

## Seminar Approximation Algorithms

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

Based on a paper of the same title by J. Garg, P. Kulkarni, and R. Kulkarni

Zeno Adrian Weil

Supervised by Dr Giovanna Varricchio

30th July 2023 · Algorithms and Complexity (Prof. Dr Martin Hoefer)

# What is the issue?



# What is the issue?

We need to distribute goods amongst recipients



# What is the issue?

We need to distribute goods amongst recipients *efficiently*



# What is the issue?

We need to distribute goods amongst recipients *efficiently* and *fairly*.



# What is the issue?

We need to distribute goods amongst recipients *efficiently* and *fairly*.

Where is this encountered?

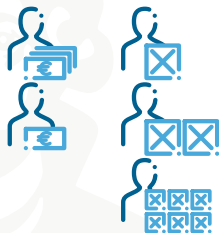


# What is the issue?

We need to distribute goods amongst recipients *efficiently* and *fairly*.

Where is this encountered?

- procurement

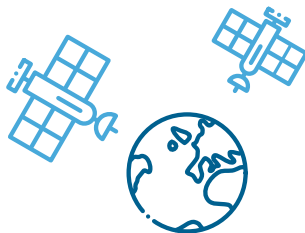
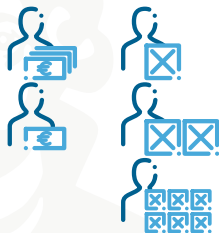


# What is the issue?

We need to distribute goods amongst recipients *efficiently* and *fairly*.

Where is this encountered?

- procurement
- satellites



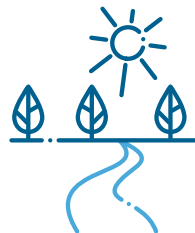
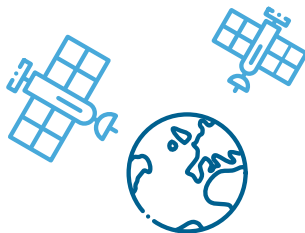
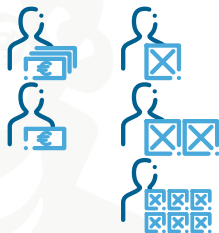


# What is the issue?

We need to distribute goods amongst recipients *efficiently* and *fairly*.

Where is this encountered?

- procurement
- satellites
- water withdrawal



# Table of Contents

## 1 Preliminaries

- Allocations
- Valuation Functions
- Maximum Nash Social Welfare Problem

## 2 RepReMatch

- Naïve Approach
- The Algorithm
- Analysing Phases I & III
- Analysing Phase II

## 3 Conclusion

# 1

## Preliminaries



# Preliminaries

## Allocations



Setting:



Setting:

- recipients: set  $\mathcal{A}$  of  $n$  agents



# Allocations

Setting:

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



Setting:

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

## Definition

An *allocation* is a tuple  $x = (x_i)_{i \in \mathcal{A}}$  of bundles  $x_i \subset \mathcal{G}$



Setting:

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

## Definition

An **allocation** is a tuple  $x = (x_i)_{i \in \mathcal{A}}$  of bundles  $x_i \subset \mathcal{G}$  such that each item is element of precisely one bundle.

Setting:

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

### Definition

An **allocation** is a tuple  $x = (x_i)_{i \in \mathcal{A}}$  of bundles  $x_i \subset \mathcal{G}$  such that each item is element of precisely one bundle.

Item  $j$  is **assigned** to agent  $i$  if  $j \in x_i$ .

Setting:

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

### Definition

An **allocation** is a tuple  $x = (x_i)_{i \in \mathcal{A}}$  of bundles  $x_i \subset \mathcal{G}$  such that each item is element of precisely one bundle.

Item  $j$  is **assigned** to agent  $i$  if  $j \in x_i$ .

But how to measure its efficiency and fairness?

# Valuation Functions



# Valuation Functions

Requirements:



# Valuation Functions

Requirements:

- **monotonically non-decreasing:**  $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_2)$  if  $\mathcal{S}_1 \subset \mathcal{S}_2$



# Valuation Functions

Requirements:

- **monotonically non-decreasing:**  $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_2)$  if  $\mathcal{S}_1 \subset \mathcal{S}_2$
- **normalised:**  $v_i(\emptyset) = 0$



# Valuation Functions

Requirements:

- **monotonically non-decreasing:**  $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_2)$  if  $\mathcal{S}_1 \subset \mathcal{S}_2$
- **normalised:**  $v_i(\emptyset) = 0$

Types:





# Valuation Functions

Requirements:

- **monotonically non-decreasing:**  $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_2)$  if  $\mathcal{S}_1 \subset \mathcal{S}_2$
- **normalised:**  $v_i(\emptyset) = 0$

Types:

- **additive:**  $v_i(\mathcal{S}) := \sum_{j \in \mathcal{S}} v_i(j)$



# Valuation Functions

Requirements:

- **monotonically non-decreasing:**  $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_2)$  if  $\mathcal{S}_1 \subset \mathcal{S}_2$
- **normalised:**  $v_i(\emptyset) = 0$

Types:

- **additive:**  $v_i(\mathcal{S}) := \sum_{j \in \mathcal{S}} v_i(j)$
- **submodular:**  $v_i(\mathcal{S}_1 \mid \mathcal{S}_2)$



# Valuation Functions

Requirements:

- **monotonically non-decreasing:**  $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_2)$  if  $\mathcal{S}_1 \subset \mathcal{S}_2$
- **normalised:**  $v_i(\emptyset) = 0$

Types:

- **additive:**  $v_i(\mathcal{S}) := \sum_{j \in \mathcal{S}} v_i(j)$
- **submodular:**  $v_i(\mathcal{S}_1 \mid \mathcal{S}_2) := v_i(\mathcal{S}_1 \cup \mathcal{S}_2) - v_i(\mathcal{S}_2)$



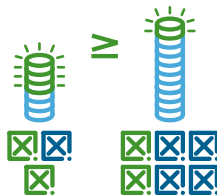
# Valuation Functions

Requirements:

- **monotonically non-decreasing:**  $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_2)$  if  $\mathcal{S}_1 \subset \mathcal{S}_2$
- **normalised:**  $v_i(\emptyset) = 0$

Types:

- **additive:**  $v_i(\mathcal{S}) := \sum_{j \in \mathcal{S}} v_i(j)$
- **submodular:**  $v_i(\mathcal{S}_1 \mid \mathcal{S}_2) := v_i(\mathcal{S}_1 \cup \mathcal{S}_2) - v_i(\mathcal{S}_2)$ 
  - diminishing returns



# Asymmetric Maximum Nash Social Welfare Problem



# Asymmetric Maximum Nash Social Welfare Problem

## Problem

$$\mathbf{x}^* \stackrel{!}{=} \arg \max_{\mathbf{x} \in X_{\mathcal{A}}(\mathcal{G})} \{\text{NSW}(\mathbf{x})\}$$

- $X_{\mathcal{A}}(\mathcal{G})$ : all possible allocations

# Asymmetric Maximum Nash Social Welfare Problem

## Problem

$$\mathbf{x}^* \stackrel{!}{=} \arg \max_{\mathbf{x} \in X_{\mathcal{A}}(\mathcal{G})} \{\text{NSW}(\mathbf{x})\}$$

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

# Asymmetric Maximum Nash Social Welfare Problem

## Problem

$$\mathbf{x}^* \stackrel{!}{=} \arg \max_{\mathbf{x} \in X_{\mathcal{A}}(\mathcal{G})} \{\text{NSW}(\mathbf{x})\} \quad \text{with} \quad \text{NSW}(\mathbf{x}) := \left( \prod_{i \in \mathcal{A}} v_i(\mathbf{x}_i)^{\eta_i} \right)^{1 / \sum_{i \in \mathcal{A}} \eta_i}$$

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



# Asymmetric Maximum Nash Social Welfare Problem

## Problem

$$\mathbf{x}^* \stackrel{!}{=} \arg \max_{\mathbf{x} \in X_{\mathcal{A}}(\mathcal{G})} \{\text{NSW}(\mathbf{x})\} \quad \text{with} \quad \text{NSW}(\mathbf{x}) := \left( \prod_{i \in \mathcal{A}} v_i(\mathbf{x}_i)^{\eta_i} \right)^{1 / \sum_{i \in \mathcal{A}} \eta_i}$$

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

The NSW strikes a middle ground between efficiency and fairness!

# Asymmetric Maximum Nash Social Welfare Problem

## Problem

$$\mathbf{x}^* \stackrel{!}{=} \arg \max_{\mathbf{x} \in X_{\mathcal{A}}(\mathcal{G})} \{\text{NSW}(\mathbf{x})\} \quad \text{with} \quad \text{NSW}(\mathbf{x}) := \left( \prod_{i \in \mathcal{A}} v_i(\mathbf{x}_i)^{\eta_i} \right)^{1 / \sum_{i \in \mathcal{A}} \eta_i}$$

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

The NSW strikes a middle ground between efficiency and fairness!

Is there a polynomial-time algorithm with an approximation factor ...

# Asymmetric Maximum Nash Social Welfare Problem

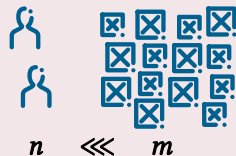
## Problem

$$\mathbf{x}^* \stackrel{!}{=} \arg \max_{\mathbf{x} \in X_{\mathcal{A}}(\mathcal{G})} \{\text{NSW}(\mathbf{x})\} \quad \text{with} \quad \text{NSW}(\mathbf{x}) := \left( \prod_{i \in \mathcal{A}} v_i(\mathbf{x}_i)^{\eta_i} \right)^{1 / \sum_{i \in \mathcal{A}} \eta_i}$$

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

The NSW strikes a middle ground between efficiency and fairness!

Is there a polynomial-time algorithm with an approximation factor ...



# Asymmetric Maximum Nash Social Welfare Problem

## Problem

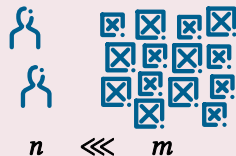
$$\mathbf{x}^* \stackrel{!}{=} \arg \max_{\mathbf{x} \in X_{\mathcal{A}}(\mathcal{G})} \{\text{NSW}(\mathbf{x})\} \quad \text{with} \quad \text{NSW}(\mathbf{x}) := \left( \prod_{i \in \mathcal{A}} v_i(\mathbf{x}_i)^{\eta_i} \right)^{1 / \sum_{i \in \mathcal{A}} \eta_i}$$

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

The NSW strikes a middle ground between efficiency and fairness!

Is there a polynomial-time algorithm with an approximation factor ...

- ... dependent on  $n$ ?



# Asymmetric Maximum Nash Social Welfare Problem

## Problem

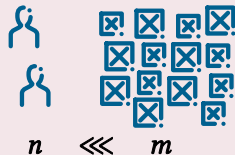
$$\mathbf{x}^* \stackrel{!}{=} \arg \max_{\mathbf{x} \in X_{\mathcal{A}}(\mathcal{G})} \{\text{NSW}(\mathbf{x})\} \quad \text{with} \quad \text{NSW}(\mathbf{x}) := \left( \prod_{i \in \mathcal{A}} v_i(\mathbf{x}_i)^{\eta_i} \right)^{1 / \sum_{i \in \mathcal{A}} \eta_i}$$

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

The NSW strikes a middle ground between efficiency and fairness!

Is there a polynomial-time algorithm with an approximation factor ...

- ... dependent on  $n$ ?
- ... independent from  $m$ ?



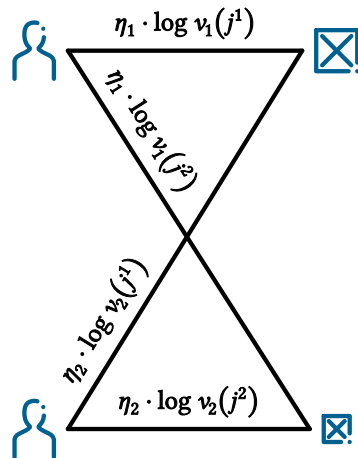
# 2

## RepReMatch

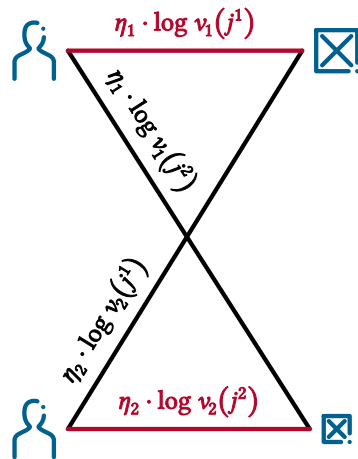


# Naïve Approach









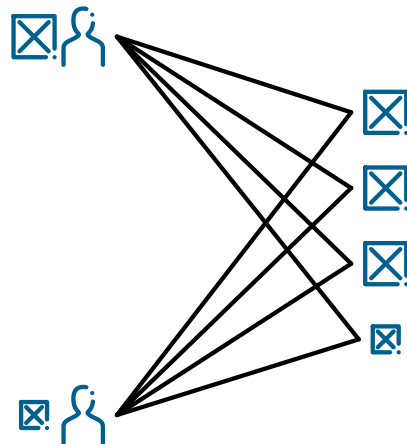
# Naïve Approach



# Naïve Approach

Greedy algorithm:

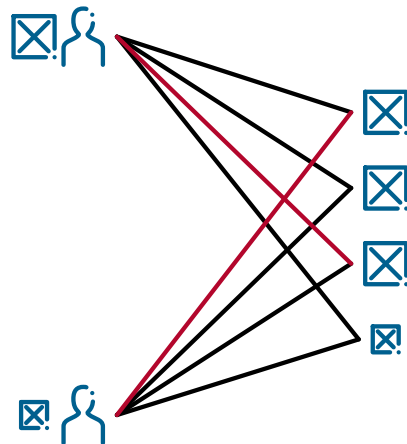
- repeatedly use maximum weight matchings



# Naïve Approach

Greedy algorithm:

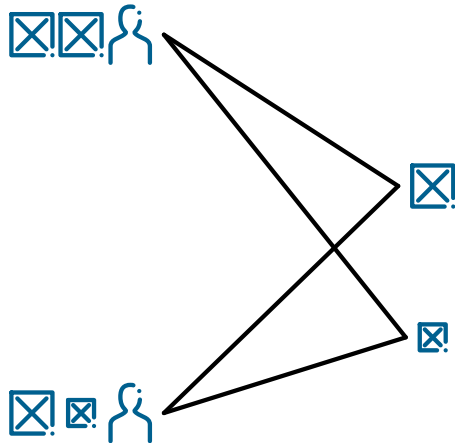
- repeatedly use maximum weight matchings



# Naïve Approach

Greedy algorithm:

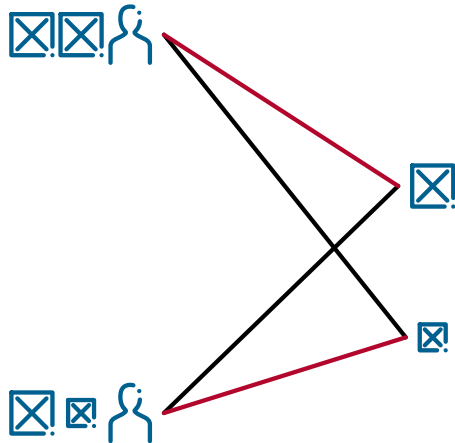
- repeatedly use maximum weight matchings



# Naïve Approach

Greedy algorithm:

- repeatedly use maximum weight matchings



# Naïve Approach

Greedy algorithm:

- repeatedly use maximum weight matchings



# Naïve Approach

Greedy algorithm:

- repeatedly use maximum weight matchings
- fails because of missing foresight





# Naïve Approach

Greedy algorithm:

- repeatedly use maximum weight matchings
- fails because of missing foresight
  - **additive valuations:** sort items by valuation  
 $\Rightarrow 2n$ -approximation (*SMatch*)



# Naïve Approach

Greedy algorithm:

- repeatedly use maximum weight matchings
- fails because of missing foresight
  - **additive valuations:** sort items by valuation  
 $\Rightarrow 2n$ -approximation (*SMatch*)
  - **submodular valuations:** worst-case valuation  
approximable only by  $\Omega(\sqrt{m/\ln m})$



# Naïve Approach

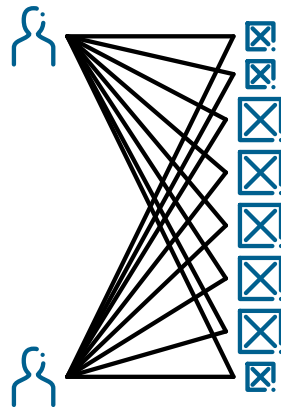
Greedy algorithm:

- repeatedly use maximum weight matchings
- fails because of missing foresight
  - **additive valuations:** sort items by valuation  
 $\Rightarrow 2n$ -approximation (*SMatch*)
  - **submodular valuations:** worst-case valuation  
approximable only by  $\Omega(\sqrt{m/\ln m})$  ⚡



# Key Ideas of the Algorithm

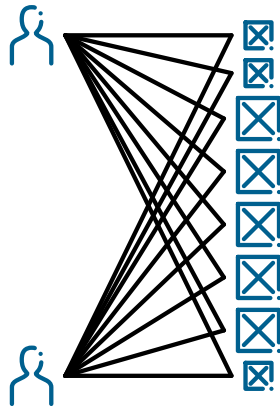
We need change the past in three phases:



# Key Ideas of the Algorithm

We need change the past in three phases:

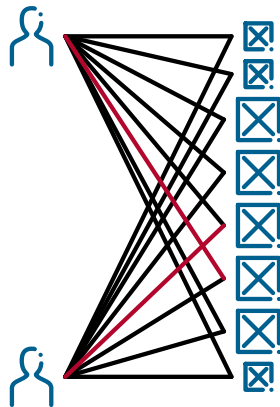
**Phase I** Assign enough high-value items **temporarily**.



# Key Ideas of the Algorithm

We need change the past in three phases:

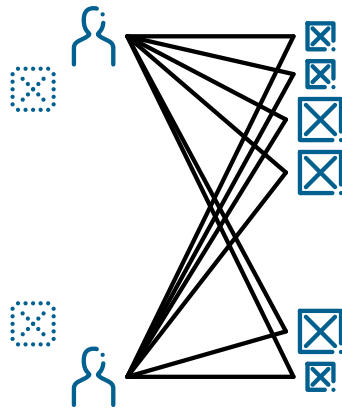
**Phase I** Assign enough high-value items temporarily.



# Key Ideas of the Algorithm

We need change the past in three phases:

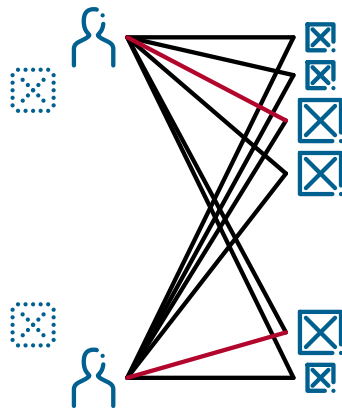
**Phase I** Assign enough high-value items temporarily.



# Key Ideas of the Algorithm

We need change the past in three phases:

**Phase I** Assign enough high-value items temporarily.

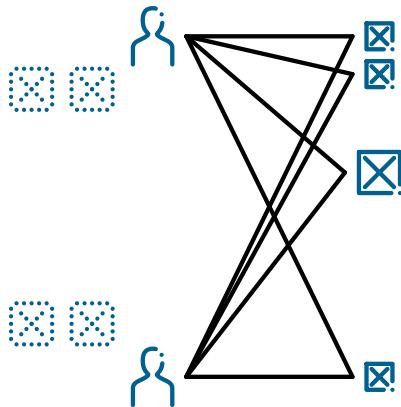




# Key Ideas of the Algorithm

We need change the past in three phases:

**Phase I** Assign enough high-value items temporarily.

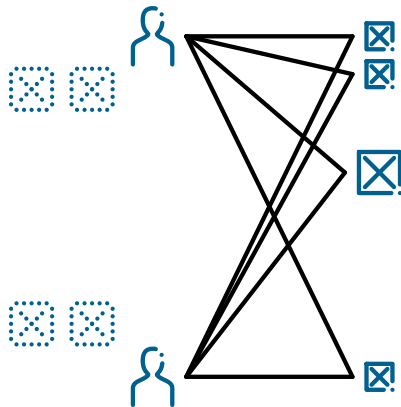


# Key Ideas of the Algorithm

We need change the past in three phases:

**Phase I** Assign enough high-value items temporarily.

**Phase II** Assign the remaining items **definitely**.

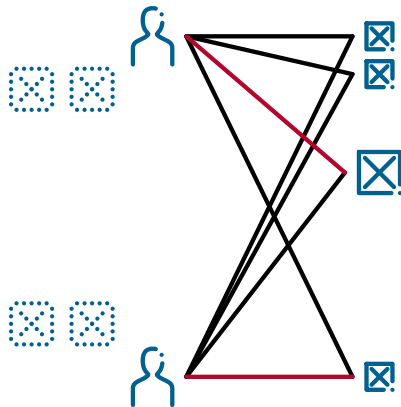


# Key Ideas of the Algorithm

We need change the past in three phases:

**Phase I** Assign enough high-value items temporarily.

**Phase II** Assign the remaining items definitely.

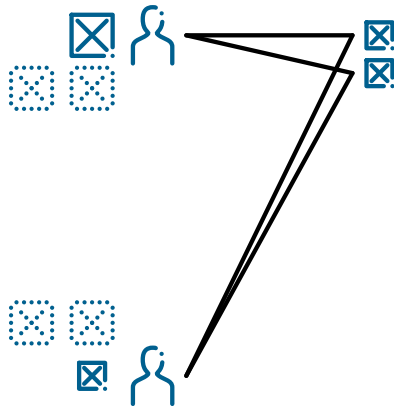


# Key Ideas of the Algorithm

We need change the past in three phases:

**Phase I** Assign enough high-value items temporarily.

**Phase II** Assign the remaining items definitely.

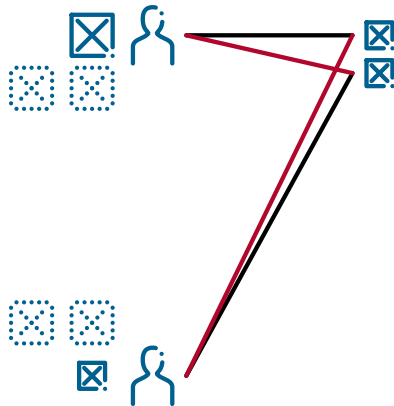


# Key Ideas of the Algorithm

We need change the past in three phases:

**Phase I** Assign enough high-value items temporarily.

**Phase II** Assign the remaining items definitely.



# Key Ideas of the Algorithm

We need change the past in three phases:

**Phase I** Assign enough high-value items temporarily.

**Phase II** Assign the remaining items definitely.



# Key Ideas of the Algorithm

We need 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**.



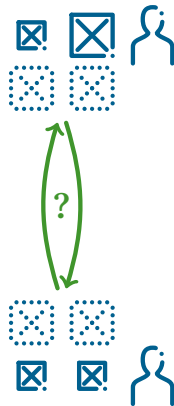
# Key Ideas of the Algorithm

We need 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.





# Key Ideas of the Algorithm

We need 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.



# Key Ideas of the Algorithm

We need 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.



## Theorem

RepReMatch guarantees a  $2n(\log_2 n + 3)$ -approximation under submodular valuations.



# The Algorithm



# The Algorithm

Phase I:



# The Algorithm

Phase I:

**1** repeat  $\lceil \log_2 n \rceil + 1$  times



# The Algorithm

Phase I:

**1** repeat  $\lceil \log_2 n \rceil + 1$  times

**1** create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$



# The Algorithm

Phase I:

- 1** repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1** create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2** compute maximum weight matching



# The Algorithm

Phase I:

- 1** repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1** create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2** compute maximum weight matching
  - 3** update bundles  $\mathbf{x}_i^1$  & remove assigned items



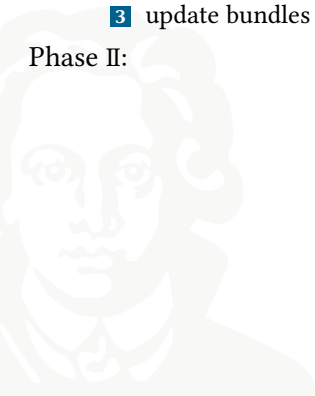


# The Algorithm

Phase I:

- 1 repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2 compute maximum weight matching
  - 3 update bundles  $x_i^1$  & remove assigned items

Phase II:



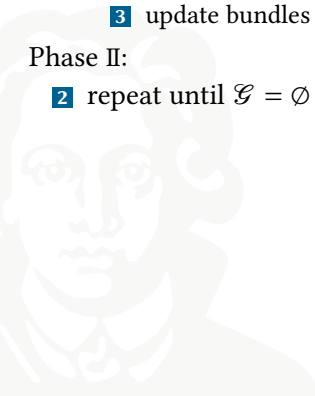
# The Algorithm

Phase I:

- 1 repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2 compute maximum weight matching
  - 3 update bundles  $x_i^1$  & remove assigned items

Phase II:

- 2 repeat until  $\mathcal{G} = \emptyset$



# The Algorithm

Phase I:

- 1 repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase II:

- 2 repeat until  $\mathcal{G} = \emptyset$ 
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$

# The Algorithm

Phase I:

- 1 repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase II:

- 2 repeat until  $\mathcal{G} = \emptyset$ 
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$
  - 2 compute maximum weight matching

# The Algorithm

Phase I:

- 1 repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase II:

- 2 repeat until  $\mathcal{G} = \emptyset$ 
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

# The Algorithm

Phase I:

- 1 repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase II:

- 2 repeat until  $\mathcal{G} = \emptyset$ 
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase III:

# The Algorithm

Phase I:

- 1 repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase II:

- 2 repeat until  $\mathcal{G} = \emptyset$ 
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase III:

- 3 create bipartite graph  $(\mathcal{A}, \bigcup_{i \in \mathcal{A}} \mathbf{x}_i^I, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$

# The Algorithm

Phase I:

- 1 repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase II:

- 2 repeat until  $\mathcal{G} = \emptyset$ 
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase III:

- 3 create bipartite graph  $(\mathcal{A}, \bigcup_{i \in \mathcal{A}} \mathbf{x}_i^I, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$
- 4 compute maximum weight matching



# The Algorithm

Phase I:

- 1 repeat  $\lceil \log_2 n \rceil + 1$  times
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(j)$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase II:

- 2 repeat until  $\mathcal{G} = \emptyset$ 
  - 1 create bipartite graph  $(\mathcal{A}, \mathcal{G}, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$
  - 2 compute maximum weight matching
  - 3 update bundles  $\mathbf{x}_i^I$  & remove assigned items

Phase III:

- 3 create bipartite graph  $(\mathcal{A}, \bigcup_{i \in \mathcal{A}} \mathbf{x}_i^I, E)$  with edge weights  $\eta_i \cdot \log v_i(\mathbf{x}_i^I \cup \{j\})$
- 4 compute maximum weight matching
- 5 create bundles  $\mathbf{x}_i^I$

# Analysing Phases I & III (1/2)



# Analysing Phases I & III (1/2)

Phase I reserves 'high-value' items.



## Analysing Phases I & III (1/2)

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



## Analysing Phases I & III (1/2)

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

### Definition

Let  $\mathbf{x}_i^* = \{o_i^1, o_i^2, \dots\}$  be an optimal bundle.



## Analysing Phases I & III (1/2)

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

### Definition

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



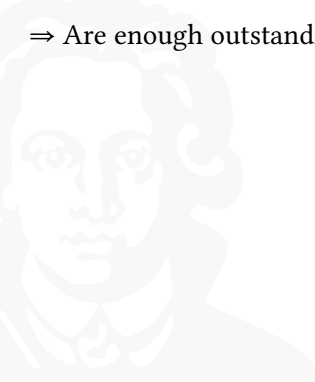
## Analysing Phases I & III (1/2)

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

### Definition

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

⇒ Are enough outstanding items reserved?



# Analysing Phases I & III (2/2)

## *Lemma*

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



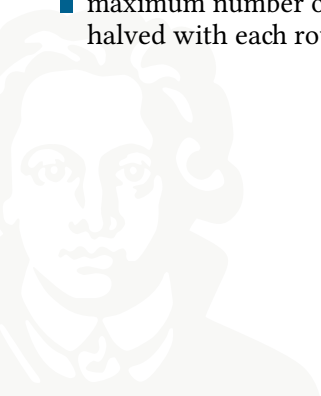


# Analysing Phases I & III (2/2)

## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I

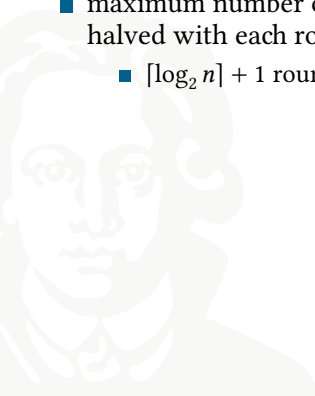


# Analysing Phases I & III (2/2)

## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough

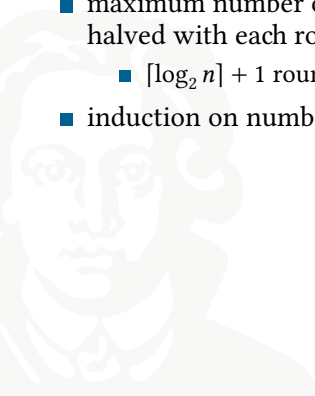


# Analysing Phases I & III (2/2)

## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I



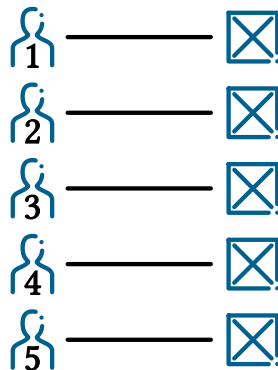
# Analysing Phases I & III (2/2)

## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either



# Analysing Phases I & III (2/2)

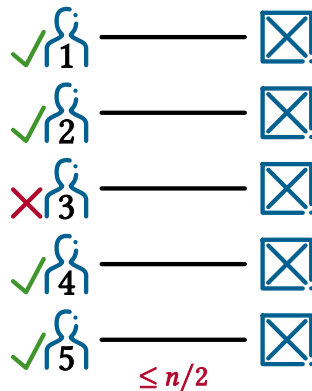
## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$  many agents without an outstanding item



# Analysing Phases I & III (2/2)

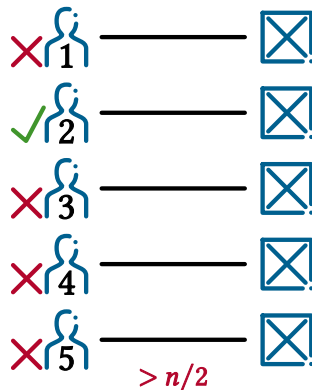
## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$  many agents without an outstanding item
- $> n/2$  many agents without an outstanding item



# Analysing Phases I & III (2/2)

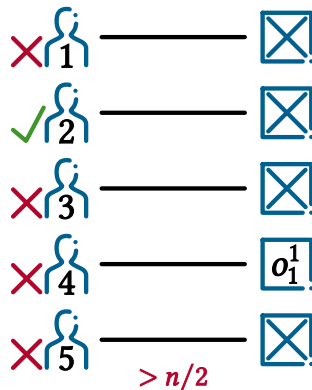
## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$  many agents without an outstanding item
- $> n/2$  many agents without an outstanding item



# Analysing Phases I & III (2/2)

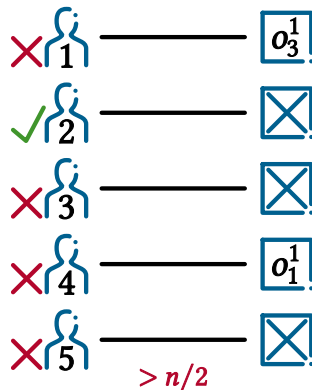
## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$  many agents without an outstanding item
- $> n/2$  many agents without an outstanding item





# Analysing Phases I & III (2/2)

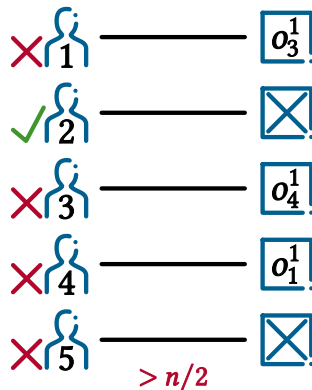
## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$  many agents without an outstanding item
- $> n/2$  many agents without an outstanding item



# Analysing Phases I & III (2/2)

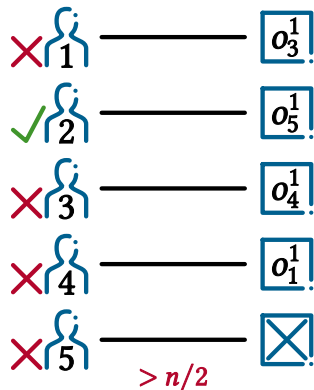
## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$  many agents without an outstanding item
- $> n/2$  many agents without an outstanding item



# Analysing Phases I & III (2/2)

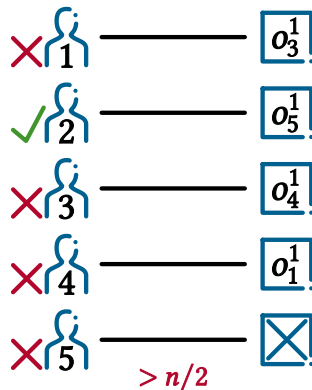
## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$  many agents without an outstanding item
- $> n/2$  many agents without an outstanding item
  - $> n/2$  many items  $o_i^1$  assigned to someone else



# Analysing Phases I & III (2/2)

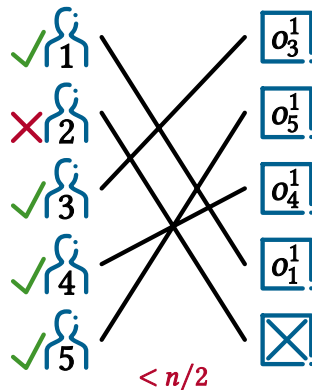
## Lemma

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

- maximum number of agents *without* outstanding item halved with each round of phase I
  - $\lceil \log_2 n \rceil + 1$  rounds in phase I are enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$  many agents without an outstanding item
- $> n/2$  many agents without an outstanding item
  - $> n/2$  many items  $o_i^1$  assigned to someone else
  - $> n/2$  many agents matched with outstanding item upon release in phase III



# Analysing Phase II (1/2)



# Analysing Phase II (1/2)

## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in \mathbf{x}_i^*$



# Analysing Phase II (1/2)

## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in \mathbf{x}_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .



## Analysing Phase II (1/2)

### Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in \mathbf{x}_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

### Definition

Let  $\mathbf{x}_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ .



## Analysing Phase II (1/2)

### Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in \mathbf{x}_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

### Definition

Let  $\mathbf{x}_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{\mathbf{x}}_{i,r}^* := \left\{ \right.$$

## Analysing Phase II (1/2)

### Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in \mathbf{x}_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

### Definition

Let  $\mathbf{x}_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{\mathbf{x}}_{i,r}^* := \left\{ \begin{array}{l} \mathbf{x}_i^* \setminus \mathcal{L}_{i,r} \\ \cup \mathcal{L}_{i,r} \end{array} \right. \quad \text{in round } r = 1,$$

## Analysing Phase II (1/2)

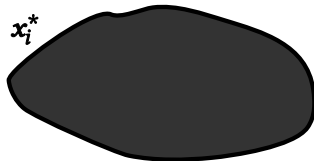
### Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

### Definition

Let  $x_i^{\Pi} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \left\{ \begin{array}{l} \text{items } j \in x_i^* \text{ such that } j \notin \mathcal{L}_{i,r} \end{array} \right\} \quad \text{in round } r = 1,$$



# Analysing Phase II (1/2)

## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\Pi} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \left\{ \begin{array}{l} \text{items } j \in x_i^* \text{ such that } j \notin \mathcal{L}_{i,r} \end{array} \right. \quad \text{in round } r = 1,$$



# Analysing Phase II (1/2)

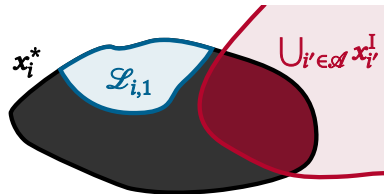
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\Pi} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \left\{ \begin{array}{l} \text{items } j \text{ such that } j \in x_i^* \text{ and } j \notin \mathcal{L}_{i,r} \end{array} \right\} \quad \text{in round } r = 1,$$



# Analysing Phase II (1/2)

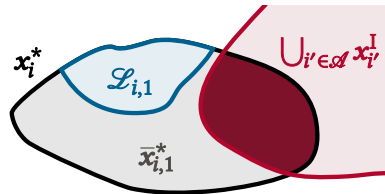
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \left\{ \begin{array}{l} \text{items } j \text{ such that } j \in x_i^* \text{ and } j \notin \mathcal{L}_{i,r} \end{array} \right. \quad \text{in round } r = 1,$$



# Analysing Phase II (1/2)

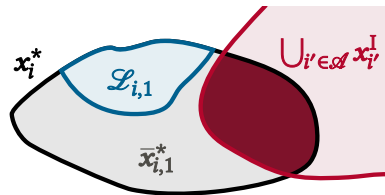
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \begin{cases} x_i^* \setminus (\bigcup_{i' \in \mathcal{A}} x_{i'}^{\text{I}} \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \end{cases}$$



# Analysing Phase II (1/2)

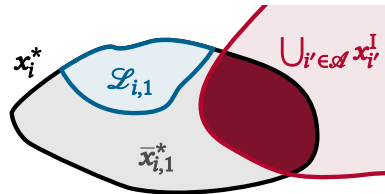
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\Pi} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \begin{cases} x_i^* \setminus (\bigcup_{i' \in \mathcal{A}} x_{i'}^I \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \\ & \text{in round } r \geq 2. \end{cases}$$





# Analysing Phase II (1/2)

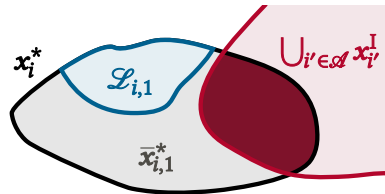
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \begin{cases} x_i^* \setminus (\bigcup_{i' \in \mathcal{A}} x_{i'}^{\text{I}} \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \\ & \text{in round } r \geq 2. \end{cases}$$



# Analysing Phase II (1/2)

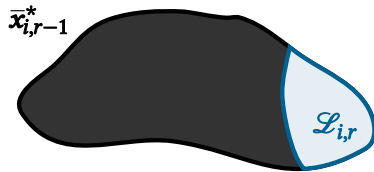
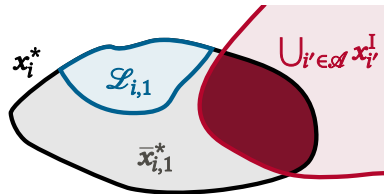
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \begin{cases} x_i^* \setminus (\bigcup_{i' \in \mathcal{A}} x_{i'}^{\text{I}} \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \\ & \text{in round } r \geq 2. \end{cases}$$



# Analysing Phase II (1/2)

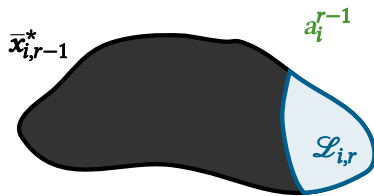
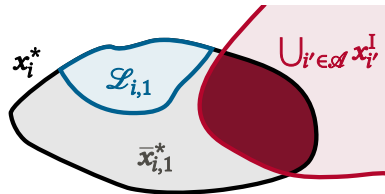
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in \mathbf{x}_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $\mathbf{x}_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{\mathbf{x}}_{i,r}^* := \begin{cases} \mathbf{x}_i^* \setminus (\bigcup_{i' \in \mathcal{A}} \mathbf{x}_{i'}^{\text{I}} \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \\ & \text{in round } r \geq 2. \end{cases}$$



# Analysing Phase II (1/2)

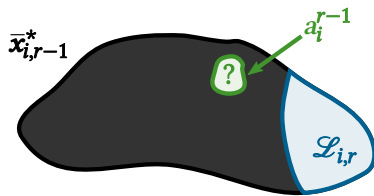
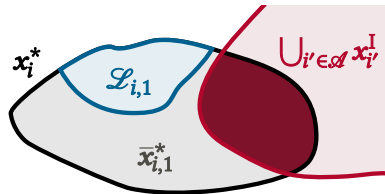
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \begin{cases} x_i^* \setminus (\bigcup_{i' \in \mathcal{A}} x_{i'}^{\text{I}} \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \\ & \text{in round } r \geq 2. \end{cases}$$



# Analysing Phase II (1/2)

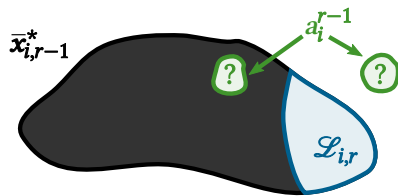
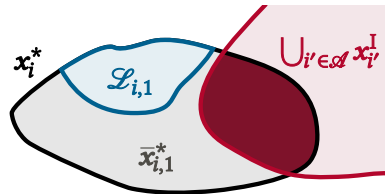
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \begin{cases} x_i^* \setminus (\bigcup_{i' \in \mathcal{A}} x_{i'}^{\text{I}} \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \\ & \text{in round } r \geq 2. \end{cases}$$



# Analysing Phase II (1/2)

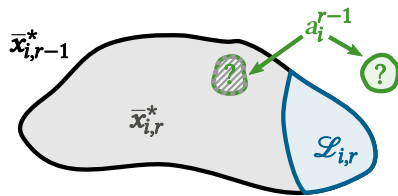
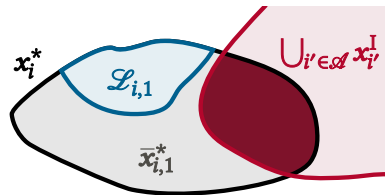
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \begin{cases} x_i^* \setminus (\bigcup_{i' \in \mathcal{A}} x_{i'}^{\text{I}} \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \\ & \text{in round } r \geq 2. \end{cases}$$



# Analysing Phase II (1/2)

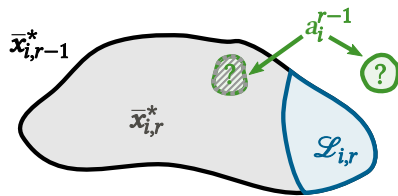
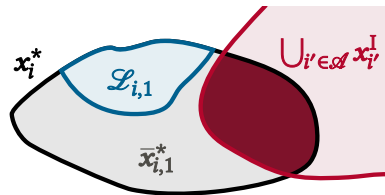
## Definition

The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in \mathbf{x}_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $\mathbf{x}_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{\mathbf{x}}_{i,r}^* := \begin{cases} \mathbf{x}_i^* \setminus (\bigcup_{i' \in \mathcal{A}} \mathbf{x}_{i'}^{\text{I}} \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \\ \bar{\mathbf{x}}_{i,r-1}^* \setminus (\mathcal{L}_{i,r} \cup \{a_i^{r-1}\}) & \text{in round } r \geq 2. \end{cases}$$



# Analysing Phase II (1/2)

## Definition

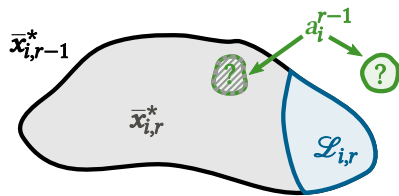
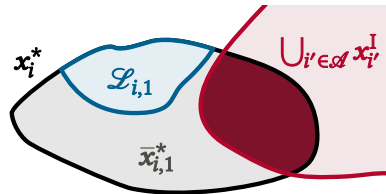
The set  $\mathcal{L}_{i,r}$  of *lost items* is the set of all optimal items  $j \in x_i^*$  matched with other agents  $i' \neq i$  in round  $r$ .

## Definition

Let  $x_i^{\text{II}} = \{a_i^1, a_i^2, \dots\}$  be the bundle of agent  $i$ . The set of *optimal and attainable items* is defined as

$$\bar{x}_{i,r}^* := \begin{cases} x_i^* \setminus (\bigcup_{i' \in \mathcal{A}} x_{i'}^{\text{I}} \cup \mathcal{L}_{i,1}) & \text{in round } r = 1, \\ \bar{x}_{i,r-1}^* \setminus (\mathcal{L}_{i,r} \cup \{a_i^{r-1}\}) & \text{in round } r \geq 2. \end{cases}$$

⇒ What is the valuation of the remaining items?





# Analysing Phase II (2/2)

## *Lemma*

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

if  $r \geq 2$



## Analysing Phase II (2/2)

## Lemma

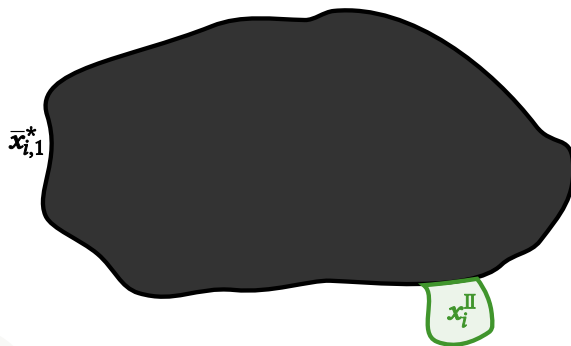
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1})$$

if  $r \geq 2$ 
$$\bar{x}_{i,1}^*$$

## Analysing Phase II (2/2)

## Lemma

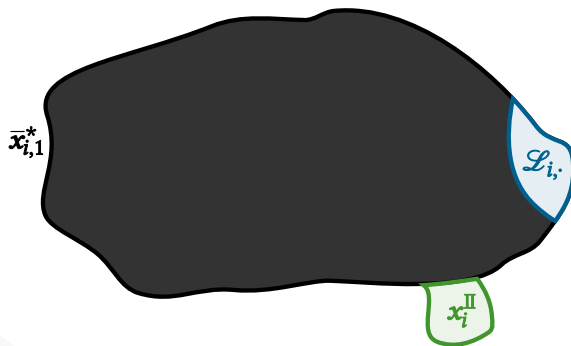
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1})$$

if  $r \geq 2$ 

## Analysing Phase II (2/2)

## Lemma

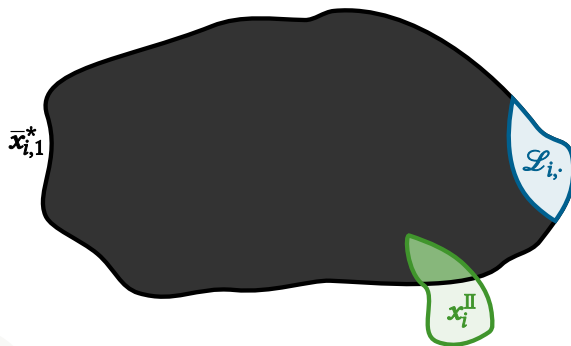
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1})$$

if  $r \geq 2$ 

## Analysing Phase II (2/2)

## Lemma

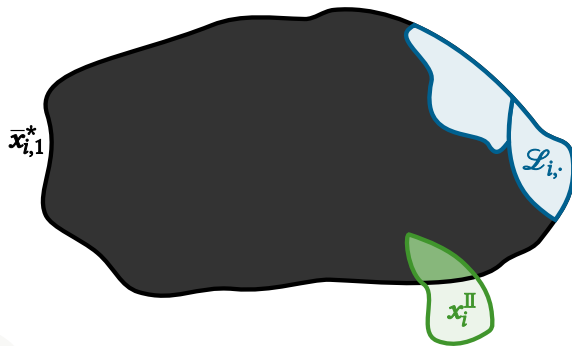
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1})$$

if  $r \geq 2$ 

## Analysing Phase II (2/2)

## Lemma

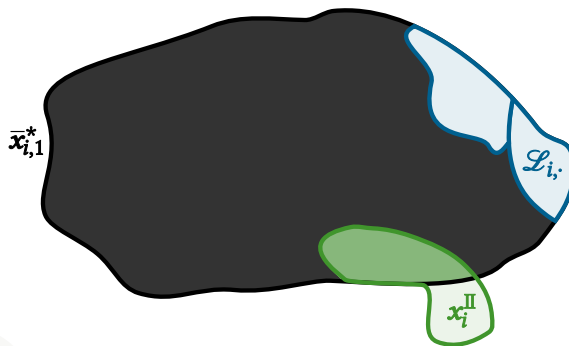
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1})$$

if  $r \geq 2$ 

## Analysing Phase II (2/2)

## Lemma

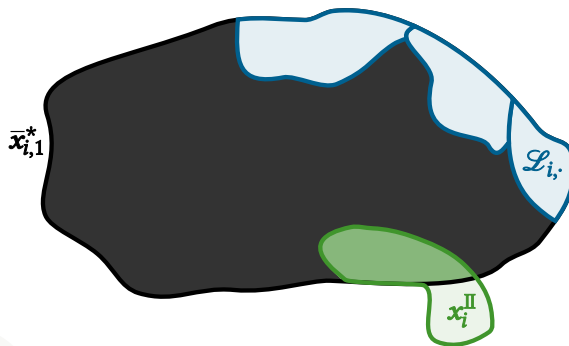
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1})$$

if  $r \geq 2$ 

## Analysing Phase II (2/2)

## Lemma

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

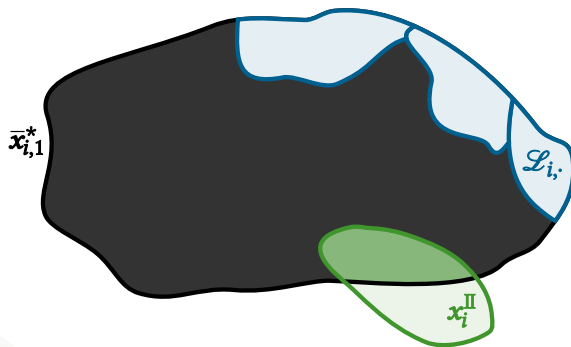
if  $r \geq 2$ 



## Analysing Phase II (2/2)

## Lemma

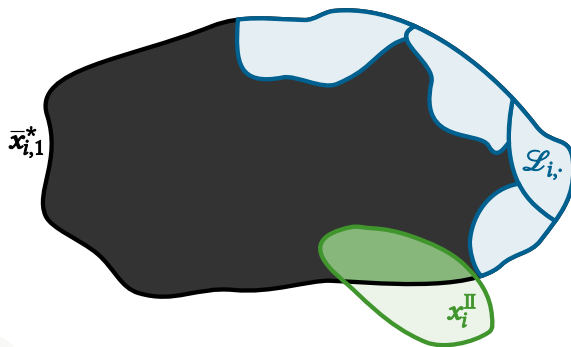
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1})$$

if  $r \geq 2$ 

## Analysing Phase II (2/2)

## Lemma

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

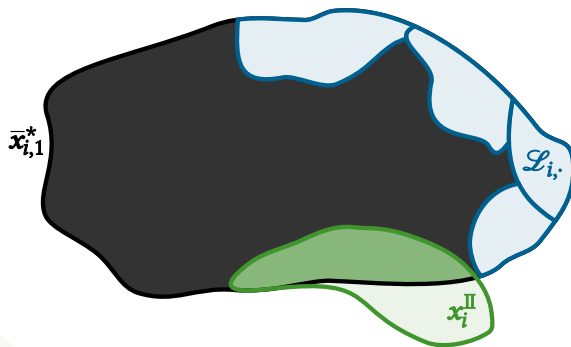
if  $r \geq 2$ 

# Analysing Phase II (2/2)

## Lemma

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

if  $r \geq 2$

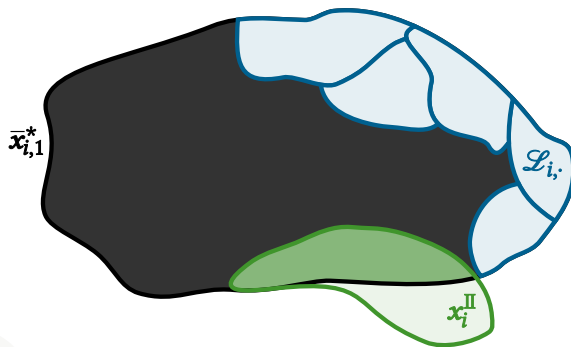


# Analysing Phase II (2/2)

## Lemma

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

if  $r \geq 2$

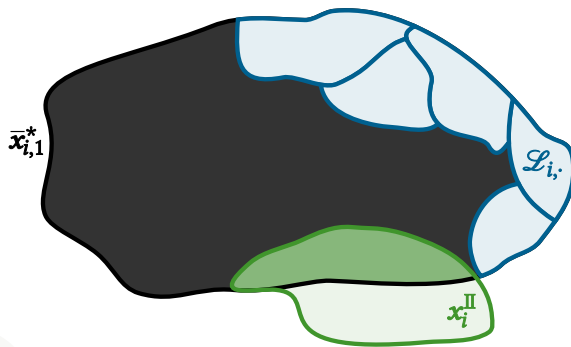


# Analysing Phase II (2/2)

## Lemma

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

if  $r \geq 2$

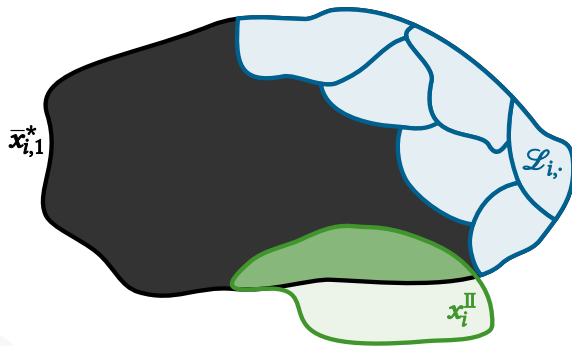


# Analysing Phase II (2/2)

## Lemma

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

if  $r \geq 2$

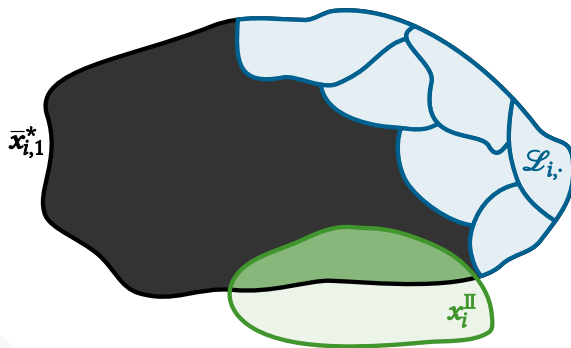


# Analysing Phase II (2/2)

## Lemma

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

if  $r \geq 2$

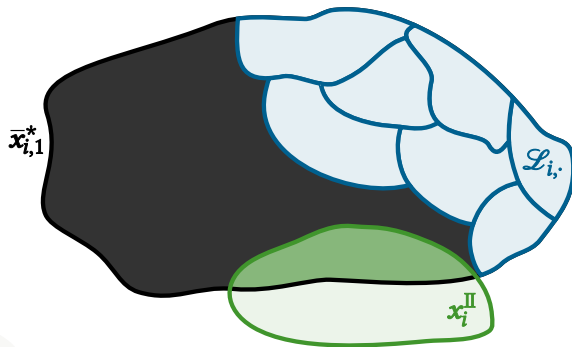


# Analysing Phase II (2/2)

## Lemma

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

if  $r \geq 2$



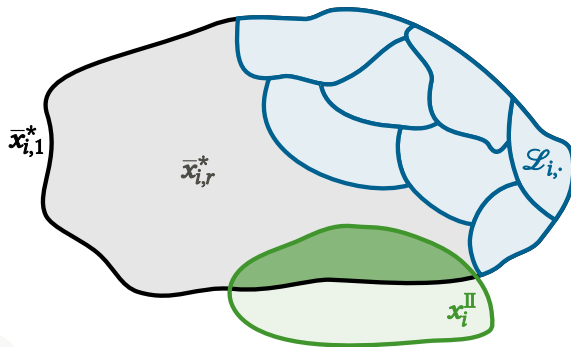


# Analysing Phase II (2/2)

## Lemma

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

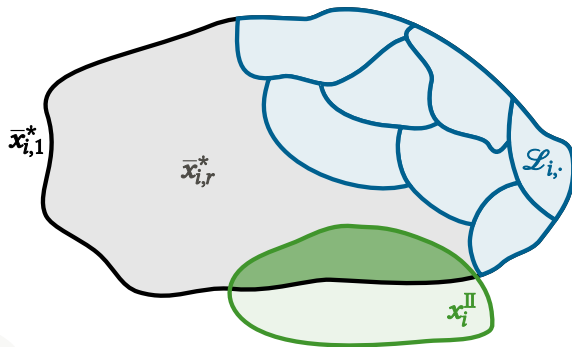
if  $r \geq 2$



# Analysing Phase II (2/2)

## Lemma

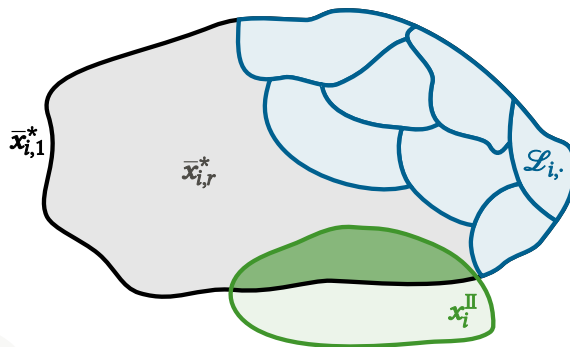
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) = v_i(\bar{x}_{i,r}^* \cup \{a_i^1, \dots, a_i^{r-1}\}) - v_i(a_i^1, \dots, a_i^{r-1}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

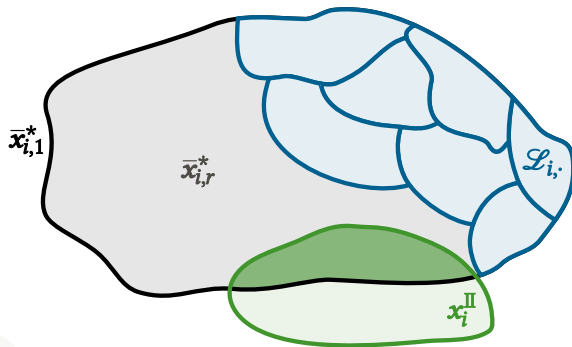
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) = v_i(\bar{x}_{i,r}^* \cup \{a_i^1, \dots, a_i^{r-1}\}) - v_i(a_i^1, \dots, a_i^{r-1}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) = -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,r}^* \cup \{a_i^1, \dots, a_i^{r-1}\}) \quad \text{if } r \geq 2$$

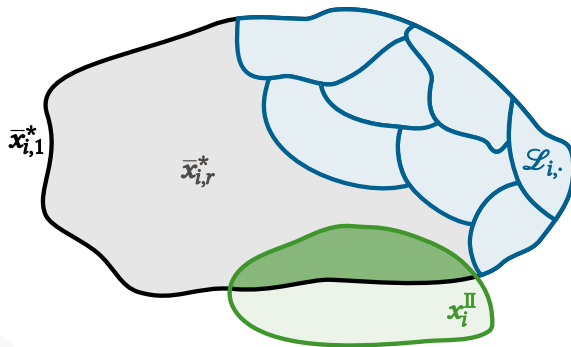


# Analysing Phase II (2/2)

## Lemma

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

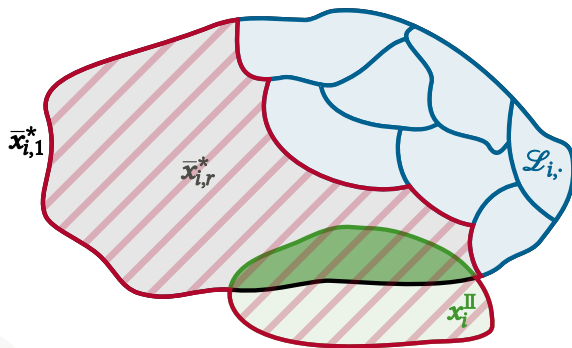
if  $r \geq 2$



# Analysing Phase II (2/2)

## Lemma

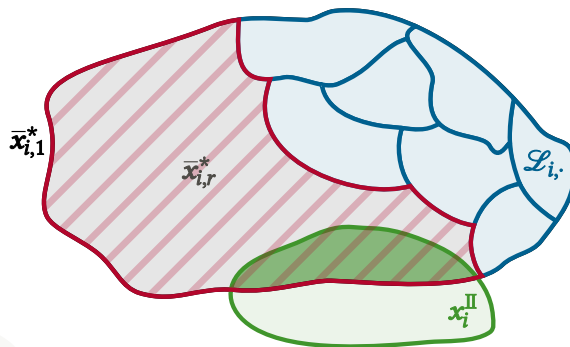
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) = -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,r}^* \cup \{a_i^1, \dots, a_i^{r-1}\}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) = -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,r}^* \cup \{a_i^1, \dots, a_i^{r-1}\}) \quad \text{if } r \geq 2$$

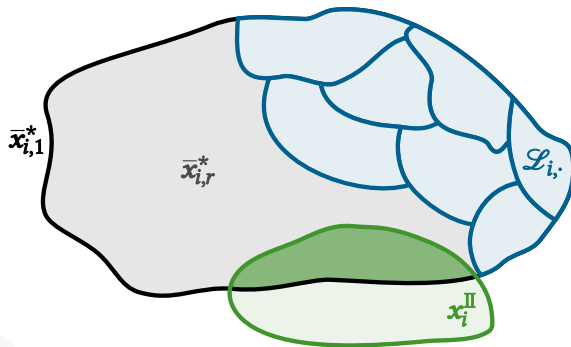


# Analysing Phase II (2/2)

## Lemma

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

if  $r \geq 2$

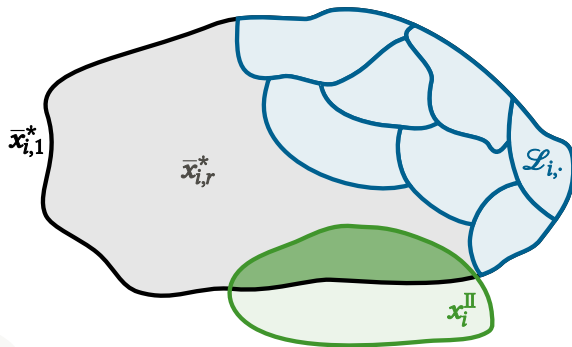




# Analysing Phase II (2/2)

## Lemma

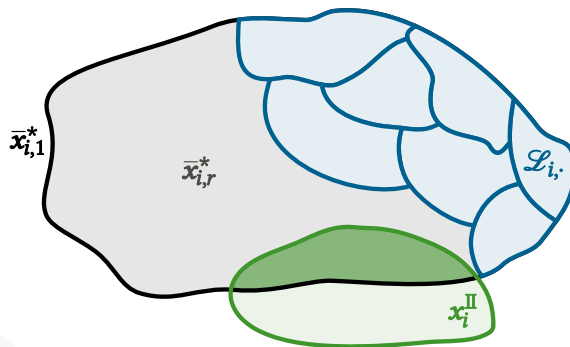
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - v_i(\mathcal{L}_{i,2} \mid a_i^1) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

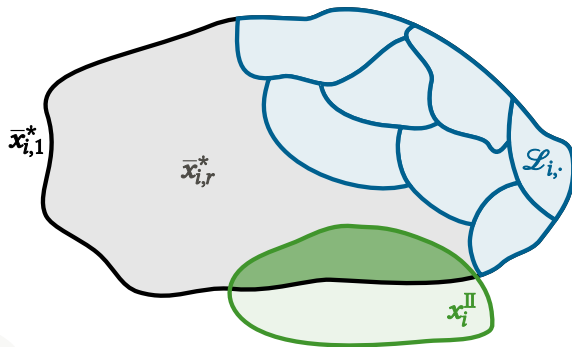
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - v_i(\mathcal{L}_{i,2} \mid a_i^1) - v_i(\mathcal{L}_{i,3} \mid a_i^1, a_i^2) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

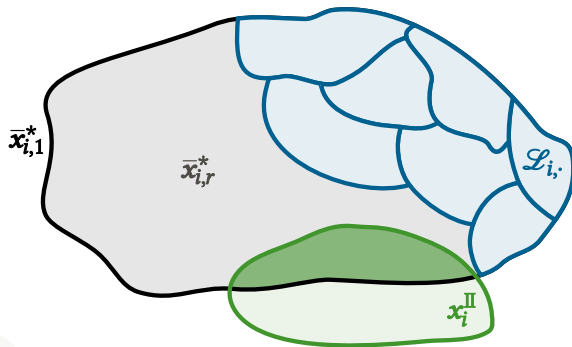
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - v_i(\mathcal{L}_{i,2} \mid a_i^1) - v_i(\mathcal{L}_{i,3} \mid a_i^1, a_i^2) - \dots \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

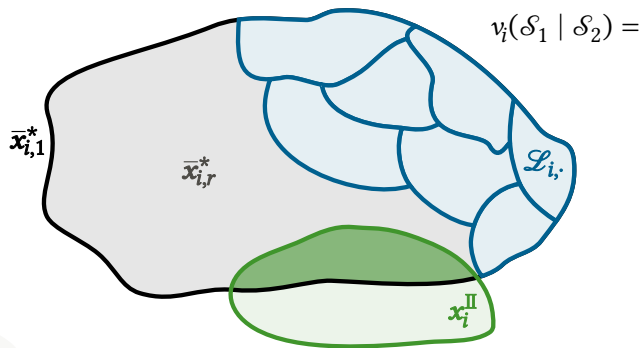
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r v_i(\mathcal{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

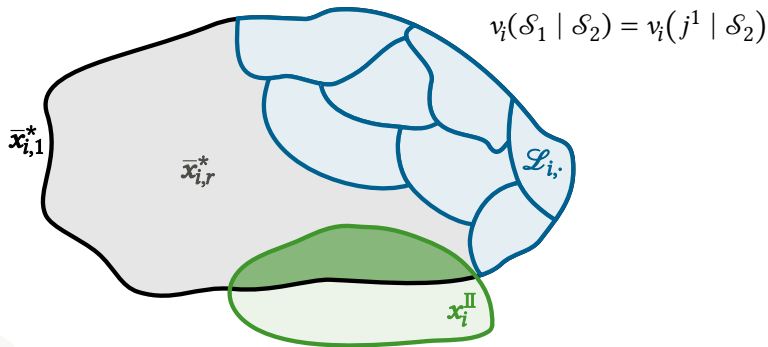
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r v_i(\mathcal{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

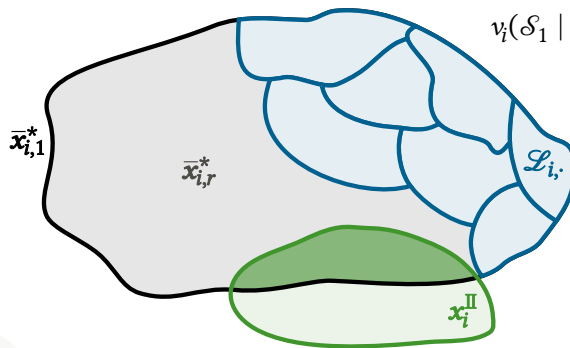
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r v_i(\mathcal{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r v_i(\mathcal{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$

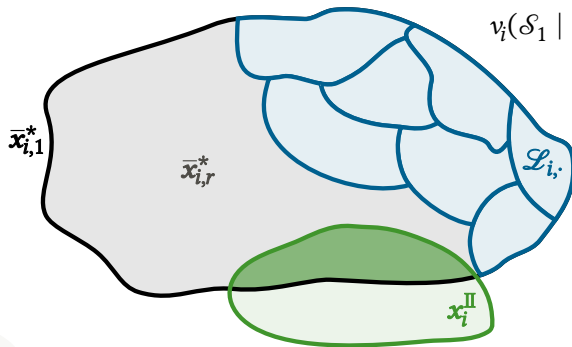


$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) = v_i(j^1 \mid \mathcal{S}_2) + v_i(j^2 \mid \mathcal{S}_2 \cup \{j^1\})$$

# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r v_i(\mathcal{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$



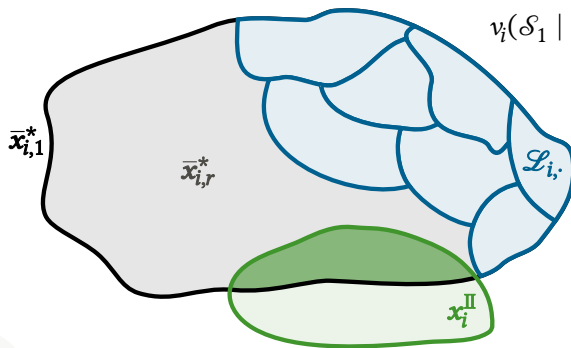
$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) = v_i(j^1 \mid \mathcal{S}_2) + v_i(j^2 \mid \mathcal{S}_2 \cup \{j^1\}) + v_i(j^3 \mid \mathcal{S}_2 \cup \{j^1, j^2\})$$



# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r v_i(\mathcal{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$

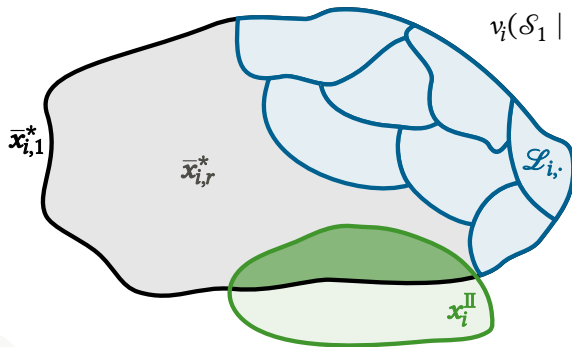


$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) = v_i(j^1 \mid \mathcal{S}_2) + \\ v_i(j^2 \mid \mathcal{S}_2 \cup \{j^1\}) + \\ v_i(j^3 \mid \mathcal{S}_2 \cup \{j^1, j^2\}) + \\ \vdots$$

# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r v_i(\mathcal{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$

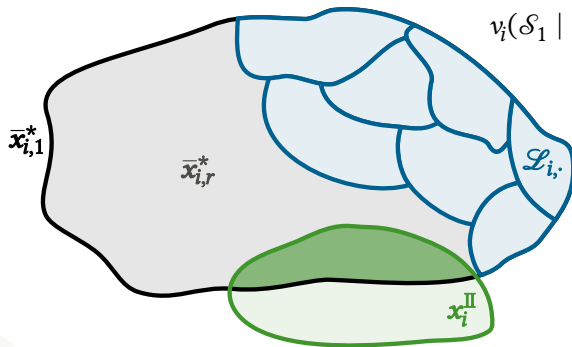


$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) = v_i(j^1 \mid \mathcal{S}_2) + \\ v_i(j^2 \mid \mathcal{S}_2 \cup \{j^1\}) + \\ v_i(j^3 \mid \mathcal{S}_2 \cup \{j^1, j^2\}) + \\ \vdots$$

# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r v_i(\mathcal{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$



$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) \leq v_i(j^1 \mid \mathcal{S}_2) +$$

$$v_i(j^2 \mid \mathcal{S}_2) +$$

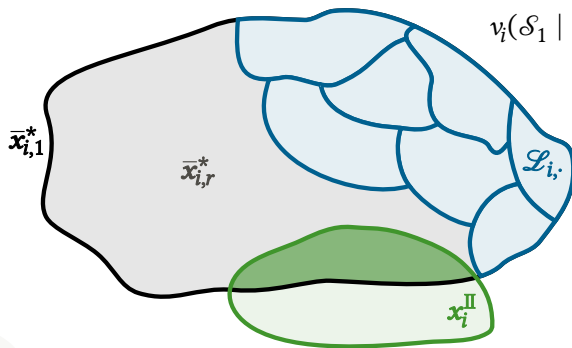
$$v_i(j^3 \mid \mathcal{S}_2) +$$

$$\vdots$$

# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r v_i(\mathcal{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$

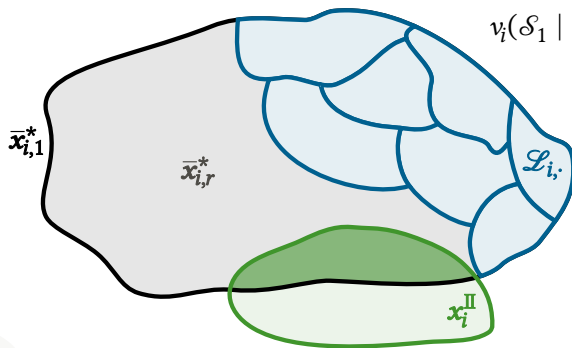


$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) \leq \sum_{j \in \mathcal{S}_1} v_i(j \mid \mathcal{S}_2)$$

# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r \sum_{j \in \mathcal{L}_{i,l}} v_i(j \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$

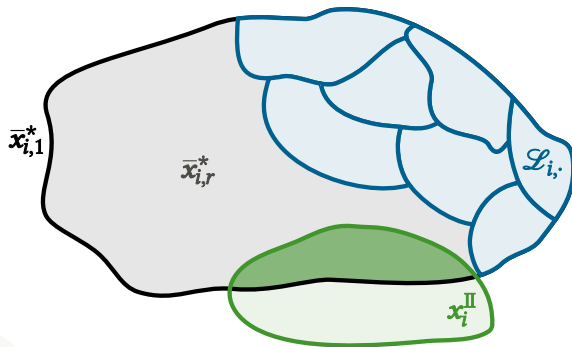


$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) \leq \sum_{j \in \mathcal{S}_1} v_i(j \mid \mathcal{S}_2)$$

# Analysing Phase II (2/2)

## Lemma

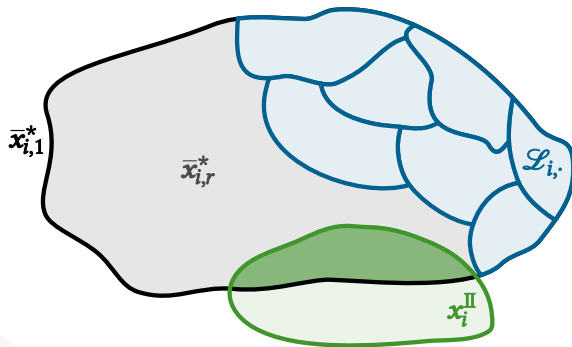
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r \sum_{j \in \mathcal{L}_{i,l}} v_i(j \mid a_i^1, \dots, a_i^{l-1}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

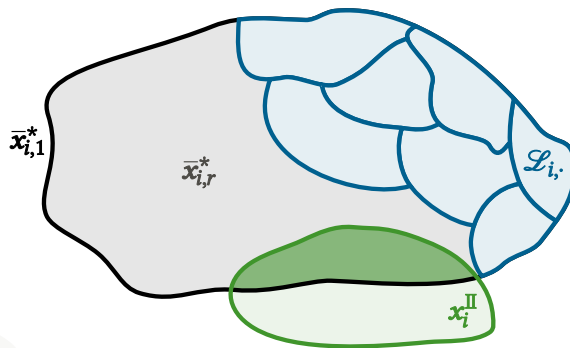
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r \sum_{j \in \mathcal{L}_{i,l}} v_i(j \mid a_i^1, \dots, \mathbf{a}_i^{l-1}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r \sum_{j \in \mathcal{L}_{i,l}} v_i(j \mid a_i^1, \dots, \mathbf{a}_i^{l-2}) \quad \text{if } r \geq 2$$

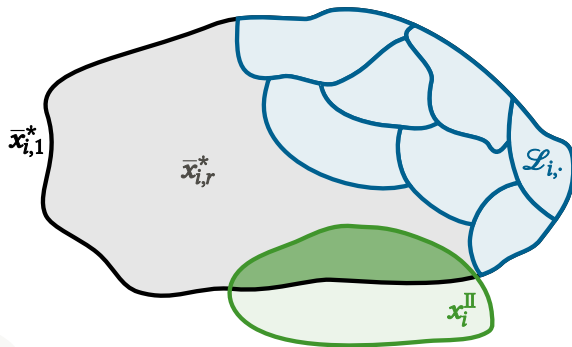




# Analysing Phase II (2/2)

## Lemma

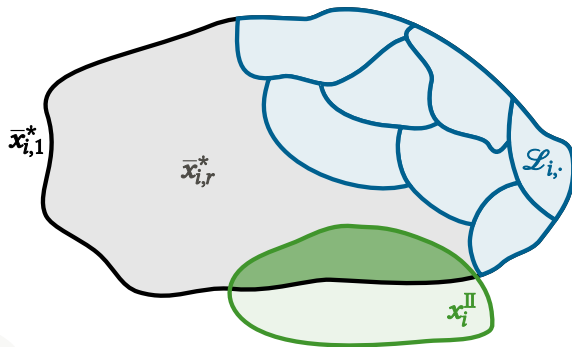
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r \sum_{j \in \mathcal{L}_{i,l}} v_i(j \mid a_i^1, \dots, a_i^{l-2}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

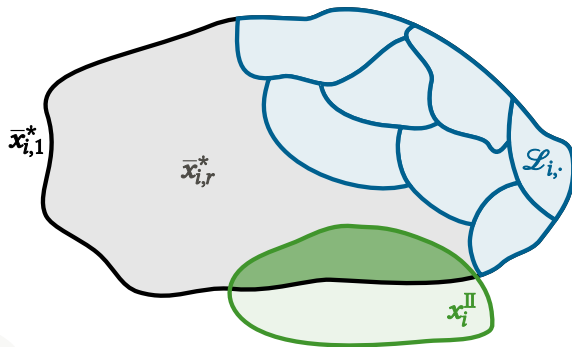
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r \sum_{j \in \mathcal{L}_{i,l}} v_i(a_i^{l-1} \mid a_i^1, \dots, a_i^{l-2}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

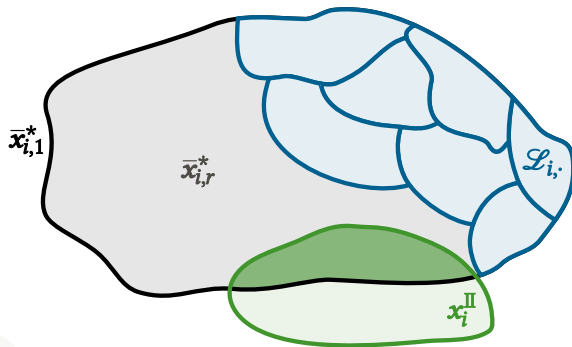
$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r |\mathcal{L}_{i,l}| \cdot v_i(a_i^{l-1} \mid a_i^1, \dots, a_i^{l-2}) \quad \text{if } r \geq 2$$



# Analysing Phase II (2/2)

## Lemma

$$v_i(\bar{x}_{i,r}^* \mid a_i^1, \dots, a_i^{r-1}) \geq -v_i(a_i^1, \dots, a_i^{r-1}) + v_i(\bar{x}_{i,1}^*) - \sum_{l=2}^r (n-1) \cdot v_i(a_i^{l-1} \mid a_i^1, \dots, a_i^{l-2}) \quad \text{if } r \geq 2$$



# 3 Conclusion



# Summary & Outlook



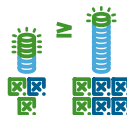
# Summary & Outlook

- allocation  $\triangleq$  tuple of bundles



# Summary & Outlook

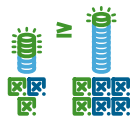
- allocation  $\triangleq$  tuple of bundles
- valuation through submodular functions





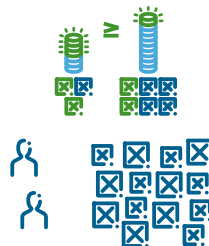
## Summary & Outlook

- allocation  $\triangleq$  tuple of bundles
- valuation through submodular functions
- Nash social welfare  $\triangleq$  geometric mean of valuations



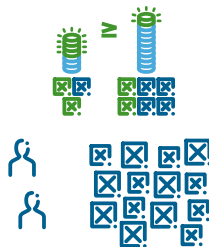
# Summary & Outlook

- allocation  $\triangleq$  tuple of bundles
- valuation through submodular functions
- Nash social welfare  $\triangleq$  geometric mean of valuations
- approximation factor independent from  $m$ ?



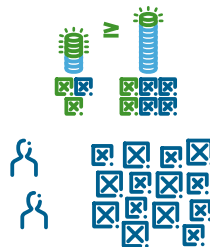
## Summary & Outlook

- allocation  $\triangleq$  tuple of bundles
- valuation through submodular functions
- Nash social welfare  $\triangleq$  geometric mean of valuations
- approximation factor independent from  $m$ ?
- simple matching failing because of missing foresight



# Summary & Outlook

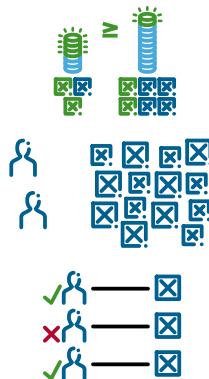
- allocation  $\triangleq$  tuple of bundles
- valuation through submodular functions
- Nash social welfare  $\triangleq$  geometric mean of valuations
- approximation factor independent from  $m$ ?
- simple matching failing because of missing foresight
- RepReMatch:  $2n(\log n + 3)$ -approximative



# Summary & Outlook

- allocation  $\triangleq$  tuple of bundles
- valuation through submodular functions
- Nash social welfare  $\triangleq$  geometric mean of valuations
- approximation factor independent from  $m$ ?
- simple matching failing because of missing foresight
- RepReMatch:  $2n(\log n + 3)$ -approximative

**Phase I** finding enough outstanding items

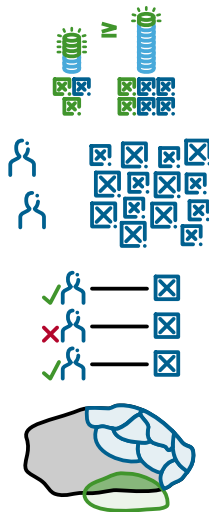


# Summary & Outlook

- allocation  $\hat{=}$  tuple of bundles
- valuation through submodular functions
- Nash social welfare  $\hat{=}$  geometric mean of valuations
- approximation factor independent from  $m$ ?
- simple matching failing because of missing foresight
- RepReMatch:  $2n(\log n + 3)$ -approximative

**Phase I** finding enough outstanding items

**Phase II** assigning remaining items



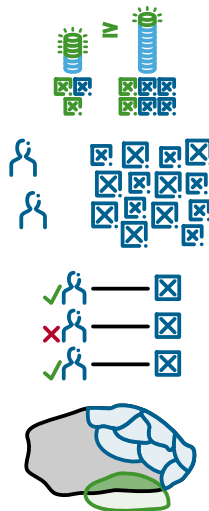
# Summary & Outlook

- allocation  $\hat{=}$  tuple of bundles
- valuation through submodular functions
- Nash social welfare  $\hat{=}$  geometric mean of valuations
- approximation factor independent from  $m$ ?
- simple matching failing because of missing foresight
- RepReMatch:  $2n(\log n + 3)$ -approximative

**Phase I** finding enough outstanding items

**Phase II** assigning remaining items

**Phase III** re-assigning outstanding items



# Summary & Outlook

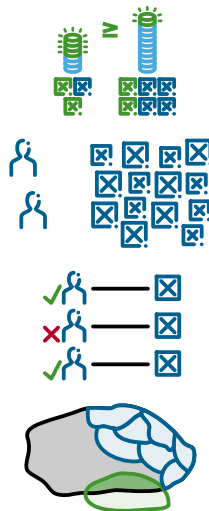
- allocation  $\triangleq$  tuple of bundles
- valuation through submodular functions
- Nash social welfare  $\triangleq$  geometric mean of valuations
- approximation factor independent from  $m$ ?
- simple matching failing because of missing foresight
- RepReMatch:  $2n(\log n + 3)$ -approximative

**Phase I** finding enough outstanding items

**Phase II** assigning remaining items

**Phase III** re-assigning outstanding items

Any room for improvement? Lower bound of  $\frac{e}{e-1} \approx 1.58!$





**End of Talk**

