

## Seminar Approximation Algorithms

# **ANSWuSVp(U)M**

Zeno Adrian Weil

21st May 2023

Examiner: Prof Dr Martin Hoefer  
Supervisor: Dr Giovanna Varricchio

## Abstract

Lorem ipsum dolor sit amet, consectetur 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, consectetur 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, consectetur 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, consectetur 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**

or rather ‘allocated’?	4
less strict def of valuations; restriction for our case later on	4
agents without items assigned have valuation zero → prevent	4
definition of approximation factor [def environment or in-text?]	4
might delete if document in two columns → unused edges in grey	6
What is the factor?	6
stimmt das?	6
decide on $x$ or $x^{\text{III}}$ ; discrepancies in orig paper in def of $x^{\text{III}}$ !	9

# 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  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathcal{P}(\mathcal{G})^n$  such that each item is element of exactly one  $\mathbf{x}_i$ , 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.

⋮

**Definition 2.** Given a set  $\mathcal{G}$  of items and a set  $\mathcal{A}$  of agents with *valuations*  $v_i: \mathcal{P}(\mathcal{G}) \rightarrow \mathbb{R}$  and *agent weights*  $\eta_i$  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

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

where  $\Pi_n(\mathcal{G})$  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  $\eta_i$  are equal, and *asymmetric* otherwise.

agents without items assigned have valuation zero  $\rightarrow$  prevent

⋮

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

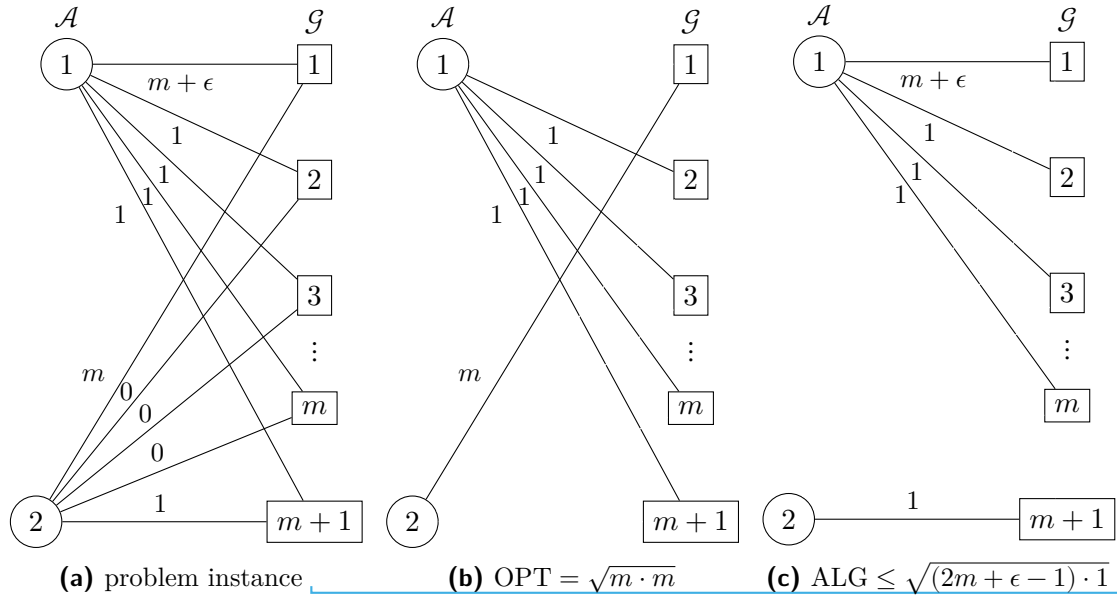
We use *additive NSW* and *submodular NSW* as shorthands for the Nash social welfare problems with additive and submodular valuation functions, respectively.

Lorem ipsum dolor sit amet, consectetur 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.

or rather  
'alloc-  
ated'?

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

Praesent in sapien. Lorem ipsum dolor sit amet, consectetur 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, consectetur 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, consectetur 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, consectetur 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, consectetur 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.



**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.

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

## 2 SMatch

In the case of an equal number of agents and items, i.e.  $|\mathcal{A}| = n = m = |\mathcal{G}|$ , 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 to just repeatedly find a maximum matching and assign the items accordingly until all items are allocated. 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 similar valuations, wherefore it may be beneficial for an agent to leave items to other agents if those agents cannot expect many more valuable items in the future.

What is stimmmt das?

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  $\frac{1}{2n}$  (cf. theorem 1 later on). 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  $\mathcal{G}_i^j$ . To obtain a well-defined order, items of equal rank are further ordered numerically. SMatch, too, does 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

---

**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  $v_i: \mathcal{P}(\mathcal{G}) \rightarrow \mathbb{R}_0^+$  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  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$  of an optimal allocation

```

1  $\mathbf{x}_i \leftarrow \emptyset \quad \forall i \in \mathcal{A}$ 
2  $u_i \leftarrow v_i(\mathcal{G}_i^{2n+1}, \dots, \mathcal{G}_i^m) \quad \forall i \in \mathcal{A}$ 
3  $\mathcal{W} \leftarrow \{\eta_i \cdot \log(v_i(j) + u_i/n) \mid i \in \mathcal{A}, j \in \mathcal{G}\} \quad \triangleright \text{edge weights}$ 
4  $G \leftarrow (\mathcal{A}, \mathcal{G}, \mathcal{W}) \quad \triangleright \text{bipartite graph}$ 
5  $\mathcal{M} \leftarrow \text{max\_weight\_matching}(G)$ 
6  $\mathbf{x}_i \leftarrow \{j \mid (i, j) \in \mathcal{M}\} \quad \forall i \in \mathcal{A} \quad \triangleright \text{allocate according to matching}$ 
7  $\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G} \setminus \{j \mid (i, j) \in \mathcal{M}\} \quad \triangleright \text{remove allocated goods}$ 
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  $\mathbf{x}$ 

```

---

and is defined as

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

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 set of items assigned to agent  $i$  in previous matchings. The addend  $v_i(\mathbf{x}_i)$  could lead to better allocations in applications, but does not improve the approximation factor asymptotically.

For convenience, we order the items in the final allocation  $\mathbf{x}_i = \{h_i^1, \dots, h_i^{\tau_i}\}$  of agent  $i$  by the order in which they were assigned, so that item  $h_i^t$  is assigned after the  $t$ -th matching. To calculate the approximation factor of SMatch, we first need to establish a lower bound on the valuation of single items.

**Lemma 1.** *For each agent  $i$  and their final allocation  $\mathbf{x}_i = \{h_i^1, \dots, h_i^{\tau_i}\}$ , the item  $h_i^t$  is worth at least as much as the overall  $tn$ -th most highly valued item  $\mathcal{G}_i^{tn}$ , i. e.  $v_i(h_i^t) \geq v_i(\mathcal{G}_i^{tn})$ , for all  $t = 1, \dots, \tau_i$ .*

*Proof.* At the start of the  $t$ -th iteration, 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 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  $k \leq tn$  by definition of  $\mathcal{G}_i^k$ , 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.**  $v_i(h_i^2, \dots, h_i^{\tau_i}) \geq u_i/n$ .

*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  $k \in \{0, \dots, n-1\}$  and, consequently, its valuation  $v_i(h_i^t)$  is at least as high as the mean valuation  $\frac{1}{n}v_i(\mathcal{G}_i^{tn}, \dots, \mathcal{G}_i^{tn+n-1})$ . Further, it holds  $\tau_i n + n \geq m$  since each agent receives items for at least  $\lfloor \frac{m}{n} \rfloor \geq \frac{m}{n} - 1$  rounds. Together, this yields

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

$$\geq \frac{1}{n} v_i(\mathcal{G}_i^{2n}, \dots, \mathcal{G}_i^{m-1}) \geq \frac{1}{n} v_i(\mathcal{G}_i^{2n+1}, \dots, \mathcal{G}_i^m) = \frac{u_i}{n} \quad (3)$$

with the last inequality stemming from  $v_i(\mathcal{G}_i^{2n}) \geq v_i(\mathcal{G}_i^m)$ .  $\square$

This allows us to calculate a lower bound on the approximation factor of SMatch.

**Theorem 1.** *SMatch has an approximation factor of  $\frac{1}{2n}$ .*

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

$$\log \text{NSW}(\mathbf{x}) = \frac{1}{\sum_{i=1}^n \eta_i} \cdot \sum_{i=1}^n \eta_i \log v_i(h_i^1, \dots, h_i^{\tau_i}) \quad (4)$$

$$= \frac{1}{\sum_{i=1}^n \eta_i} \cdot \sum_{i=1}^n \eta_i \log(v_i(h_i^1) + v_i(h_i^2, \dots, h_i^{\tau_i})) \quad (5)$$

$$\geq \frac{1}{\sum_{i=1}^n \eta_i} \cdot \sum_{i=1}^n \eta_i \log(v_i(h_i^1) + u_i/n) \quad (6)$$

Notice that the first matching of SMatch maximises the sum in eq. (6). Thus, assigning all agents  $i$  their respective most highly valued item  $g_i^1$  in an optimal allocation  $\mathbf{x}_i^* = \{g_i^1, \dots, g_i^{\tau_i}\}$  yields the even lower bound

$$\log \text{NSW}(\mathbf{x}) \geq \frac{1}{\sum_{i=1}^n \eta_i} \cdot \sum_{i=1}^n \eta_i \log(v_i(g_i^1) + u_i/n). \quad (7)$$

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

$$u_i = \min_{\substack{\mathcal{S} \subset \mathcal{G} \\ |\mathcal{S}| \leq 2n}} \{v_i(\mathcal{G} \setminus \mathcal{S})\} \quad \text{or, alternatively,} \quad u_i = v_i(\mathcal{G} \setminus \mathcal{S}) \quad \text{with } \mathcal{S} := \arg \min_{\substack{\mathcal{S} \subset \mathcal{G} \\ |\mathcal{S}| \leq 2n}} \{v_i(\mathcal{G} \setminus \mathcal{S})\} \quad (8)$$

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

$$\mathcal{S}^* := \arg \min_{\substack{\mathcal{S} \subset \mathcal{G} \\ |\mathcal{S}| \leq 2n}} \{v_i(\mathbf{x}_i^* \setminus \mathcal{S})\}. \quad (9)$$

We get the lower bound  $v_i(g_i^1) \geq \frac{1}{2n} v_i(\mathcal{S}^*)$  from a similar argument as in the proof of lemma 2. Further, it holds  $u_i = v_i(\mathcal{G} \setminus \mathcal{S}) \geq v_i(\mathbf{x}_i^* \setminus \mathcal{S}) \geq v_i(\mathbf{x}_i^* \setminus \mathcal{S}^*)$ . We can substitute these into eq. (7) and prove the theorem thereby:

$$\log \text{NSW}(\mathbf{x}) \geq \frac{1}{\sum_{i=1}^n \eta_i} \cdot \sum_{i=1}^n \eta_i \log \left( \frac{v_i(\mathcal{S}^*)}{2n} + \frac{v_i(\mathbf{x}_i^* \setminus \mathcal{S}^*)}{n} \right) \quad (10)$$

$$\geq \frac{1}{\sum_{i=1}^n \eta_i} \cdot \sum_{i=1}^n \eta_i \log \left( \frac{v_i(\mathbf{x}_i^*)}{2n} \right) = \log \left( \frac{\text{OPT}}{2n} \right) \quad (11)$$

$\square$



---

**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  $v_i: \mathcal{P}(\mathcal{G}) \rightarrow \mathbb{R}_0^+$  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  $\mathbf{x}^{\text{III}} = (\mathbf{x}_1^{\text{III}}, \dots, \mathbf{x}_n^{\text{III}})$  of an optimal allocation

*Phase I:*

```

1  $\mathbf{x}_i^{\text{I}} \leftarrow \emptyset \quad \forall i \in \mathcal{A}$ 
2  $\mathcal{G}^{\text{rem}} \leftarrow \mathcal{G}$ 
3 for  $t = 0, \dots, \lceil \log n \rceil - 1$  do
4   if  $\mathcal{G}^{\text{rem}} \neq \emptyset$  then
5      $\mathcal{W} \leftarrow \{ \eta_i \cdot \log(v_i(j)) \mid i \in \mathcal{A}, j \in \mathcal{G} \}$ 
6      $G \leftarrow (\mathcal{A}, \mathcal{G}, \mathcal{W})$ 
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  $\mathbf{x}_i^{\text{II}} \leftarrow \emptyset \quad \forall i \in \mathcal{A}$ 
13 while  $\mathcal{G}^{\text{rem}} \neq \emptyset$  do
14    $\mathcal{W} \leftarrow \{ \eta_i \cdot \log(v_i(\mathbf{x}_i^{\text{II}} \cup \{j\})) \mid i \in \mathcal{A}, j \in \mathcal{G} \}$ 
15    $G \leftarrow (\mathcal{A}, \mathcal{G}, \mathcal{W})$ 
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  $\mathcal{G}^{\text{rem}} \leftarrow \bigcup_{i \in \mathcal{A}} \mathbf{x}_i^{\text{I}} \quad \triangleright \text{release items allocated in first phase}$ 
21  $\mathcal{W} \leftarrow \{ \eta_i \cdot \log(v_i(\mathbf{x}_i^{\text{II}} \cup \{j\})) \mid i \in \mathcal{A}, j \in \mathcal{G} \}$ 
22  $G \leftarrow (\mathcal{A}, \mathcal{G}, \mathcal{W})$ 
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  $\mathbf{x}_i^{\text{III}} \leftarrow \text{arbitrary\_allocation}(\mathcal{A}, \mathcal{G}^{\text{rem}}, \mathbf{x}_i^{\text{III}}, (v_i)_{i \in \mathcal{A}})$ 
27 return  $\mathbf{x}^{\text{III}}$ 

```

---

decide on  $\mathbf{x}$  or  $\mathbf{x}^{\text{III}}$ ; discrepancies in orig paper in def of  $\mathbf{x}^{\text{III}}$ !