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

1st August 2023 · Algorithms and Complexity (Prof. Dr Martin Hoefer)

GOETHE UNIVERSITÄT
FRANKFURT AM MAIN

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*.

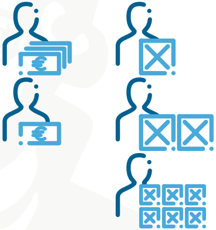
Where is this encountered?

What is the issue?

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

Where is this encountered?

- procurement

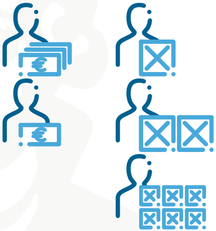


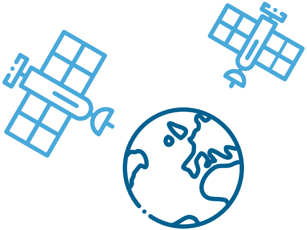
What is the issue?

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

Where is this encountered?

- procurement
- satellites



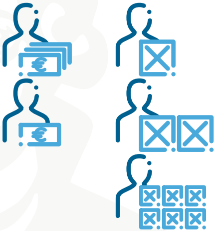


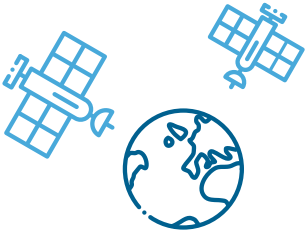
What is the issue?

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

Where is this encountered?

- procurement
- satellites
- water withdrawal





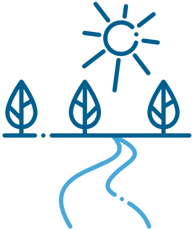


Table of Contents

1 Preliminaries

Allocations

Allocations

Setting:

Allocations

Setting:

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

Allocations

Setting:

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

Allocations

Setting:

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

Definition

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

Allocations

Setting:

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

Definition

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

Allocations

Setting:

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

Definition

An ***allocation*** is a tuple $\boldsymbol{x} = (\boldsymbol{x}_i)_{i\in\mathscr{A}}$ of bundles $\boldsymbol{x}_i \subset \mathscr{G}$ such that each item is element of precisely one bundle.
Item j is ***assigned*** to agent i if $j \in \boldsymbol{x}_i$.

Allocations

Setting:

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

Definition

An ***allocation*** is a tuple $\boldsymbol{x} = (\boldsymbol{x}_i)_{i \in \mathscr{A}}$ of bundles $\boldsymbol{x}_i \subset \mathscr{G}$ such that each item is element of precisely one bundle.
Item j is ***assigned*** to agent i if $j \in \boldsymbol{x}_i$.

But how to measure its efficiency and fairness?

Valuation Functions

Valuation Functions

Requirements:

Valuation Functions

Requirements:

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

Valuation Functions

Requirements:

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

Valuation Functions

Requirements:

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

Types:

Valuation Functions

Requirements:

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

Types:

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

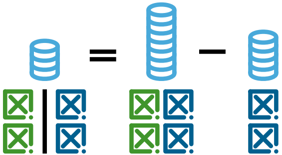
Valuation Functions

Requirements:

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

Types:

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



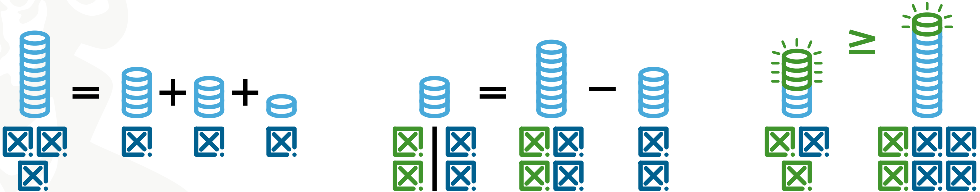
Valuation Functions

Requirements:

- **monotonically non-decreasing:** $v_i(\mathcal{S}_1) \leq v_i(\mathcal{S}_1 \cup \mathcal{S}_2)$
- **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:** $v_i(\mathcal{S}_1 \mid \mathcal{S}_2) \geq v_i(\mathcal{S}_1 \mid \mathcal{S}_2 \cup \mathcal{S}_3)$



Asymmetric Maximum Nash Social Welfare Problem

Asymmetric Maximum Nash Social Welfare Problem

Problem

$$\boldsymbol{x}^* \stackrel{!}{=} \operatorname*{arg\,max}_{\boldsymbol{x} \in \boldsymbol{X}_{\mathscr{A}}(\mathscr{G})} \{\mathrm{NSW}(\boldsymbol{x})\}$$

- $\boldsymbol{X}_{\mathscr{A}}(\mathscr{G})$: all possible allocations

Asymmetric Maximum Nash Social Welfare Problem

Problem

$$\boldsymbol{x}^* \stackrel{!}{=} \arg\max_{\boldsymbol{x} \in X_{\mathscr{A}}(\mathscr{G})} \{\mathrm{NSW}(\boldsymbol{x})\}$$

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

Asymmetric Maximum Nash Social Welfare Problem

Problem

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

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

Asymmetric Maximum Nash Social Welfare Problem

Problem

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

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

The NSW strikes a middle ground between efficiency and fairness!

Asymmetric Maximum Nash Social Welfare Problem

Problem

$$\boldsymbol{x}^* \overset{!}{=} \operatorname*{arg\,max}_{\boldsymbol{x} \in X_{\mathscr{A}}(\mathscr{G})} \{\mathrm{NSW}(\boldsymbol{x})\} \qquad \text{with} \qquad \mathrm{NSW}(\boldsymbol{x}) := \Big(\prod_{i \in \mathscr{A}} v_i(\boldsymbol{x}_i)^{\eta_i}\Big)^{1/\sum_{i \in \mathscr{A}} \eta_i}$$

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

The NSW strikes a middle ground between efficiency and fairness!

Is there an algorithm with an approximation factor …

Asymmetric Maximum Nash Social Welfare Problem

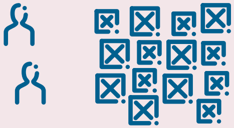
Problem

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

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

The NSW strikes a middle ground between efficiency and fairness!

Is there an algorithm with an approximation factor …



$n \lll m$

Asymmetric Maximum Nash Social Welfare Problem

Problem

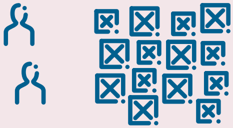
$$x^* \overset{!}{=} \arg\max_{x \in X_{\mathscr{A}}(\mathscr{G})} \{\mathrm{NSW}(x)\} \qquad \text{with} \qquad \mathrm{NSW}(x) := \Big(\prod_{i \in \mathscr{A}} v_i(x_i)^{\eta_i}\Big)^{1/\sum_{i \in \mathscr{A}} \eta_i}$$

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

The NSW strikes a middle ground between efficiency and fairness!

Is there an algorithm with an approximation factor …

- … dependent on n?



$n \lll m$

Asymmetric Maximum Nash Social Welfare Problem

Problem

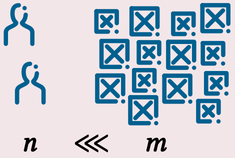
$$x^* \stackrel{!}{=} \arg\max_{x \in X_{\mathscr{A}}(\mathscr{G})} \{\mathrm{NSW}(x)\} \qquad \text{with} \qquad \mathrm{NSW}(x) := \Big(\prod_{i \in \mathscr{A}} v_i(x_i)^{\eta_i}\Big)^{1/\sum_{i \in \mathscr{A}} \eta_i}$$

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

The NSW strikes a middle ground between efficiency and fairness!

Is there an algorithm with an approximation factor …

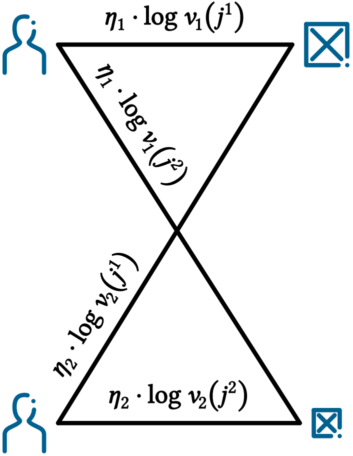
- … dependent on n?
- … independent from m?



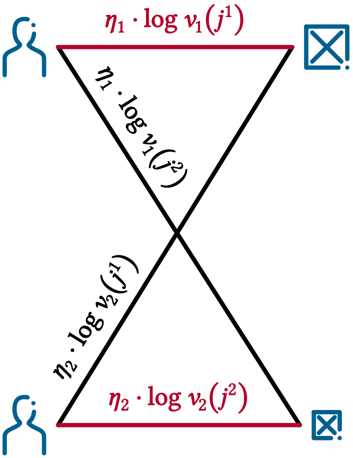
RepReMatch

Naïve Approach

Naïve Approach



Naïve Approach

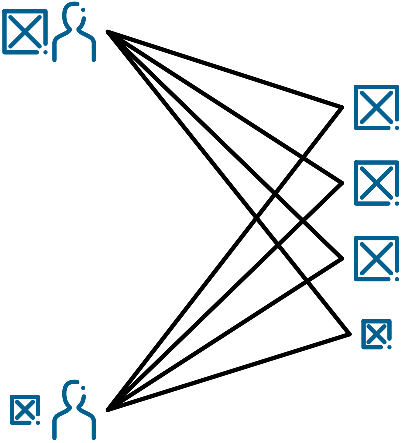


Naïve Approach

Naïve Approach

Greedy algorithm:

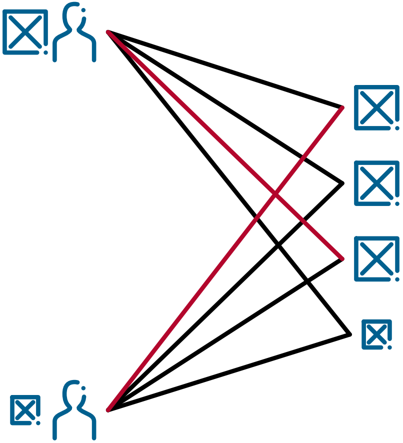
- repeatedly use maximum weight matchings



Naïve Approach

Greedy algorithm:

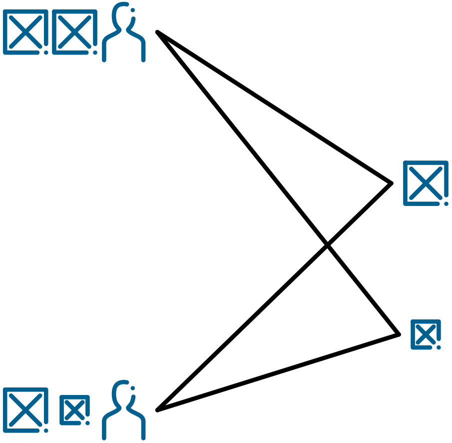
- repeatedly use maximum weight matchings



Naïve Approach

Greedy algorithm:

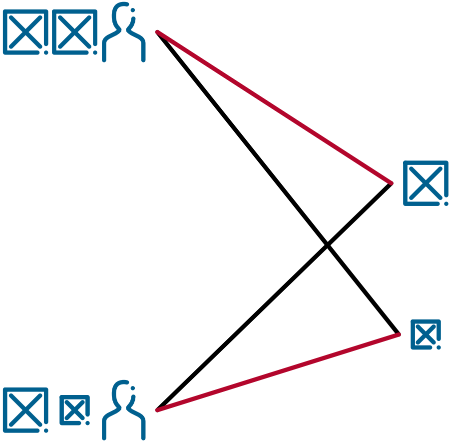
- repeatedly use maximum weight matchings



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 $\Omega(\sqrt{m/\ln m})$

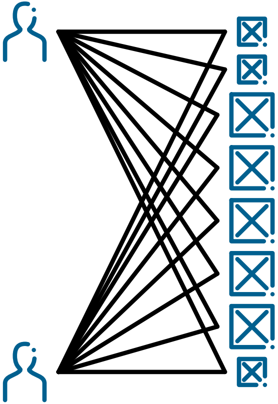
Naïve Approach

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

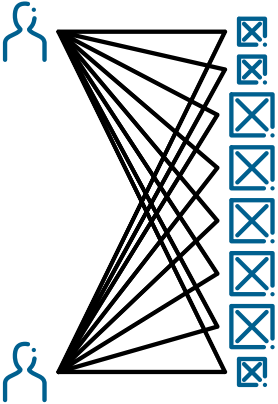
We change the past in three phases:



Key Ideas of the Algorithm

We change the past in three phases:

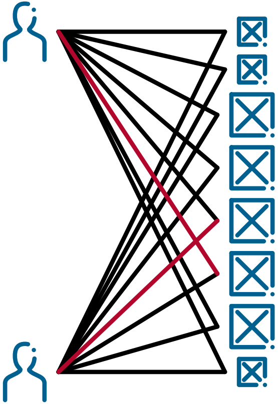
Phase 1 Assign enough high-value items **temporarily**.



Key Ideas of the Algorithm

We change the past in three phases:

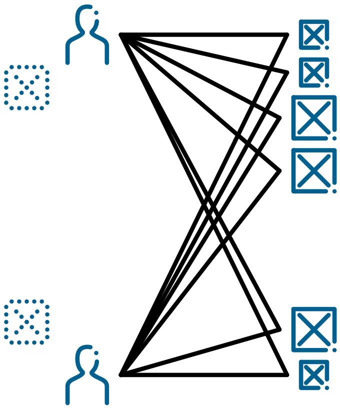
Phase I Assign enough high-value items temporarily.



Key Ideas of the Algorithm

We change the past in three phases:

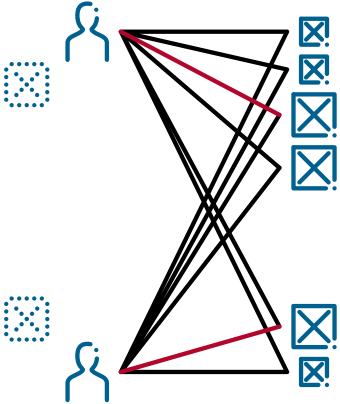
Phase I Assign enough high-value items temporarily.



Key Ideas of the Algorithm

We change the past in three phases:

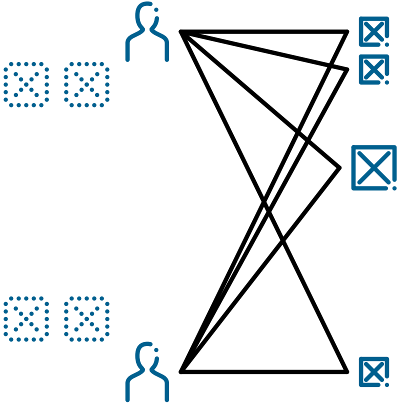
Phase I Assign enough high-value items temporarily.



Key Ideas of the Algorithm

We change the past in three phases:

Phase I Assign enough high-value items temporarily.

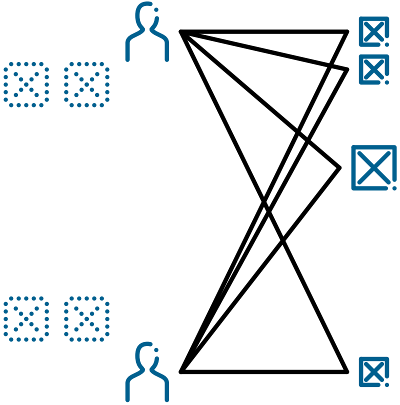


Key Ideas of the Algorithm

We change the past in three phases:

Phase I Assign enough high-value items temporarily.

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

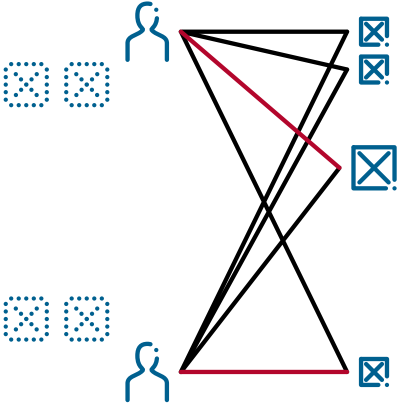


Key Ideas of the Algorithm

We change the past in three phases:

Phase I Assign enough high-value items temporarily.

Phase II Assign the remaining items definitely.

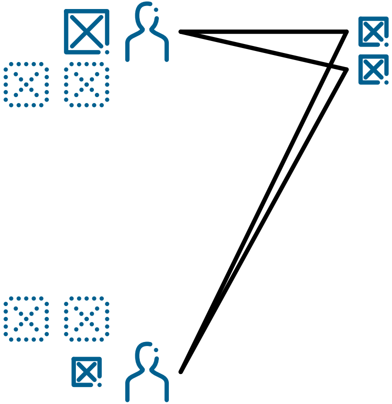


Key Ideas of the Algorithm

We change the past in three phases:

Phase I Assign enough high-value items temporarily.

Phase II Assign the remaining items definitely.

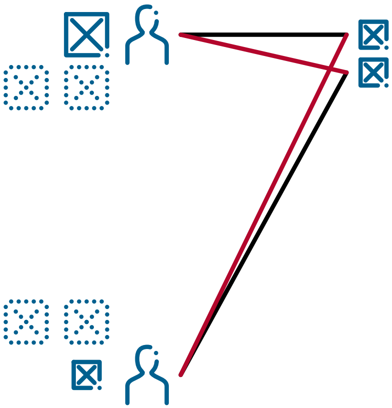


Key Ideas of the Algorithm

We change the past in three phases:

Phase I Assign enough high-value items temporarily.

Phase II Assign the remaining items definitely.

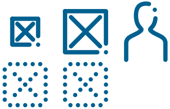


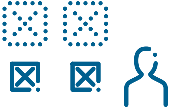
Key Ideas of the Algorithm

We change the past in three phases:

Phase I Assign enough high-value items temporarily.

Phase II Assign the remaining items definitely.





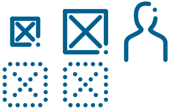
Key Ideas of the Algorithm

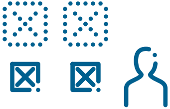
We change the past in three phases:

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**.





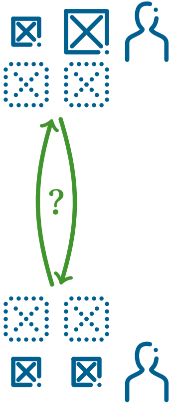
Key Ideas of the Algorithm

We 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.



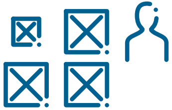
Key Ideas of the Algorithm

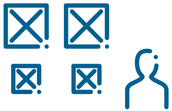
We change the past in three phases:

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.





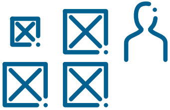
Key Ideas of the Algorithm

We change the past in three phases:

Phase I Assign enough high-value items temporarily.

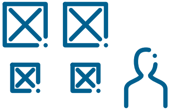
Phase II Assign the remaining items definitely.

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 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 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 with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

The Algorithm

Phase I:

1. repeat $\lceil \log_2 n \rceil + 1$ times
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

Phase II:

The Algorithm

Phase I:

1. repeat $\lceil \log_2 n \rceil + 1$ times
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

Phase II:

2. repeat until all items assigned

The Algorithm

Phase I:

1. repeat $\lceil \log_2 n \rceil + 1$ times
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

Phase II:

2. repeat until all items assigned
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$

The Algorithm

Phase I:

1. repeat $\lceil \log_2 n \rceil + 1$ times
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

Phase II:

2. repeat until all items assigned
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$
   2. compute maximum weight matching

The Algorithm

Phase I:

1. repeat $\lceil \log_2 n \rceil + 1$ times
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

Phase II:

2. repeat until all items assigned
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{II}}$

The Algorithm

Phase I:

1. repeat $\lceil \log_2 n \rceil + 1$ times
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

Phase II:

2. repeat until all items assigned
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{II}}$

Phase III:

The Algorithm

Phase I:

1. repeat $\lceil \log_2 n \rceil + 1$ times
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

Phase II:

2. repeat until all items assigned
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{II}}$

Phase III:

3. release items & create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$

The Algorithm

Phase I:

1. repeat $\lceil \log_2 n \rceil + 1$ times
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

Phase II:

2. repeat until all items assigned
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{II}}$

Phase III:

3. release items & create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$
4. compute maximum weight matching

The Algorithm

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Phase I:

1. repeat $\lceil \log_2 n \rceil + 1$ times
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(j)$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{I}}$

Phase II:

2. repeat until all items assigned
   1. create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$
   2. compute maximum weight matching
   3. update bundles $\boldsymbol{x}_i^{\mathrm{II}}$

Phase III:

3. release items & create bipartite graph with edge weights $\eta_i \cdot \log v_i(\boldsymbol{x}_i^{\mathrm{II}} \cup \{j\})$
4. compute maximum weight matching
5. create bundles $\boldsymbol{x}_i^{\mathrm{III}}$

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 $o_i^1 := \arg\max_{o \in \boldsymbol{x}_i^*} v_i(o)$.

Analysing Phases I & III (1/2)

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

Definition

Let $o_i^1 := \arg\max_{o \in \boldsymbol{x}_i^*} v_i(o)$. 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 $o_i^1 := \arg\max_{o \in \boldsymbol{x}_i^*} v_i(o)$. An item $j \in \mathcal{G}$ is ***outstanding*** if $v_i(j) \geq v_i(o_i^1)$.

$\Rightarrow$ 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 *not matchable* with 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 *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough

Analysing Phases I & III (2/2)

Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

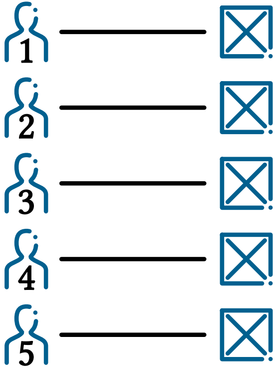
Analysing Phases I & III (2/2)

Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either



Analysing Phases I & III (2/2)

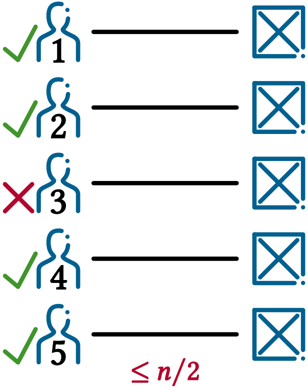
Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$ many agents not matched with outstanding item



Analysing Phases I & III (2/2)

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

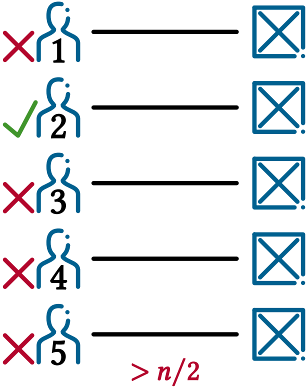
Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$ many agents not matched with outstanding item
- $> n/2$ many agents not matched outstanding item



Analysing Phases I & III (2/2)

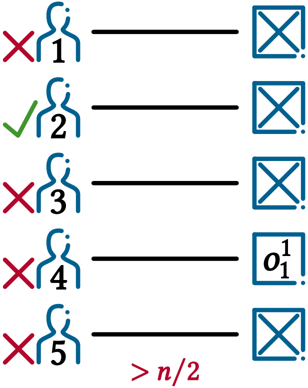
Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$ many agents not matched with outstanding item
- $> n/2$ many agents not matched outstanding item



Analysing Phases I & III (2/2)

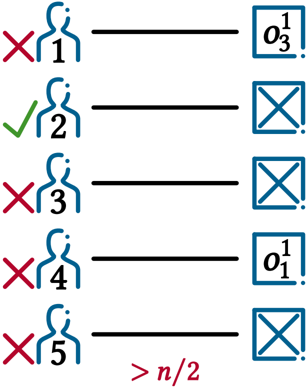
Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
    - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$ many agents not matched with outstanding item
- $> n/2$ many agents not matched outstanding item



Analysing Phases I & III (2/2)

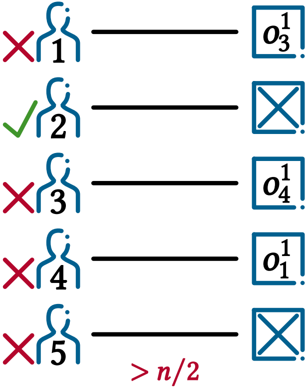
Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$ many agents not matched with outstanding item
- $> n/2$ many agents not matched outstanding item



Analysing Phases I & III (2/2)

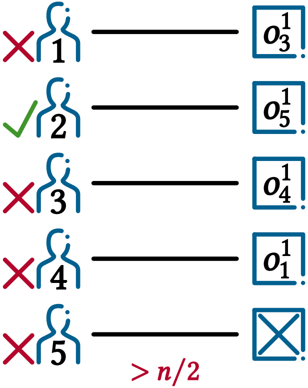
Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$ many agents not matched with outstanding item
- $> n/2$ many agents not matched outstanding item



Analysing Phases I & III (2/2)

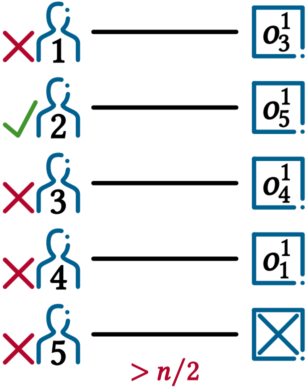
Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$ many agents not matched with outstanding item
- $> n/2$ many agents not matched outstanding item
  - $> n/2$ many items o_i^1 assigned to someone else



Analysing Phases I & III (2/2)

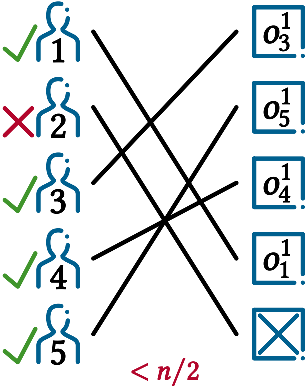
Lemma

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

- maximum number of agents *not matchable* with outstanding item *halved* with each round of phase I
  - $\lceil \log_2 n \rceil + 1$ rounds in phase I enough
- induction on number of rounds in phase I

Base Case: In round 1 of phase I, either

- $\leq n/2$ many agents not matched with outstanding item
- $> n/2$ many agents not matched outstanding item
  - $> n/2$ many items o_i^1 assigned to someone else
  - $> n/2$ many agents matchable with outstanding item upon release in phase III



Analysing Phase II (1/2)

Analysing Phase II (1/2)

Definition

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

Analysing Phase II (1/2)

Definition

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

Analysing Phase II (1/2)

Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \ldots\}$.

Analysing Phase II (1/2)

Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \dots\}$. The set of ***optimal and attainable items*** is defined as

$$\bar{\boldsymbol{x}}_{i,r}^* := \Bigg\{$$

Analysing Phase II (1/2)

Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \dots\}$. The set of ***optimal and attainable items*** is defined as

$$\bar{\boldsymbol{x}}_{i,r}^* := \begin{cases} & \text{in round } r = 1, \end{cases}$$

Analysing Phase II (1/2)

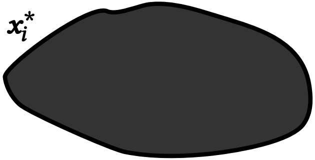
Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \ldots\}$. The set of ***optimal and attainable items*** is defined as

$$\bar{\boldsymbol{x}}_{i,r}^* := \begin{cases} & \text{in round } r = 1, \\ & \end{cases}$$



Analysing Phase II (1/2)

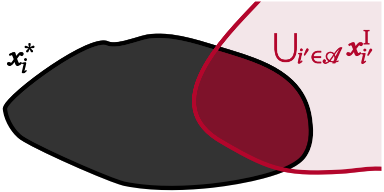
Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \ldots\}$. The set of ***optimal and attainable items*** is defined as

$$\overline{\boldsymbol{x}}_{i,r}^* := \begin{cases} & \text{in round } r = 1, \\ & \end{cases}$$

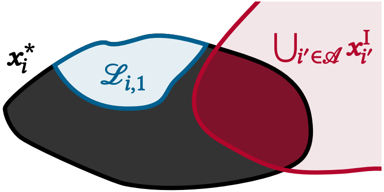


Analysing Phase II (1/2)

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Definition

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



Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \ldots\}$. The set of ***optimal and attainable items*** is defined as

$$\overline{\boldsymbol{x}}_{i,r}^* := \begin{cases} & \text{in round } r = 1, \\ & \end{cases}$$

Analysing Phase II (1/2)

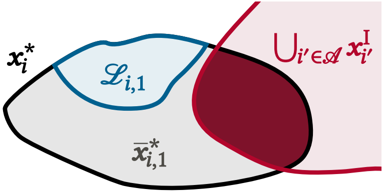
Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \ldots\}$. The set of ***optimal and attainable items*** is defined as

$$\overline{\boldsymbol{x}}_{i,r}^* := \begin{cases} & \text{in round } r = 1, \\ & \end{cases}$$



Analysing Phase II (1/2)

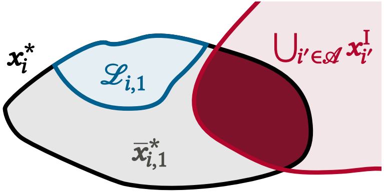
Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \ldots\}$. The set of ***optimal and attainable items*** is defined as

$$\overline{\boldsymbol{x}}_{i,r}^* := \begin{cases} \boldsymbol{x}_i^* \setminus \left(\bigcup_{i' \in \mathscr{A}} \boldsymbol{x}_{i'}^{\mathrm{I}} \cup \mathscr{L}_{i,1}\right) & \text{in round } r = 1, \\ & \end{cases}$$



Analysing Phase II (1/2)

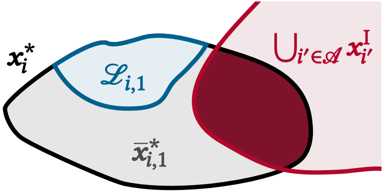
Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \dots\}$. The set of ***optimal and attainable items*** is defined as

$$\overline{\boldsymbol{x}}_{i,r}^* := \begin{cases} \boldsymbol{x}_i^* \setminus \left(\bigcup_{i' \in \mathscr{A}} \boldsymbol{x}_{i'}^{\mathrm{I}} \cup \mathscr{L}_{i,1}\right) & \text{in round } r = 1, \\ & \text{in round } r \geq 2. \end{cases}$$



Analysing Phase II (1/2)

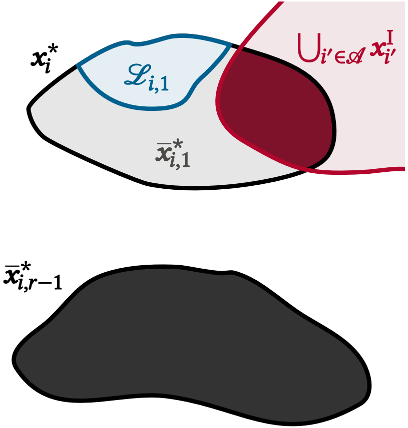
Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \dots\}$. The set of ***optimal and attainable items*** is defined as

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



Analysing Phase II (1/2)

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

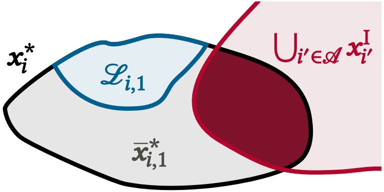
Definition

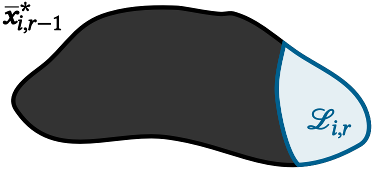
The set $\mathscr{L}_{i,r}$ of ***lost items*** is the set of all optimal items $j \in \boldsymbol{x}_i^*$ matched with other agents $i' \neq i$ in round r.

Definition

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

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





Analysing Phase II (1/2)

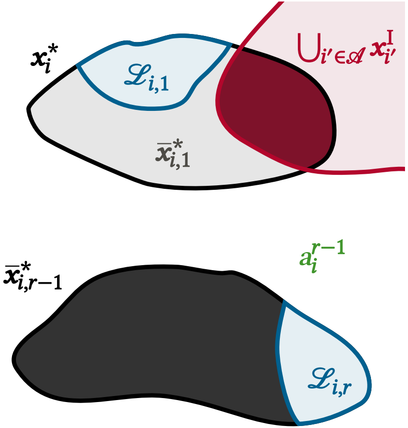
Definition

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

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \dots\}$. The set of ***optimal and attainable items*** is defined as

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



Analysing Phase II (1/2)

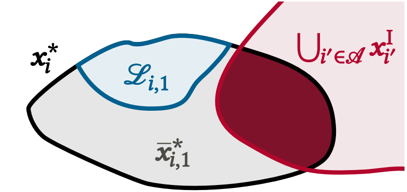
Definition

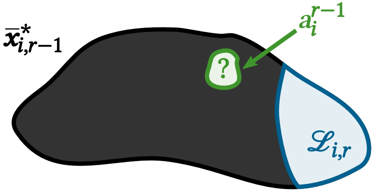
The set $\mathscr{L}_{i,r}$ of ***lost items*** is the set of all optimal items $j \in \boldsymbol{x}_i^*$ matched with other agents $i' \neq i$ in round r.

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \dots\}$. The set of ***optimal and attainable items*** is defined as

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





Analysing Phase II (1/2)

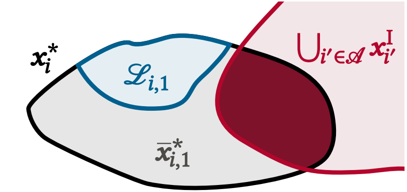
Definition

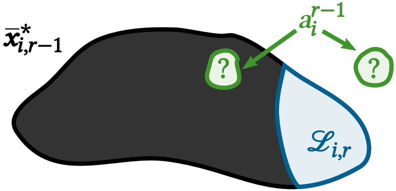
The set $\mathscr{L}_{i,r}$ of ***lost items*** is the set of all optimal items $j \in \boldsymbol{x}_i^*$ matched with other agents $i' \neq i$ in round r.

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \ldots\}$. The set of ***optimal and attainable items*** is defined as

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





Analysing Phase II (1/2)

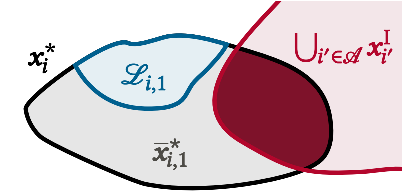
Definition

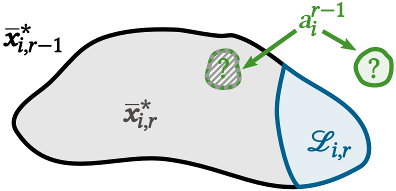
The set $\mathscr{L}_{i,r}$ of ***lost items*** is the set of all optimal items $j \in \boldsymbol{x}_i^*$ matched with other agents $i' \neq i$ in round r.

Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \ldots\}$. The set of ***optimal and attainable items*** is defined as

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

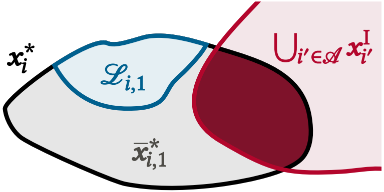




Analysing Phase II (1/2)

Definition

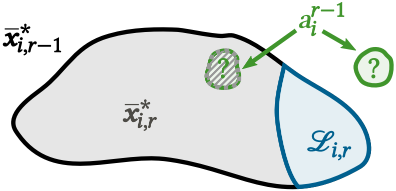
The set $\mathscr{L}_{i,r}$ of ***lost items*** is the set of all optimal items $j \in \boldsymbol{x}_i^*$ matched with other agents $i' \neq i$ in round r.



Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \ldots\}$. The set of ***optimal and attainable items*** is defined as

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



Analysing Phase II (1/2)

Definition

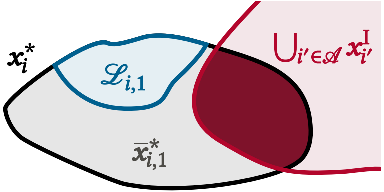
The set $\mathscr{L}_{i,r}$ of ***lost items*** is the set of all optimal items $j \in \boldsymbol{x}_i^*$ matched with other agents $i' \neq i$ in round r.

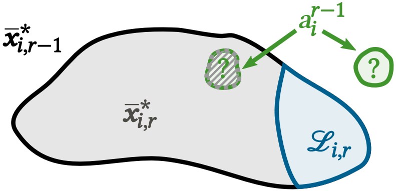
Definition

Let $\boldsymbol{x}_i^{\mathrm{II}} = \{a_i^1, a_i^2, \dots\}$. The set of ***optimal and attainable items*** is defined as

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

$\Rightarrow$ What is the valuation of the remaining items?





Analysing Phase II (2/2)

Lemma

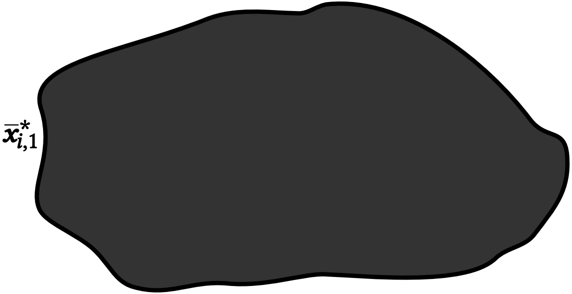
$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \qquad \text{if } r \geq 2$$

RepReMatch

Analysing Phase II (2/2)

Lemma

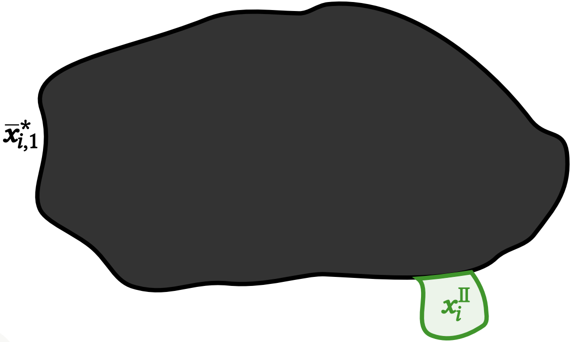
$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

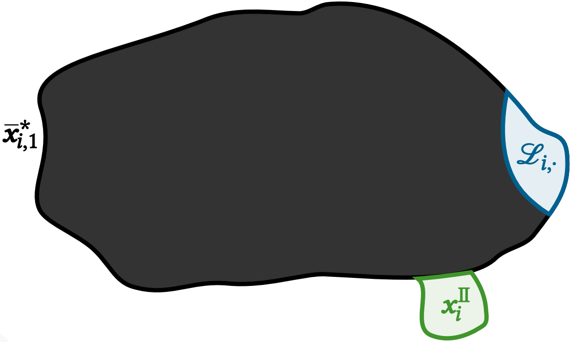
$v_i\left(\bar{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

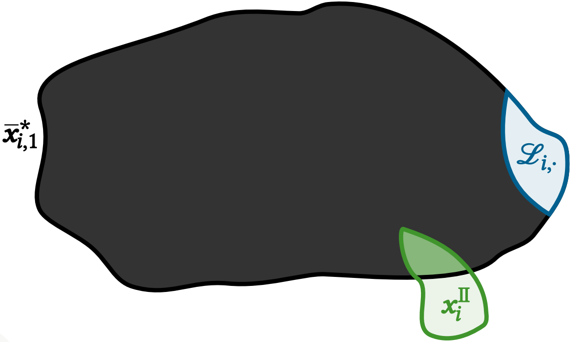
$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

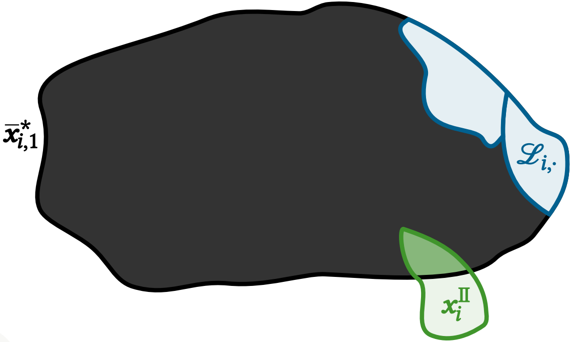
$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

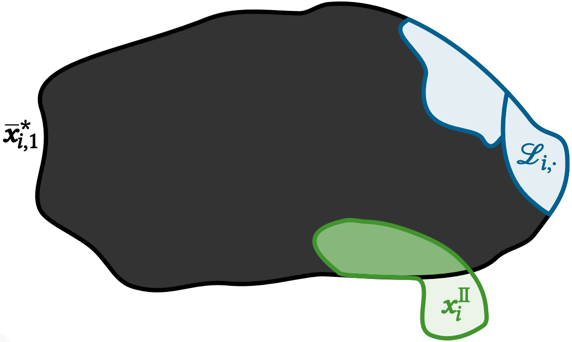
$$v_i\left(\bar{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

$$v_i\left(\bar{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) \qquad \text{if } r \geq 2$$

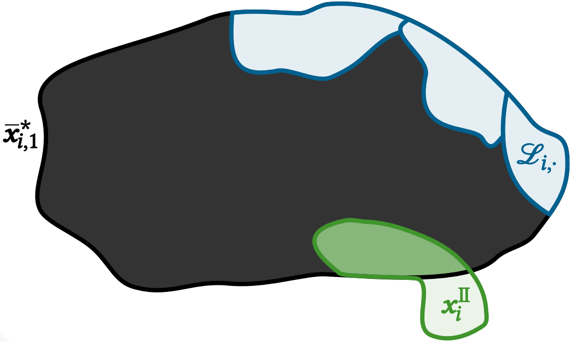


Analysing Phase II (2/2)

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Lemma

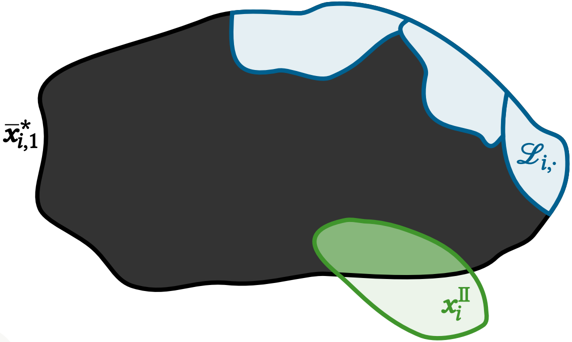
$$v_i\left(\bar{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

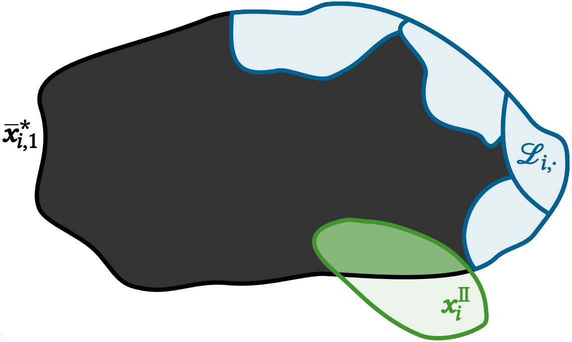
$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

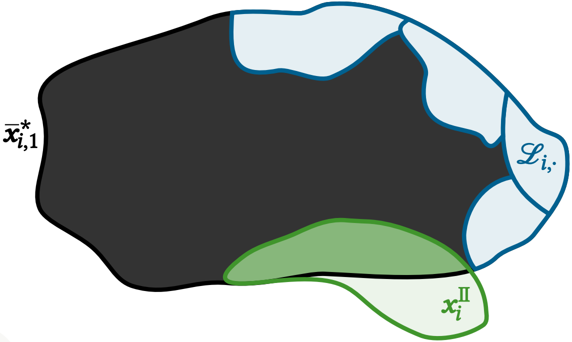
$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

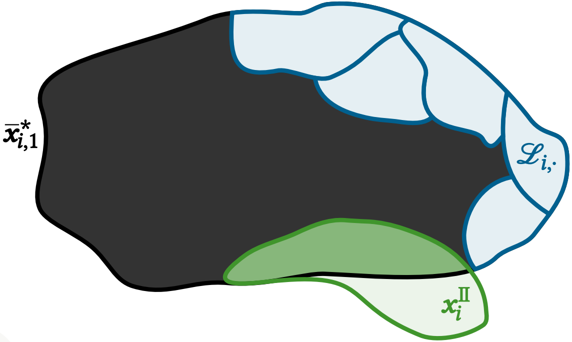
$$v_i\left(\bar{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

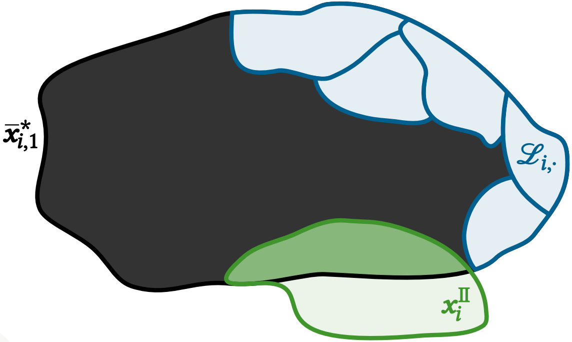
$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

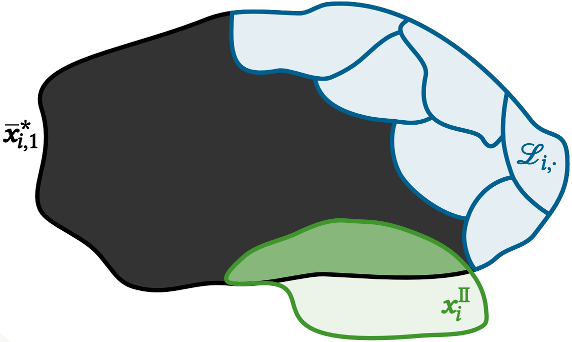
$v_i\left(\bar{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

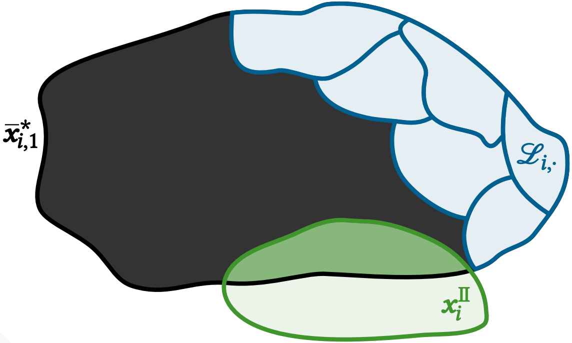
$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

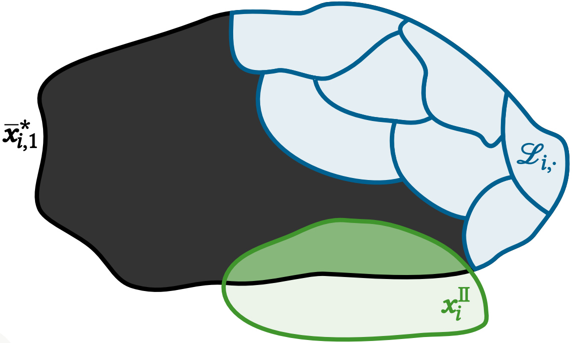
$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

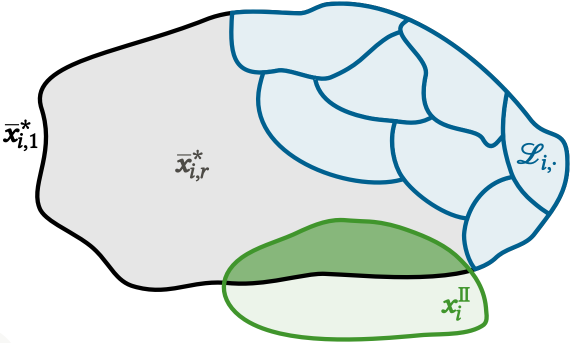
$v_i\left(\bar{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

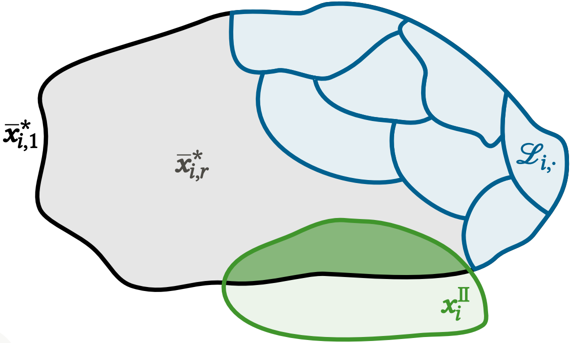
$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right)$ if $r \geq 2$



Analysing Phase II (2/2)

Lemma

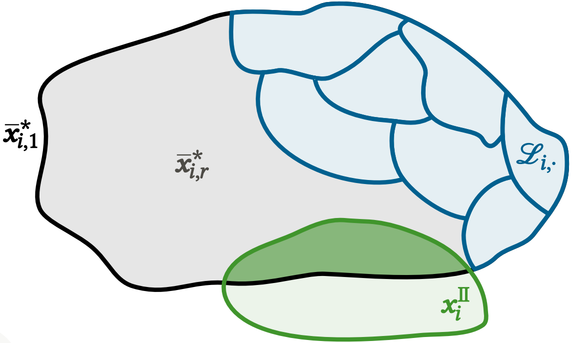
$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) = v_i\left(\overline{x}^*_{i,r} \cup \left\{a_i^1, \dots, a_i^{r-1}\right\}\right) - v_i\left(a_i^1, \dots, a_i^{r-1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

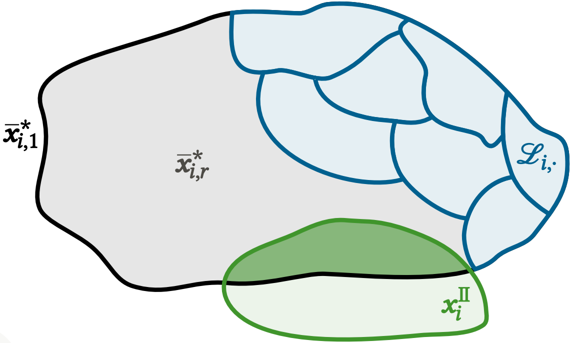
$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) = v_i\left(\overline{x}^*_{i,r} \cup \{a_i^1, \ldots, a_i^{r-1}\}\right) - v_i\left(a_i^1, \ldots, a_i^{r-1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

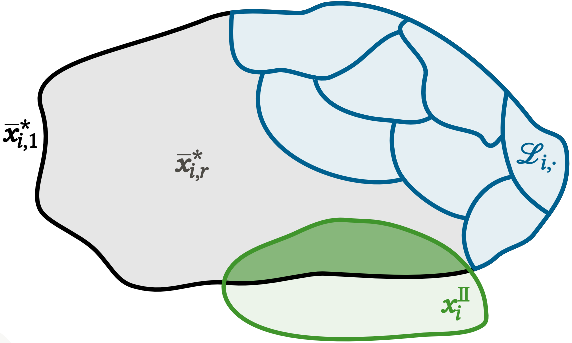
$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) = -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\overline{x}^*_{i,r} \cup \{a_i^1, \dots, a_i^{r-1}\}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

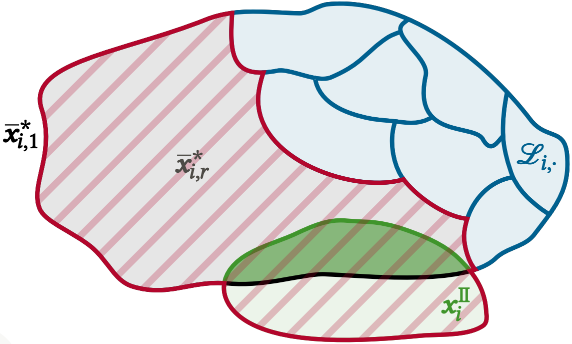
$$v_i\left(\bar{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) = -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\bar{x}^*_{i,r} \cup \{a_i^1, \dots, a_i^{r-1}\}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

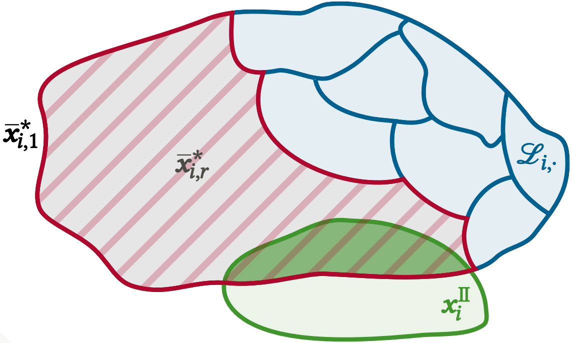
$$v_i\left(\bar{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) = -v_i\left(a_i^1, \ldots, a_i^{r-1}\right) + v_i\left(\bar{x}^*_{i,r} \cup \{a_i^1, \ldots, a_i^{r-1}\}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

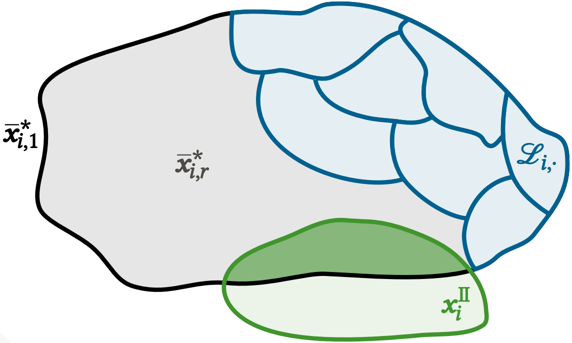
$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) = -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\overline{x}^*_{i,r} \cup \left\{a_i^1, \dots, a_i^{r-1}\right\}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

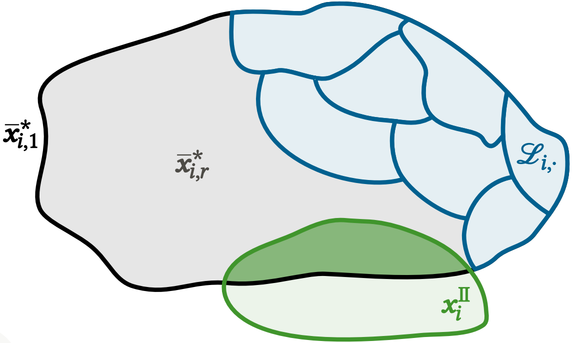
$$v_i\left(\overline{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \ldots, a_i^{r-1}\right) + v_i\left(\overline{\boldsymbol{x}}^*_{i,1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

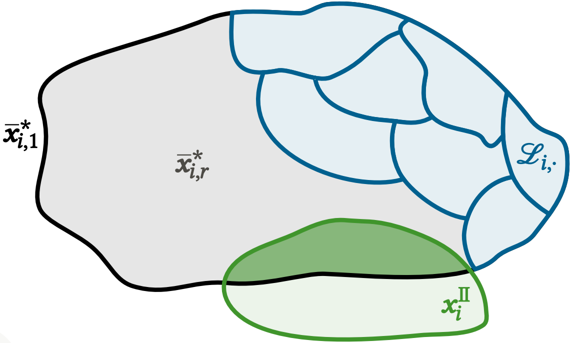
$$v_i\left(\bar{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \ldots, a_i^{r-1}\right) + v_i\left(\bar{x}^*_{i,1}\right) - v_i\left(\mathscr{L}_{i,2} \mid a_i^1\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

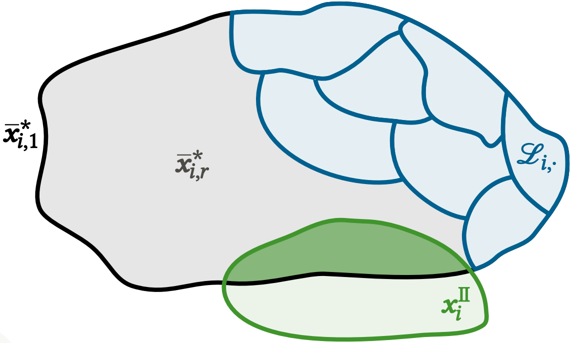
$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\overline{x}^*_{i,1}\right) - v_i\left(\mathscr{L}_{i,2} \mid a_i^1\right) - v_i\left(\mathscr{L}_{i,3} \mid a_i^1, a_i^2\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

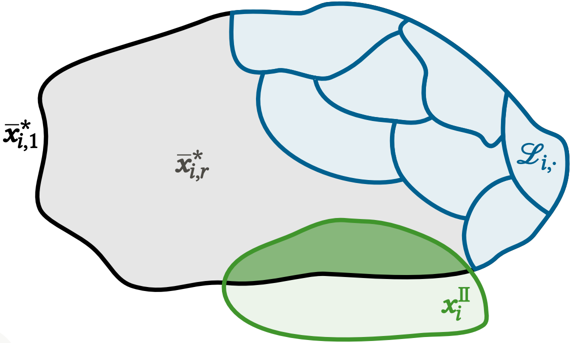
$$v_i\left(\bar{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\bar{\boldsymbol{x}}^*_{i,1}\right) - v_i\left(\mathscr{L}_{i,2} \mid a_i^1\right) - v_i\left(\mathscr{L}_{i,3} \mid a_i^1, a_i^2\right) - \dots \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \ldots, a_i^{r-1}\right) + v_i\left(\overline{x}^*_{i,1}\right) - \sum_{l=2}^{r} v_i\left(\mathscr{L}_{i,l} \mid a_i^1, \ldots, a_i^{l-1}\right) \qquad \text{if } r \geq 2$$

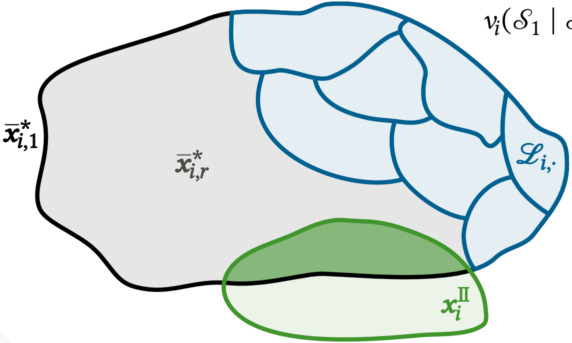


Analysing Phase II (2/2)

Lemma

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

$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) =$

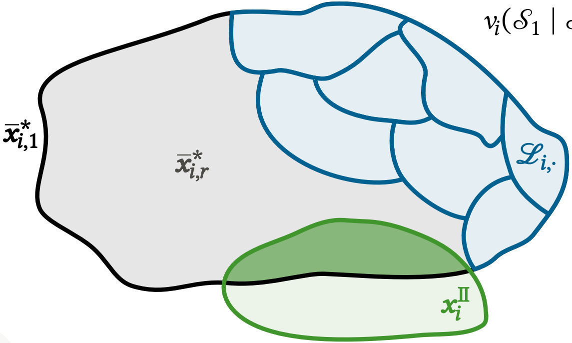


Analysing Phase II (2/2)

Lemma

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

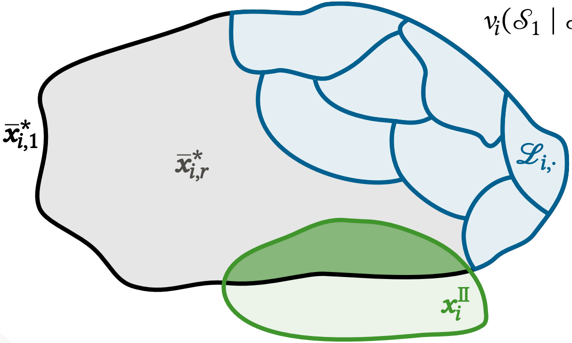
$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) = v_i\left(j^1 \mid \mathcal{S}_2\right)$$



Analysing Phase II (2/2)

Lemma

$$v_i\left(\overline{\boldsymbol{x}}_{i,r}^* \mid a_i^1,\dots,a_i^{r-1}\right) \geq -v_i\left(a_i^1,\dots,a_i^{r-1}\right) + v_i\left(\overline{\boldsymbol{x}}_{i,1}^*\right) - \sum_{l=2}^{r} v_i\left(\mathscr{L}_{i,l} \mid a_i^1,\dots,a_i^{l-1}\right) \qquad \text{if } r \geq 2$$

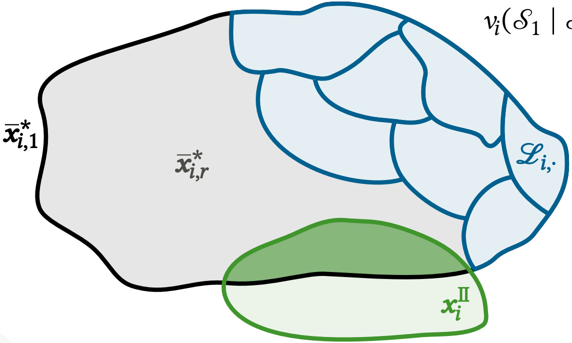


$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) = v_i\left(j^1 \mid \mathcal{S}_2\right) + v_i\left(j^2 \mid \mathcal{S}_2 \cup \{j^1\}\right)$$

Analysing Phase II (2/2)

Lemma

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



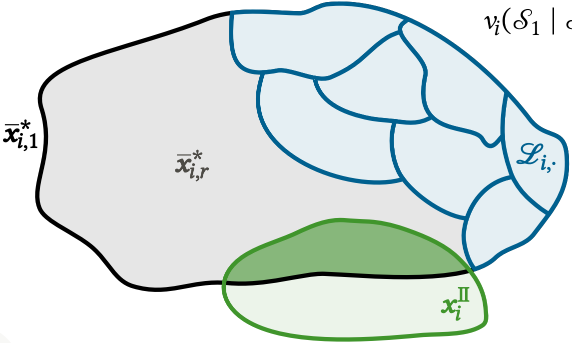
$$\begin{aligned} v_i(\mathcal{S}_1 \mid \mathcal{S}_2) = {} & v_i\left(j^1 \mid \mathcal{S}_2\right) + \\ & v_i\left(j^2 \mid \mathcal{S}_2 \cup \{j^1\}\right) + \\ & v_i\left(j^3 \mid \mathcal{S}_2 \cup \{j^1, j^2\}\right) \end{aligned}$$

Analysing Phase II (2/2)

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Lemma

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

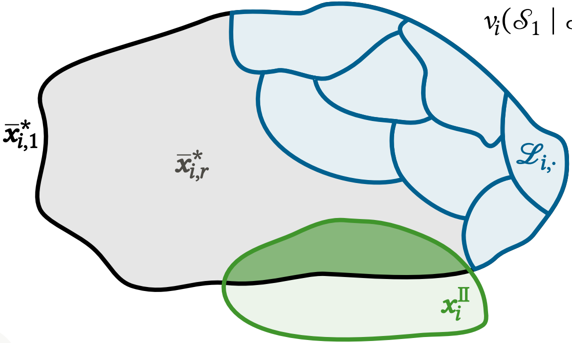


$$\begin{aligned} v_i(\mathcal{S}_1 \mid \mathcal{S}_2) = {} & v_i\left(j^1 \mid \mathcal{S}_2\right) + \\ & v_i\left(j^2 \mid \mathcal{S}_2 \cup \{j^1\}\right) + \\ & v_i\left(j^3 \mid \mathcal{S}_2 \cup \{j^1, j^2\}\right) + \\ & \vdots \end{aligned}$$

Analysing Phase II (2/2)

Lemma

$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \ldots, a_i^{r-1}\right) + v_i\left(\overline{x}^*_{i,1}\right) - \sum_{l=2}^{r} v_i\left(\mathscr{L}_{i,l} \mid a_i^1, \ldots, a_i^{l-1}\right) \qquad \text{if } r \geq 2$$

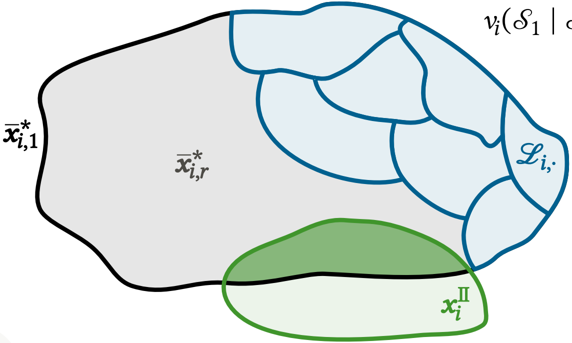


$$\begin{aligned} v_i(\mathcal{S}_1 \mid \mathcal{S}_2) = {} & v_i\left(j^1 \mid \mathcal{S}_2\right) + \\ & v_i\left(j^2 \mid \mathcal{S}_2 \cup \{j^1\}\right) + \\ & v_i\left(j^3 \mid \mathcal{S}_2 \cup \{j^1, j^2\}\right) + \\ & \vdots \end{aligned}$$

Analysing Phase II (2/2)

Lemma

$$v_i\left(\overline{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\overline{\boldsymbol{x}}^*_{i,1}\right) - \sum_{l=2}^{r} v_i\left(\mathscr{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}\right) \qquad \text{if } r \geq 2$$



$$\begin{aligned} v_i(\mathcal{S}_1 \mid \mathcal{S}_2) \leq\ & v_i\left(j^1 \mid \mathcal{S}_2\right) + \\ & v_i\left(j^2 \mid \mathcal{S}_2\right) + \\ & v_i\left(j^3 \mid \mathcal{S}_2\right) + \\ & \vdots \end{aligned}$$

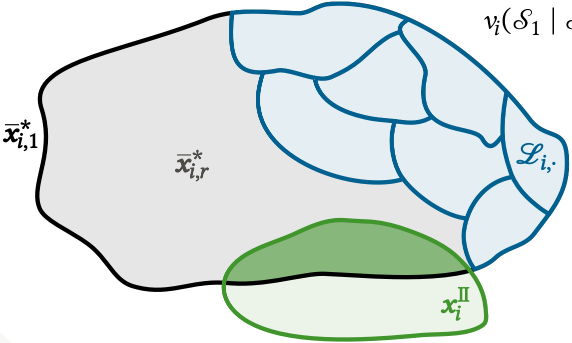
Analysing Phase II (2/2)

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Lemma

$$v_i\left(\overline{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\overline{\boldsymbol{x}}^*_{i,1}\right) - \sum_{l=2}^{r} v_i\left(\mathscr{L}_{i,l} \mid a_i^1, \dots, a_i^{l-1}\right) \qquad \text{if } r \geq 2$$

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

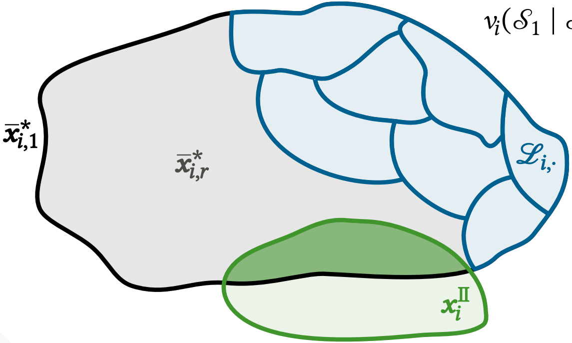


Analysing Phase II (2/2)

Lemma

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

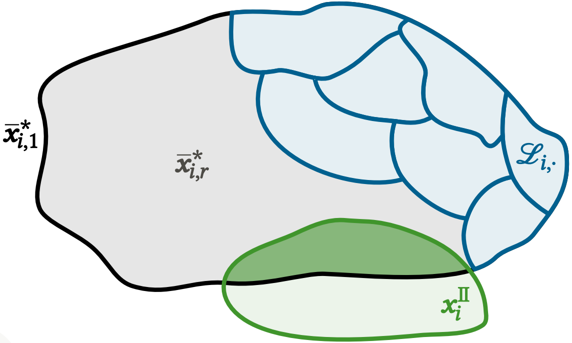
$$v_i(\mathcal{S}_1 \mid \mathcal{S}_2) \leq \sum_{j \in \mathcal{S}_1} v_i(j \mid \mathcal{S}_2)$$



Analysing Phase II (2/2)

Lemma

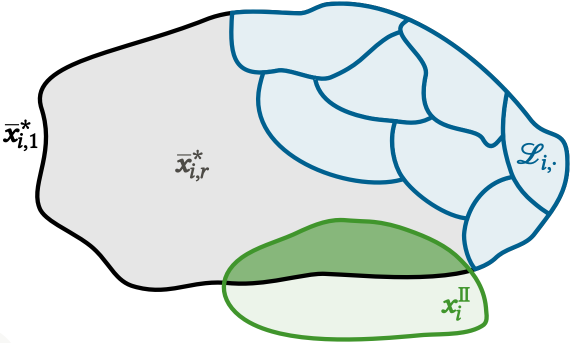
$$v_i\left(\overline{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\overline{\boldsymbol{x}}^*_{i,1}\right) - \sum_{l=2}^{r} \sum_{j \in \mathscr{L}_{i,l}} v_i\left(j \mid a_i^1, \dots, a_i^{l-1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

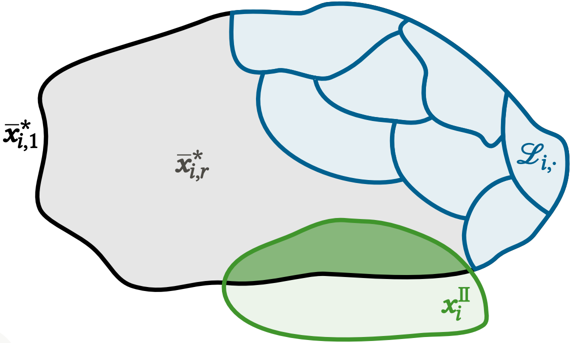
$$v_i\left(\overline{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\overline{x}^*_{i,1}\right) - \sum_{l=2}^{r} \sum_{j \in \mathcal{L}_{i,l}} v_i\left(j \mid a_i^1, \dots, a_i^{l-1}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

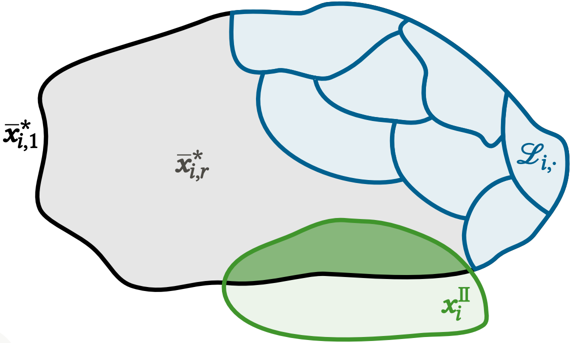
$$v_i\left(\bar{x}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \ldots, a_i^{r-1}\right) + v_i\left(\bar{x}^*_{i,1}\right) - \sum_{l=2}^{r} \sum_{j \in \mathscr{L}_{i,l}} v_i\left(j \mid a_i^1, \ldots, a_i^{l-2}\right) \quad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

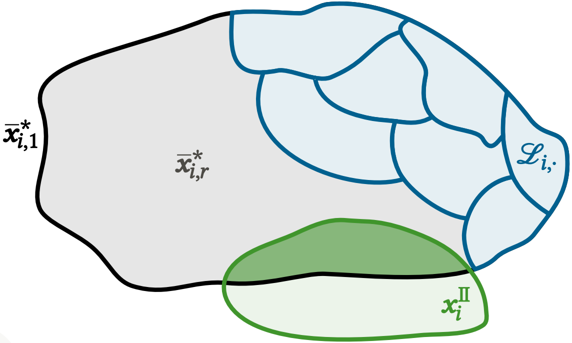
$$v_i\left(\bar{x}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\bar{x}^*_{i,1}\right) - \sum_{l=2}^{r} \sum_{j \in \mathscr{L}_{i,l}} v_i\left(j \mid a_i^1, \dots, a_i^{l-2}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

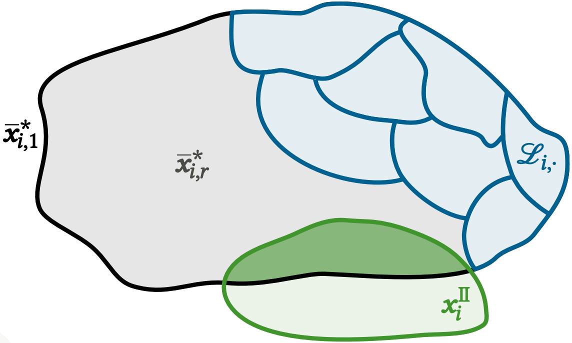
$$v_i\left(\bar{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \ldots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \ldots, a_i^{r-1}\right) + v_i\left(\bar{\boldsymbol{x}}^*_{i,1}\right) - \sum_{l=2}^{r} \sum_{j \in \mathscr{L}_{i,l}} v_i\left(a_i^{l-1} \mid a_i^1, \ldots, a_i^{l-2}\right) \qquad \text{if } r \geq 2$$



Analysing Phase II (2/2)

Lemma

$$v_i\left(\overline{\boldsymbol{x}}^*_{i,r} \mid a_i^1, \dots, a_i^{r-1}\right) \geq -v_i\left(a_i^1, \dots, a_i^{r-1}\right) + v_i\left(\overline{\boldsymbol{x}}^*_{i,1}\right) - \sum_{l=2}^{r} |\mathscr{L}_{i,l}| \cdot v_i\left(a_i^{l-1} \mid a_i^1, \dots, a_i^{l-2}\right) \qquad \text{if } r \geq 2$$



Conclusion

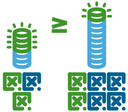
Summary & Outlook

Summary & Outlook

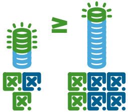
- allocation ≙ tuple of bundles

Summary & Outlook

- allocation ≙ tuple of bundles
- valuation through submodular functions



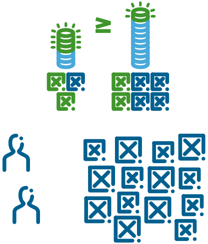
Summary & Outlook



- allocation ≙ tuple of bundles
- valuation through submodular functions
- Nash social welfare ≙ geometric mean of valuations

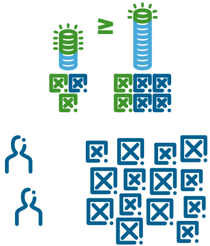
Summary & Outlook

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



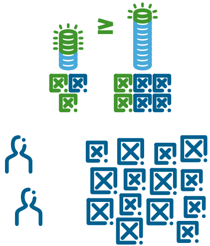
Summary & Outlook

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



Summary & Outlook

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

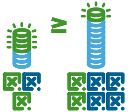


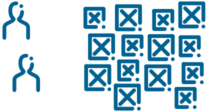
Conclusion

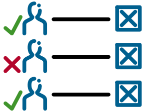
Summary & Outlook

- allocation ≙ tuple of bundles
- valuation through submodular functions
- Nash social welfare ≙ 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



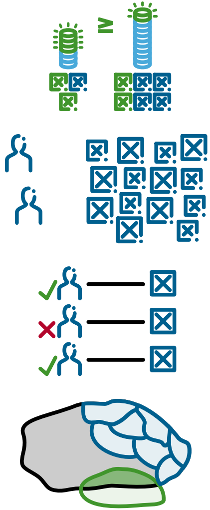




Summary & Outlook

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

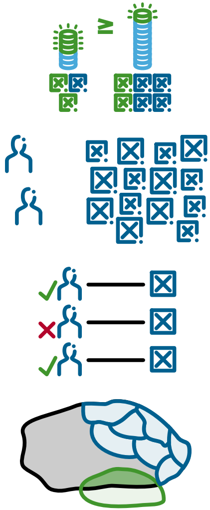
Phase I finding enough outstanding items
Phase II assigning remaining items



Summary & Outlook

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

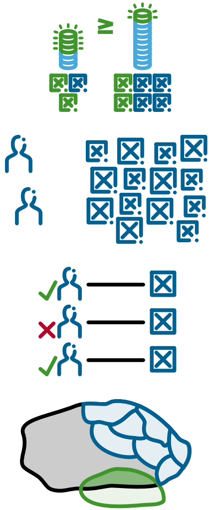
  Phase I finding enough outstanding items
  Phase II assigning remaining items
  Phase III re-assigning outstanding items



Summary & Outlook

- allocation ≙ tuple of bundles
- valuation through submodular functions
- Nash social welfare ≙ 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

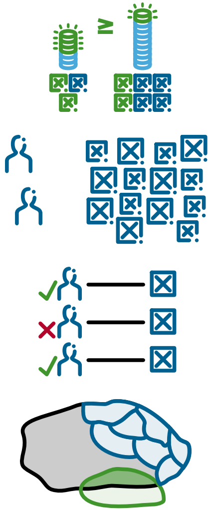


Any room for improvement?

Summary & Outlook

- allocation ≙ tuple of bundles
- valuation through submodular functions
- Nash social welfare ≙ 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