while
```

Phase III:

```

20  $\mathcal{G}^{\text{rem}} \leftarrow \bigcup_{i \in \mathcal{A}} \mathbf{x}_i^{\text{I}}$   $\triangleright$  release items assigned in phase I
21  $\mathcal{W} \leftarrow \{ \eta_i \cdot \log(v_i(\mathbf{x}_i^{\text{II}} \cup \{j\})) \mid i \in \mathcal{A}, j \in \mathcal{G}^{\text{rem}} \}$   $\triangleright$  val. of item & cur. bundle
22  $G \leftarrow (\mathcal{A}, \mathcal{G}^{\text{rem}}, \mathcal{W})$ 
23  $\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$ 
24  $\mathbf{x}_i^{\text{III}} \leftarrow \mathbf{x}_i^{\text{II}} \cup \{j\} \quad \forall (i, j) \in \mathcal{M}$ 
25  $\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G}^{\text{rem}} \setminus \{j \mid (i, j) \in \mathcal{M}\}$ 
26  $\mathbf{x}^{\text{III}} \leftarrow \text{arbitrary\_allocation}(\mathcal{A}, \mathcal{G}^{\text{rem}}, \mathbf{x}^{\text{III}}, (v_i)_{i \in \mathcal{A}})$ 
27 return  $\mathbf{x}^{\text{III}}$ 
```

temporarily only. The edge weights reflect this by taking the valuations of just single items into account.

In phase II, the remaining items are assigned normally through repeated matchings. Consequently, each edge weight is updated in each round to be the weighted logarithm of the valuation of both the respective item and the items assigned so far.

In phase III, the high-value items assigned in phase I are released. With the knowledge of items assigned in phase II, one maximum weight matching is calculated, and the matched items are assigned accordingly. Again each edge weight is the weighted logarithm of the valuation of both the respective item and the respective agent's bundle from phase II. The remaining released items are assigned arbitrarily.

We start by analysing phase II as it is the first phase with definitive assignments. To this end, we introduce two types of item sets. Note that we use the term *round* to refer to the iterations of the loops in the phases I and II. For ease of notation, we refer to the moment right before the first iteration in phase II as round 0.

Definition 2. Let \mathbf{x}_i^* be an optimal bundle of some agent $i \in \mathcal{A}$. For any round $r \geq 1$ in phase II, the set $\mathcal{L}_{i,r} \subset \mathbf{x}_i^*$ of *lost* items is the set of all items $j \in \mathbf{x}_i^*$ assigned to other agents $i' \neq i$ in that round.

Definition 3. Let \mathbf{x}_i^* be an optimal bundle of some agent $i \in \mathcal{A}$ and $\mathbf{x}_i^{\text{II}} = \{h_i^1, \dots, h_i^{\tau_i^{\text{II}}}\}$ be her bundle in phase II. The set $\bar{\mathbf{x}}_{i,r}^*$ of *optimal and attainable* items is defined as $\bar{\mathbf{x}}_{i,0}^* := \mathbf{x}_i^* \setminus \bigcup_{i' \in \mathcal{A}} \mathbf{x}_{i'}^{\text{I}}$ in round 0 and as $\bar{\mathbf{x}}_{i,r}^* := \bar{\mathbf{x}}_{i,r-1}^* \setminus (\mathcal{L}_{i,r} \cup \{h_i^{r-1}\})$ in round $r \in [1, \tau_i^{\text{II}}]$.

We denote their sizes by $\ell_{i,r} := |\mathcal{L}_{i,r}|$ and $\bar{\tau}_{i,r}^* := |\bar{\mathbf{x}}_{i,r}^*|$, respectively. First, we give a lower bound on the valuations of optimal and attainable items.

Lemma 3. For each agent $i \in \mathcal{A}$ and her bundle $\mathbf{x}_i^{\text{II}} = \{h_i^1, \dots, h_i^{\tau_i^{\text{II}}}\}$, it holds in all rounds $r = 2, \dots, \tau_i^{\text{II}}$ of phase II that

$$v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^{r-1}) \geq v_i(\bar{\mathbf{x}}_{i,1}^*) - \ell_{i,2} \cdot v_i(h_i^1) - \sum_{r'=2}^{r-1} \ell_{i,r'+1} \cdot v_i(h_i^{r'} \mid h_i^1, \dots, h_i^{r'-1}) - v_i(h_i^1, \dots, h_i^{r-1}).$$

Proof. We prove the lemma by induction on the number r of rounds. In the beginning of the base case $r = 2$, agent i has already been assigned item h_i^1 . For each of the optimal and attainable items $j \in \bar{\mathbf{x}}_{i,1}^*$ in round 1, the marginal valuation $v_i(j \mid \emptyset)$ over the empty set was at most $v_i(h_i^1 \mid \emptyset)$, as otherwise item h_i^1 would not have been assigned first. The marginal valuation $v_i(j \mid h_i^1)$ over $\{h_i^1\}$ is upper-bounded by $v_i(h_i^1 \mid \emptyset)$, too, due to the submodularity of valuations. During round 2, a further $\ell_{i,2}$ of these items j are assigned to other agents, and item h_i^2 is assigned to agent i . We can bound the marginal valuation of the remaining optimal and attainable items in round 2 in the following way:

Case 1— $h_i^1 \in \bar{\mathbf{x}}_{i,1}^*$: It holds $v_i(\bar{\mathbf{x}}_{i,2}^* \mid h_i^1) = v_i(\bar{\mathbf{x}}_{i,2}^* \cup \{h_i^1\}) - v_i(h_i^1) = v_i(\bar{\mathbf{x}}_{i,1}^* \setminus \mathcal{L}_{i,2}) - v_i(h_i^1)$.

Case 2— $h_i^1 \notin \bar{\mathbf{x}}_{i,1}^*$: Due to the monotonicity of the valuation functions, it holds $v_i(\bar{\mathbf{x}}_{i,2}^* \cup \{h_i^1\}) \geq v_i(\bar{\mathbf{x}}_{i,2}^*)$ and, therefore, $v_i(\bar{\mathbf{x}}_{i,2}^* \mid h_i^1) \geq v_i(\bar{\mathbf{x}}_{i,2}^*) - v_i(h_i^1) = v_i(\bar{\mathbf{x}}_{i,1}^* \setminus \mathcal{L}_{i,2}) - v_i(h_i^1)$.

In both cases, the base case is proven because

$$v_i(\bar{\mathbf{x}}_{i,2}^* \mid h_i^1) \geq v_i(\bar{\mathbf{x}}_{i,1}^* \setminus \mathcal{L}_{i,2}) - v_i(h_i^1) \tag{12}$$

$$\geq v_i(\bar{\mathbf{x}}_{i,1}^*) - v_i(\mathcal{L}_{i,2}) - v_i(h_i^1) \tag{13}$$

$$\geq v_i(\bar{\mathbf{x}}_{i,1}^*) - \ell_{i,2} v_i(h_i^1) - v_i(h_i^1), \tag{14}$$

where the second inequality can be shown inductively with the definition of submodularity, and the third inequality is due all $\ell_{i,2}$ items j in set $\mathcal{L}_{i,2}$ not being assigned in round 1 although attainable, implying $v_i(j) \leq v_i(h_i^1)$.

i: Would it be 'dirty' to include notation in the definitions?

Do that or, alternatively, find a paper showing the other def. [IR-fCAwSUF]

For the induction hypothesis, we assume that the lemma holds true for all rounds up to some r . In the induction step $r \rightarrow r+1$, we differentiate the same two cases again:

Case 1— $h_i^r \in \bar{\mathbf{x}}_{i,r}^*$: Again we exploit the submodularity of the valuation functions to

ditto

obtain a lower bound on the marginal valuation of $\bar{\mathbf{x}}_{i,r+1}^*$.

$$v_i(\bar{\mathbf{x}}_{i,r+1}^* \mid h_i^1, \dots, h_i^r) = v_i(\bar{\mathbf{x}}_{i,r+1}^* \cup \{h_i^r\} \mid h_i^1, \dots, h_i^{r-1}) - v_i(h_i^r \mid h_i^1, \dots, h_i^{r-1}) \quad (15)$$

$$= v_i(\bar{\mathbf{x}}_{i,r}^* \setminus \mathcal{L}_{i,r+1} \mid h_i^1, \dots, h_i^{r-1}) - v_i(h_i^r \mid h_i^1, \dots, h_i^{r-1}) \quad (16)$$

$$\geq v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^{r-1}) - v_i(h_i^r \mid h_i^1, \dots, h_i^{r-1}) - v_i(\mathcal{L}_{i,r+1} \mid h_i^1, \dots, h_i^{r-1}) \quad (17)$$

Case 2— $h_i^r \notin \bar{\mathbf{x}}_{i,r}^*$: At first, we use the monotonicity of the valuation functions to get the inequality

$$v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^r) = v_i(\bar{\mathbf{x}}_{i,r}^* \cup \{h_i^1, \dots, h_i^r\}) - v_i(h_i^1, \dots, h_i^r) \quad (18)$$

$$\geq v_i(\bar{\mathbf{x}}_{i,r}^* \cup \{h_i^1, \dots, h_i^{r-1}\}) - v_i(h_i^1, \dots, h_i^r) \quad (19)$$

$$= (v_i(\bar{\mathbf{x}}_{i,r}^* \cup \{h_i^1, \dots, h_i^{r-1}\}) - v_i(h_i^1, \dots, h_i^{r-1})) - (v_i(h_i^1, \dots, h_i^r) - v_i(h_i^1, \dots, h_i^{r-1})) \quad (20)$$

$$= v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^{r-1}) - v_i(h_i^r \mid h_i^1, \dots, h_i^{r-1}). \quad (21)$$

Together with the submodularity of valuation, we obtain the same lower bound again:

$$v_i(\bar{\mathbf{x}}_{i,r+1}^* \mid h_i^1, \dots, h_i^r) = v_i(\bar{\mathbf{x}}_{i,r}^* \setminus \mathcal{L}_{i,r+1} \mid h_i^1, \dots, h_i^r) \quad (22)$$

$$\geq v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^r) - v_i(\mathcal{L}_{i,r+1} \mid h_i^1, \dots, h_i^r) \quad (23)$$

$$\geq v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^r) - v_i(\mathcal{L}_{i,r+1} \mid h_i^1, \dots, h_i^{r-1}) \quad (24)$$

$$\geq v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^{r-1}) - v_i(h_i^r \mid h_i^1, \dots, h_i^{r-1}) - v_i(\mathcal{L}_{i,r+1} \mid h_i^1, \dots, h_i^{r-1}) \quad (25)$$

In both cases, we can replace $v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^{r-1})$ by the induction hypothesis and $v_i(\mathcal{L}_{i,r+1} \mid h_i^1, \dots, h_i^{r-1})$ by $\ell_{i,r+1} \cdot v_i(h_i^r \mid h_i^1, \dots, h_i^{r-1})$ to prove the lemma. For a detailed calculation we refer to Garg, Kulkarni and Kulkarni [4, p. 14]. \square

include
if space
enough

The lemma can be used to find a lower bound on the marginal valuation of the items assigned in each round r .

Corollary 1. *From lemma 3 follows*

$$v_i(h_i^r \mid h_i^1, \dots, h_i^{r-1}) \geq \left(v_i(\bar{\mathbf{x}}_{i,1}^*) - \ell_{i,2} \cdot v_i(h_i^1) - \sum_{r'=2}^{r-1} \ell_{i,r'+1} \cdot v_i(h_i^{r'} \mid h_i^1, \dots, h_i^{r'-1}) - v_i(h_i^1, \dots, h_i^{r-1}) \right) / \bar{\tau}_{i,r}^*.$$

Proof. The valuation functions are monotonic, i.e., $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_2)$ for all item sets $\mathcal{S}_1 \subset \mathcal{S}_2 \subset \mathcal{G}$. Induction shows that there must be an item $j \in \bar{\mathbf{x}}_{i,r}^*$ with a marginal valuation of at least $v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^{r-1}) / \bar{\tau}_{i,r}^*$. As item h_i^r was the one to be assigned, the marginal valuation of it cannot be smaller. Using lemma 3 for the value of $v_i(\bar{\mathbf{x}}_{i,r}^* \mid h_i^1, \dots, h_i^{r-1})$ proves the corollary. \square

change
proof
when intro
is finished

This, finally, enables us to give a lower bound on the valuation of the bundles \mathbf{x}_i^{Π} .

Lemma 4. For each agent $i \in \mathcal{A}$ and her bundle $\mathbf{x}_i^\Pi = \{h_i^1, \dots, h_i^{\tau_i^\Pi}\}$, it holds

$$v_i(h_i^1, \dots, h_i^{\tau_i^\Pi}) \geq v_i(\bar{\mathbf{x}}_{i,1}^*)/n.$$

Proof. In each round $r = 1, \dots, \tau_i^\Pi$, $\ell_{i,r}$ optimal and attainable items of agent i are assigned to other agents. As there are n agents in total, $n-1$ is an upper bound on $\ell_{i,r}$. Furthermore, after τ_i^Π rounds, the number $\bar{\tau}_{i,\tau_i^\Pi}^*$ of optimal and attainable items is at most $n-1 \leq n$ elsewise agent i would have been assigned yet another item. Together with corollary 1, this proves the lemma:

$$v_i(h_i^1, \dots, h_i^{\tau_i^\Pi}) = v_i(h_i^{\tau_i^\Pi} \mid h_i^1, \dots, h_i^{\tau_i^\Pi}) + v_i(h_i^1, \dots, h_i^{\tau_i^\Pi-1}) \quad (26)$$

$$\begin{aligned} &\geq \left(v_i(\bar{\mathbf{x}}_{i,1}^*) - \ell_{i,2} \cdot v_i(h_i^1) - \sum_{r'=2}^{\tau_i^\Pi-1} \ell_{i,r'+1} \cdot v_i(h_i^{r'} \mid h_i^1, \dots, h_i^{r'-1}) \right. \\ &\quad \left. - v_i(h_i^1, \dots, h_i^{\tau_i^\Pi-1}) \right) / \bar{\tau}_{i,r}^* + v_i(h_i^1, \dots, h_i^{\tau_i^\Pi-1}) \end{aligned} \quad (27)$$

$$\begin{aligned} &\geq \left(v_i(\bar{\mathbf{x}}_{i,1}^*) - (n-1)v_i(h_i^1) - \sum_{r'=2}^{\tau_i^\Pi-1} (n-1)v_i(h_i^{r'} \mid h_i^1, \dots, h_i^{r'-1}) \right. \\ &\quad \left. - v_i(h_i^1, \dots, h_i^{\tau_i^\Pi-1}) \right) / n + v_i(h_i^1, \dots, h_i^{\tau_i^\Pi-1}) \end{aligned} \quad (28)$$

$$\begin{aligned} &\geq (v_i(\bar{\mathbf{x}}_{i,1}^*) - (n-1)v_i(h_i^1, \dots, h_i^{\tau_i^\Pi-1}) - v_i(h_i^1, \dots, h_i^{\tau_i^\Pi-1})) / n \\ &\quad + v_i(h_i^1, \dots, h_i^{\tau_i^\Pi-1}) \end{aligned} \quad (29)$$

$$= v_i(\bar{\mathbf{x}}_{i,1}^*)/n \quad (30)$$

□

After having obtained a lower bound on the valuation of items assigned in phase II, we need a lower bound for phase III as well. Therefore we introduce a third type of item set.

Definition 4. Let $\mathbf{x}_i^* = \{g_i^1, \dots, h_i^{\tau_i^*}\}$ be an optimal bundle of some agent $i \in \mathcal{A}$. The set \mathcal{G}_i^+ of *overly good* items is defined as $\mathcal{G}_i^+ := \{j \in \mathcal{G} \mid v_i(j) \geq v_i(g_i^1)\}$.

Lemma 5. In phase III, there exists a matching such that each agent $i \in \mathcal{A}$ is matched to one of her overly good items in the set $\bigcup_{i' \in \mathcal{A}} \mathbf{x}_{i'}^I$ of released items.

Proof. If all items were matched in phase I, i.e., $\bigcup_{i' \in \mathcal{A}} \mathbf{x}_{i'}^I = \mathcal{G}$, then all optimal items are released in phase III and each agent can be matched to one; the lemma is proven immediately. If not, imagine for some t that only the items assigned in the first t rounds of phase I were released. Now choose some matching \mathcal{M}_t with the following properties:

1. If for an agent i all overly good items were amongst the released items, she gets matched with an overly good item $j \in \mathcal{G}_i^+$.
2. The number of agents matched with one of their overly good items is maximal amongst all matchings fulfilling property 1.

Property 1 is always satisfiable as each set \mathcal{G}_i^+ is the only one to contain the item g_i^1 , which can be matched with agent i . Property 2 leads to all agents being matched with an overly good item for $t = \lceil \log_2 n \rceil + 1$, i.e. the number of rounds in phase I, whence the lemma follows. To prove this, we denote by \mathcal{A}_t^- the set of agents who are *not* matched with one of their overly good items, and show by induction on t that it holds $|\mathcal{A}_t^-| \leq n/2^t$.

In the base case $t = 1$, none of the items are assigned initially. Denote by α the number of agents who were not assigned an overly good item in the first round of phase I. If $\alpha \leq n/2$, then a matching \mathcal{M}_1 obviously exists and the base case is immediately proven. Otherwise,

wrong!
the union
contains it
though

all items from at least α many sets \mathcal{G}_i^+ got assigned to someone. Again: Each set \mathcal{G}_i^+ is the only one containing the item g_i^1 , so the union of these sets contains at least α items which can be matched with at least α agents upon release. This then leaves at most $n - \alpha < n/2$ agents not matched with an overly good item.

For the induction hypothesis, we assume that the statement holds true for all rounds up to some t . In the induction step $t \rightarrow t + 1$, by property 1, there is at least one unassigned item in each set $\mathcal{G}_{i'}^+$ for all agents $i' \in \mathcal{A}_t^-$ at the start of round $t + 1$. Analogously to the base case, for at least half of those agents i' these unassigned items will be assigned to them or someone else and it can be argued accordingly. By the induction hypothesis, it holds $|\mathcal{A}_{t+1}^-| \leq |\mathcal{A}_t^-|/2 \leq (n/2^t)/2 = n/2^{t+1}$. \square

This allows us to calculate an approximation factor for RepReMatch by comparing its output with an optimal allocation \mathbf{x}^* .

Theorem 2. *RepReMatch has an approximation factor of $2n(\log_2 n + 3)$.*

Proof. By lemma 5, we can assign each agent i an overly good item $j_i^+ \in \mathcal{G}_i^+$ in the beginning of phase III. RepReMatch maximises the logarithmic Nash social welfare, so

$$\log \text{NSW}(\mathbf{x}^{\text{III}}) \geq \frac{1}{\sum_{i \in \mathcal{A}} \eta_i} \cdot \sum_{i \in \mathcal{A}} \eta_i \log v_i(j_i^+, h_i^1, \dots, h_i^{\tau_i^{\text{II}}}) \quad (31)$$

is a lower bound on the logarithmic NSW after the first matching in phase III, whereby $\mathbf{x}_i^{\text{II}} = \{h_i^1, \dots, h_i^{\tau_i^{\text{II}}}\}$ is the bundle of agent i from phase II.

Item j_i^+ was released in phase III, which means it was assigned in phase I, implying $j_i^+ \in \mathbf{x}_i^* \setminus \bar{\mathbf{x}}_{i,0}^*$ and, subsequently, $j_i^+ \in (\mathbf{x}_i^* \setminus \bar{\mathbf{x}}_{i,0}^*) \cup \mathcal{L}_{i,1}$. Phase I runs for at most $\lceil \log_2 n \rceil + 1$ rounds, and at most n items are assigned in each iteration. Therefore, at most $n(\log_2 n + 2)$ optimal items are assigned in that phase, i. e., $|\mathbf{x}_i^* \setminus \bar{\mathbf{x}}_{i,0}^*| \leq n(\log_2 n + 2)$. Furthermore, it holds $n \geq \ell_{i,1} = |\mathcal{L}_{i,1}|$ as in lemma 4. Together with the monotonicity of the valuation functions, this yields

$$v_i(j_i^+, h_i^1, \dots, h_i^{\tau_i^{\text{II}}}) \geq v_i(j_i^+) \geq \frac{v_i((\mathbf{x}_i^* \setminus \bar{\mathbf{x}}_{i,0}^*) \cup \mathcal{L}_{i,1})}{n(\log_2 n + 3)} \quad (32)$$

as lower bound on the valuations of bundles. Moreover, lemma 2 and the monotonicity of the valuation functions yield

$$v_i(j_i^+, h_i^1, \dots, h_i^{\tau_i^{\text{II}}}) \geq v_i(h_i^1, \dots, h_i^{\tau_i^{\text{II}}}) \geq \frac{v_i(\bar{\mathbf{x}}_{i,1}^*)}{n} \geq \frac{v_i(\bar{\mathbf{x}}_{i,1}^*)}{n(\log_2 n + 3)} = \frac{v_i(\bar{\mathbf{x}}_{i,0}^* \setminus \mathcal{L}_{i,1})}{n(\log_2 n + 3)} \quad (33)$$

as yet another lower bound. The mean of eqs. (32) to (33) and the monotonicity of the valuation functions give the concise lower bound

$$v_i(j_i^+, h_i^1, \dots, h_i^{\tau_i^{\text{II}}}) \geq \frac{1}{2} \left(\frac{v_i((\mathbf{x}_i^* \setminus \bar{\mathbf{x}}_{i,0}^*) \cup \mathcal{L}_{i,1})}{n(\log_2 n + 3)} + \frac{v_i(\bar{\mathbf{x}}_{i,0}^* \setminus \mathcal{L}_{i,1})}{n(\log_2 n + 3)} \right) \quad (34)$$

$$\geq \frac{1}{2} \cdot \frac{v_i(((\mathbf{x}_i^* \setminus \bar{\mathbf{x}}_{i,0}^*) \cup \mathcal{L}_{i,1}) \cup (\bar{\mathbf{x}}_{i,0}^* \setminus \mathcal{L}_{i,1}))}{n(\log_2 n + 3)} \quad (35)$$

$$= \frac{v_i(\mathbf{x}_i^*)}{2n(\log_2 n + 3)}. \quad (36)$$

We can insert this lower bound into eq. (31) and prove the theorem thereby:

$$\log \text{NSW}(\mathbf{x}^{\text{III}}) \geq \frac{1}{\sum_{i \in \mathcal{A}} \eta_i} \cdot \sum_{i \in \mathcal{A}} \eta_i \log \left(\frac{v_i(\mathbf{x}_i^*)}{2n(\log_2 n + 3)} \right) = \log \left(\frac{\text{NSW}(\mathbf{x}^*)}{2n(\log_2 n + 3)} \right) \quad (37)$$

\square

i: Error in paper? see also lemma 5

4 Hardness of Approximation

Garg, Kulkarni and Kulkarni [4, Section 4] provide the following hardness result.

Theorem 3. *The submodular NSW is not approximable within a factor of $\frac{e}{e-1}$ in polynomial time unless $\mathcal{P} = \mathcal{NP}$.*

i: Should it not be $\frac{e-1}{e} - \epsilon$?

Proof. Consider the related USW problem¹.

Problem 2. Given a set \mathcal{G} of indivisible items and a set \mathcal{A} of agents with monotonic, submodular valuation functions $v_i: \mathcal{P}(\mathcal{G}) \rightarrow \mathbb{R}$ for all agents $i \in \mathcal{A}$, the *symmetric submodular Utilitarian Social Welfare problem* (USW) is to find an allocation maximising the sum of valuations, that is

$$\arg \max_{\mathbf{x} \in \mathbf{X}_n(\mathcal{G})} \left\{ \sum_{i \in \mathcal{A}} v_i(\mathbf{x}_i) \right\}$$

where $\mathbf{X}_n(\mathcal{G})$ is the set of all possible allocations of the items in \mathcal{G} amongst n agents.

Note that this problem is identical to the symmetric submodular NSW except for the sum in the target function instead of a product. We will exploit this fact to calculate the NSW of instances for USW. Khot et al. [5] supply a polynomial-time reduction of USW from the following problem:

Problem 3. Given a graph $G = (V, E)$ and a constant $c \leq 1$, the *c-Gap-Max-3-Colouring problem* is to decide whether, for any 3-colouring of graph G which maximises the number of edges with different coloured endpoints, the number of such edges is $|E|$ (*Yes instance*) or $c|E|$ and below (*No instance*).

Proposition 1. *There exists a constant $c \leq 1$ such that the c-Gap-Max-3-Colouring problem is \mathcal{NP} -hard.*

Reducing an instance of the c-Gap-Max-3-Colouring problem yields an instance of the symmetric submodular USW problem with identical valuation functions. Its properties are as follows:

Yes instance The USW is nC because every agent values her bundle at n , whereby C is a constant depending on the input graph. The NSW of the instance would be C .

No instance The USW is $\frac{e-1}{e}nC$. Applying the inequality of arithmetic and geometric means, i. e., $(x_1 + \dots + x_n)/n \geq \sqrt[n]{x_1 \dots x_n}$ for all nonnegative numbers $x_1, \dots, x_n \in \mathbb{R}_0^+$, reveals that the NSW of the instance is at most $\frac{e-1}{e}C$.

Thereout follows that the submodular NSW problem, even when symmetric and with identical valuation functions, cannot be approximated within a factor better than $\frac{e}{e-1}$; otherwise one could decide the c-Gap-Max-3-Colouring problem in polynomial time by checking whether the corresponding NSW instance has a value above $\frac{e-1}{e}C$. \square

For a constant number of agents, Garg, Kulkarni and Kulkarni [4, Section 5.1] describe a family $(A_\epsilon)_{\epsilon > 0}$ of algorithms for the asymmetric submodular NSW problem where each algorithm A_ϵ achieves an approximation factor of $\frac{e}{e-1} + \epsilon$.

¹Garg, Kulkarni and Kulkarni (and many others) call problem 2 the ‘Allocation problem’. We changed the name to match the naming scheme of the NSW problem and to avoid confusion, as both problems are about allocations.

5 conclusion

- Of course a short rehearsal of the results for the now knowledgeable reader.
- An outlook would be nice to have. Its content would mostly depend on what recent research has not yet answered.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

References

- [1] Tayebbeh Bahreini, Hossein Badri and Daniel Grosu. ‘An Envy-Free Auction Mechanism for Resource Allocation in Edge Computing Systems’. In: *SEC 2018*. 2018 IEEE/ACM Symposium on Edge Computing (Seattle, WA, USA, 25th–27th Oct. 2018). Ed. by Patrick Kellenberger. The Institute of Electrical and Electronics Engineers, Inc., pp. 313–322. DOI: [10.1109/SEC.2018.00030](https://doi.org/10.1109/SEC.2018.00030). URL: https://www.researchgate.net/publication/329561811_An_Envy-Free_Auction_Mechanism_for_Resource_Allocation_in_Edge_Computing_Systems (visited on 08/07/2023).
- [2] Yann Chevaleyre et al. ‘Issues in Multiagent Resource Allocation’. In: *Informatica* 30.1 (Jan. 2006), pp. 3–31. URL: <https://www.informatica.si/index.php/informatica/article/view/70/62> (visited on 07/06/2023).
- [3] Schahram Dustdar, ed. *NSF Workshop Report on Grand Challenges in Edge Computing*. NSF Edge Computing Workshop (Washington, DC, USA, 26th Oct. 2016). National Science Foundation. 15th Nov. 2017. 17 pp. URL: https://dsg.tuwien.ac.at/team/sd/papers/Bericht_NSF_S_Dustdar.pdf (visited on 08/07/2023).
- [4] Jugal Garg, Pooja Kulkarni and Rucha Kulkarni. *Approximating Nash Social Welfare under Submodular Valuations through (Un)Matchings*. 28th Dec. 2019. arXiv: [1912.12541v1](https://arxiv.org/abs/1912.12541) [cs.GT].
- [5] Subhash Khot et al. ‘Inapproximability Results for Combinatorial Auctions with Submodular Utility Functions’. In: *Algorithmica* 52 (1 13th Oct. 2007), pp. 3–18. ISSN: 0178-4617. DOI: [10.1007/s00453-007-9105-7](https://doi.org/10.1007/s00453-007-9105-7) (visited on 05/07/2023).
- [6] Zoya Svitkina and Lisa Fleischer. *Submodular approximation: sampling-based algorithms and lower bounds*. 2010. arXiv: [0805.1071](https://arxiv.org/abs/0805.1071) [cs.DS] (visited on 01/07/2023).