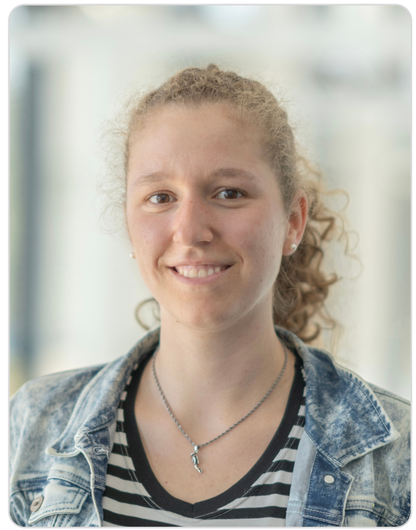
Optimal Metric Distortion for Learning-Augmented Matchings on the Line



Jabari Hastings and **Marena Richter**

WINE, 1–4 December 2026, Hong Kong



Motivation – School Choice Problem







Motivation – School Choice Problem













Motivation – School Choice Problem





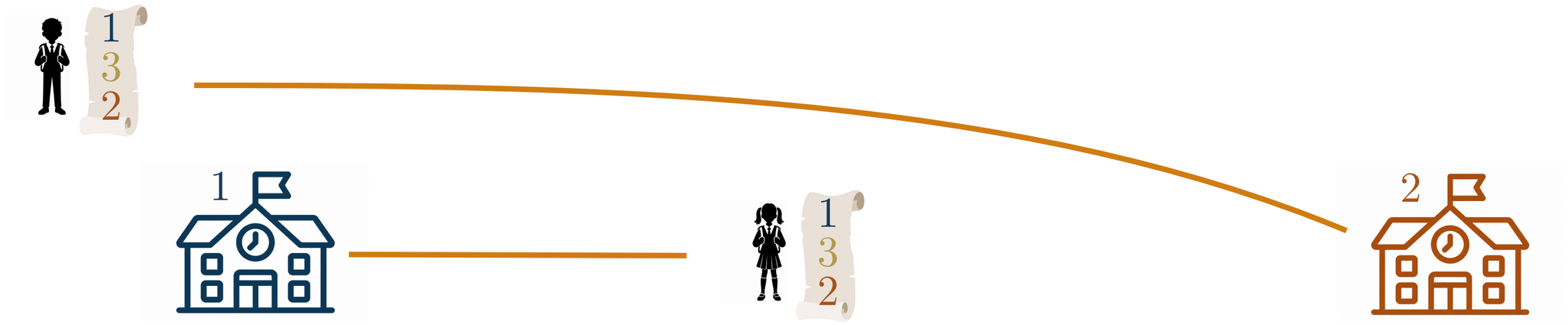








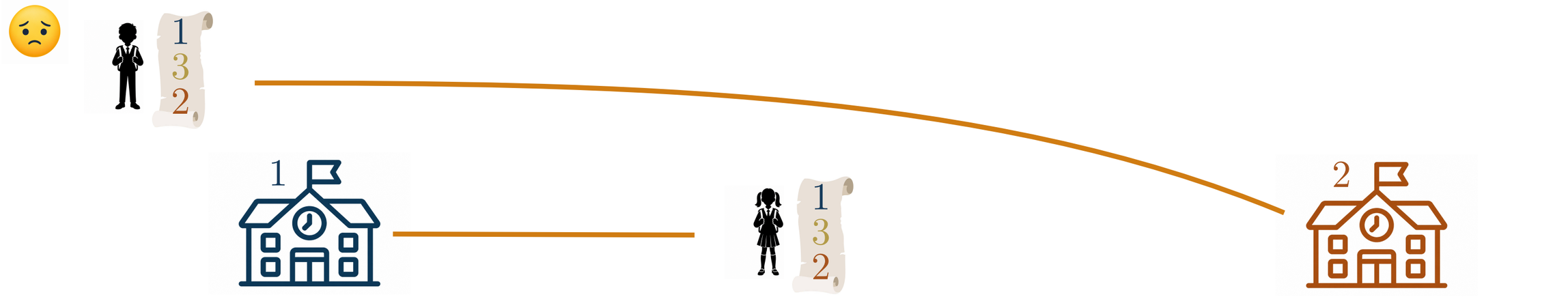
Motivation – School Choice Problem







Motivation – School Choice Problem







Motivation – School Choice Problem









St. Petersburg Times

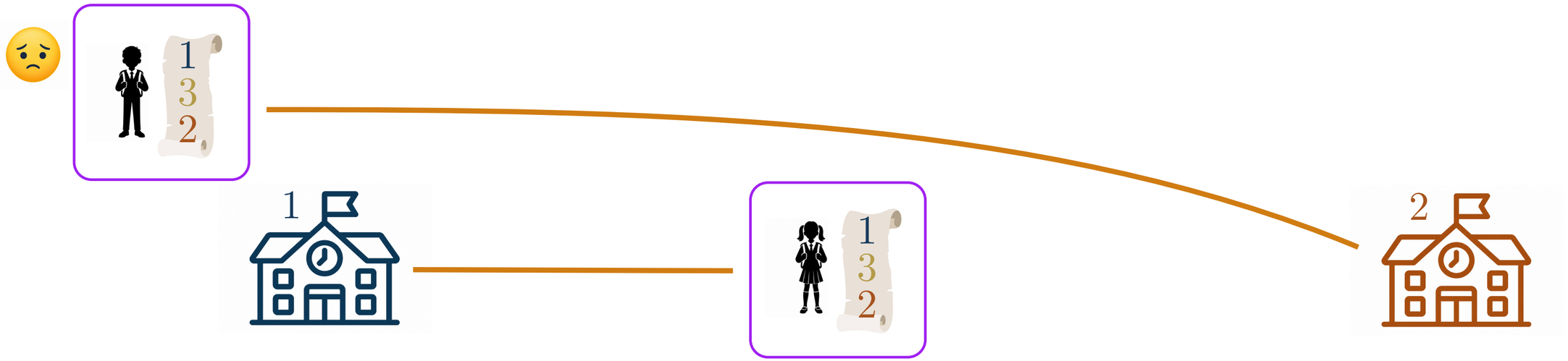
School choice

"This complicated new system requires you to be a savvy parent. Some parents have treated the choice application with a casualness befitting a form for summer camp. Yet it requires the same level of care you would give an IRS tax return."

Thomas Tobin — St. Petersburg Times, 2003



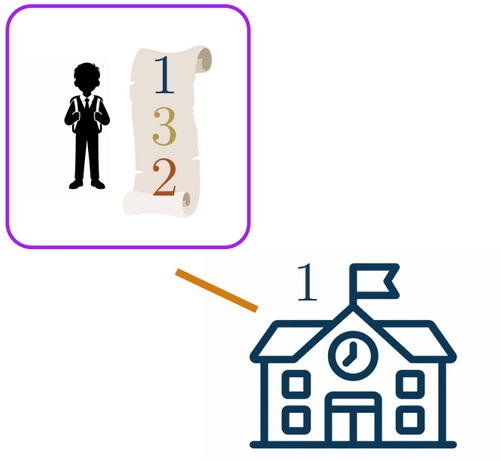
Motivation – School Choice Problem



Rankings: Loose information



Motivation – School Choice Problem





Rankings: Loose information



Motivation – School Choice Problem









Rankings: Loose information

How can we do better?





Motivation – Predictions

Input:

Rankings







+

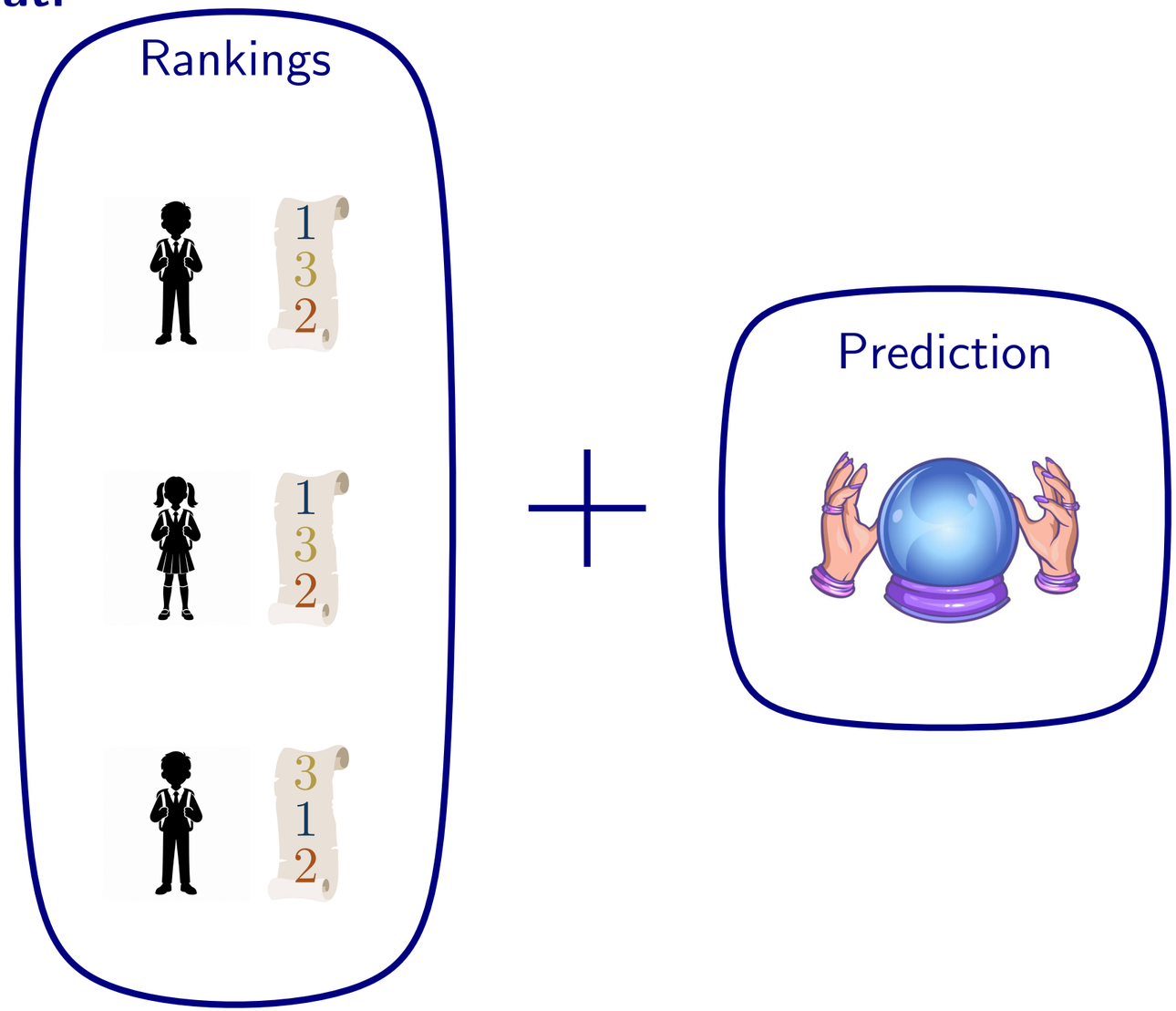
Prediction





Motivation – Predictions

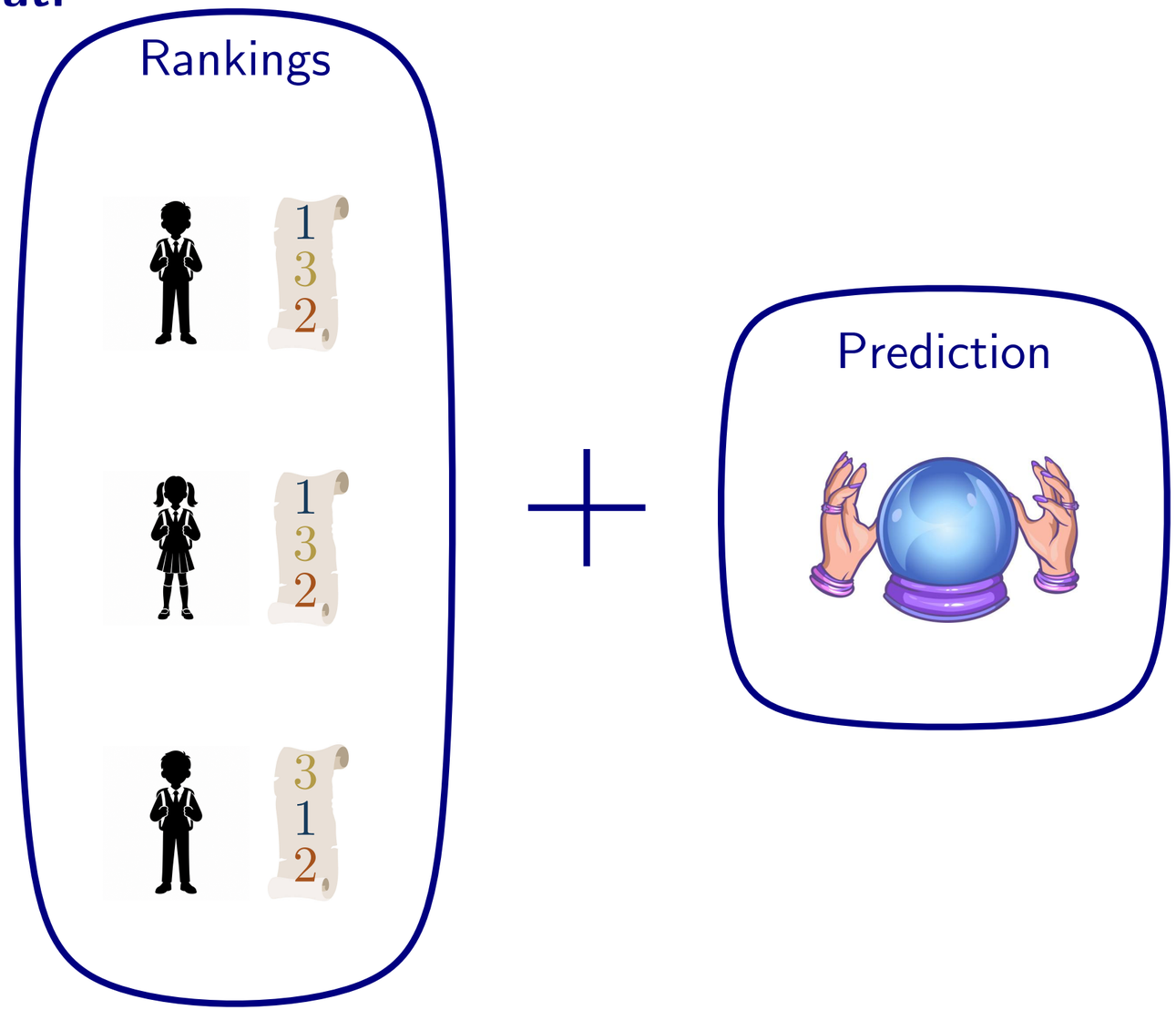
Input:



- Bring predictions to matching world

Motivation – Predictions

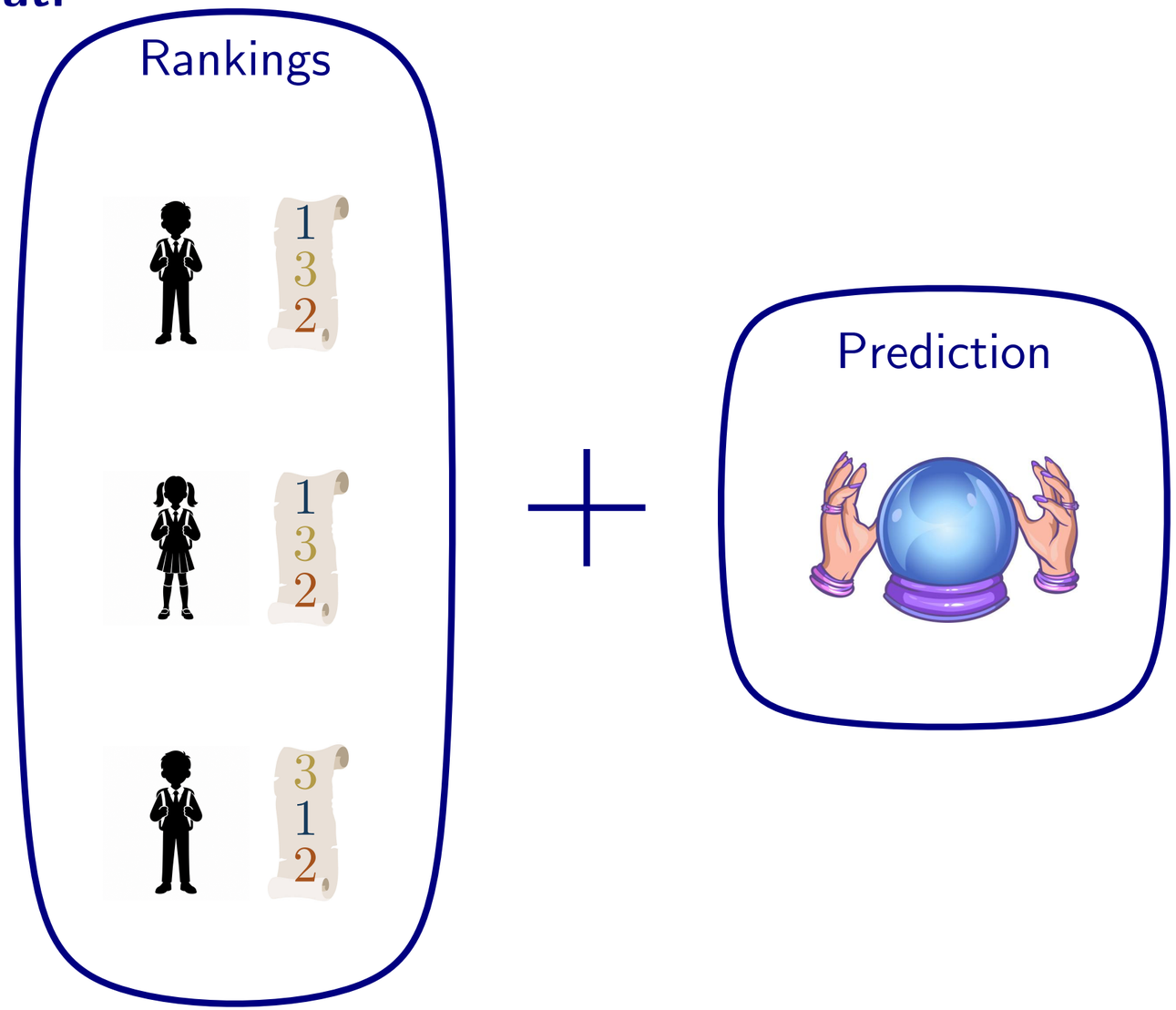
Input:



- Bring predictions to matching world
- Can additional data help?

Motivation – Predictions

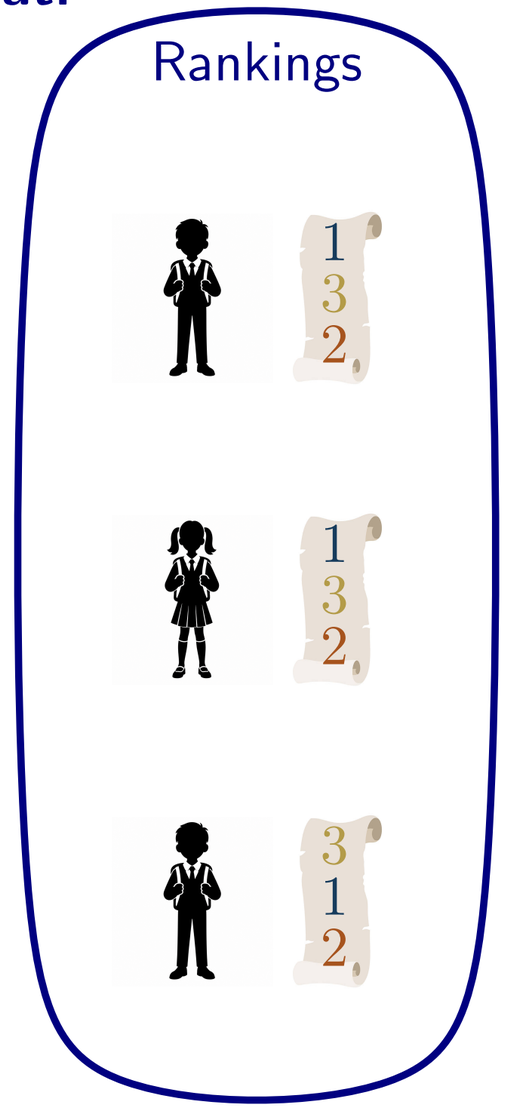
Input:



- Bring predictions to matching world
- Can additional data help?
- Based on past data

Motivation – Predictions

Input:





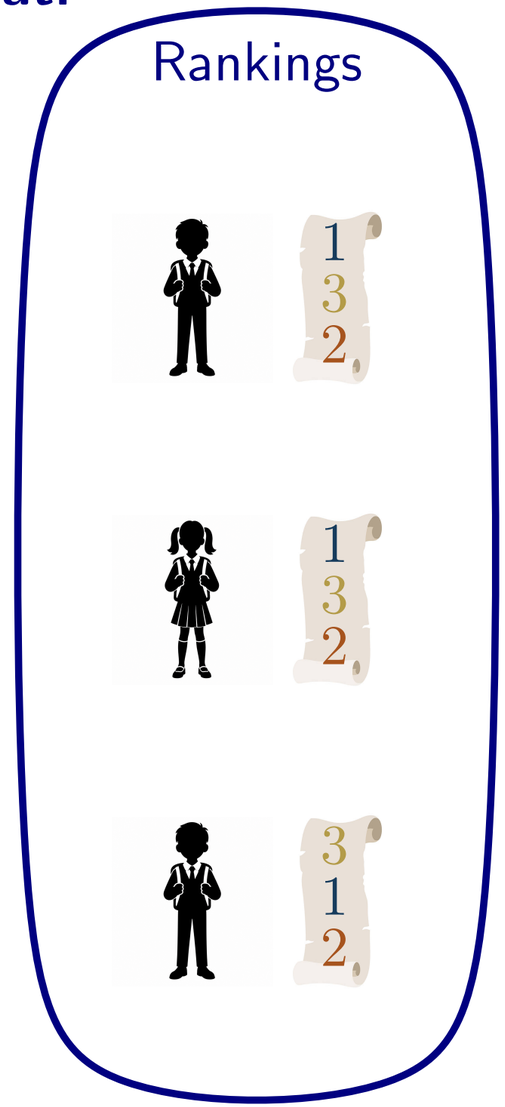


- Bring predictions to matching world
- Can additional data help?
- Based on past data
- Much data, unknown quality



Motivation – Predictions

Input:







- Bring predictions to matching world
- Can additional data help?
- Based on past data
- Much data, unknown quality

Proof of concept for matchings



line instances

Motivation – Predictions

Input:

Rankings







+

Prediction





Two goals for predictions



Consistency

With correct advice:
how close to optimum?

Motivation – Predictions

Input:

Rankings







+

Prediction



Two goals for predictions



Consistency

With correct advice:
how close to optimum?

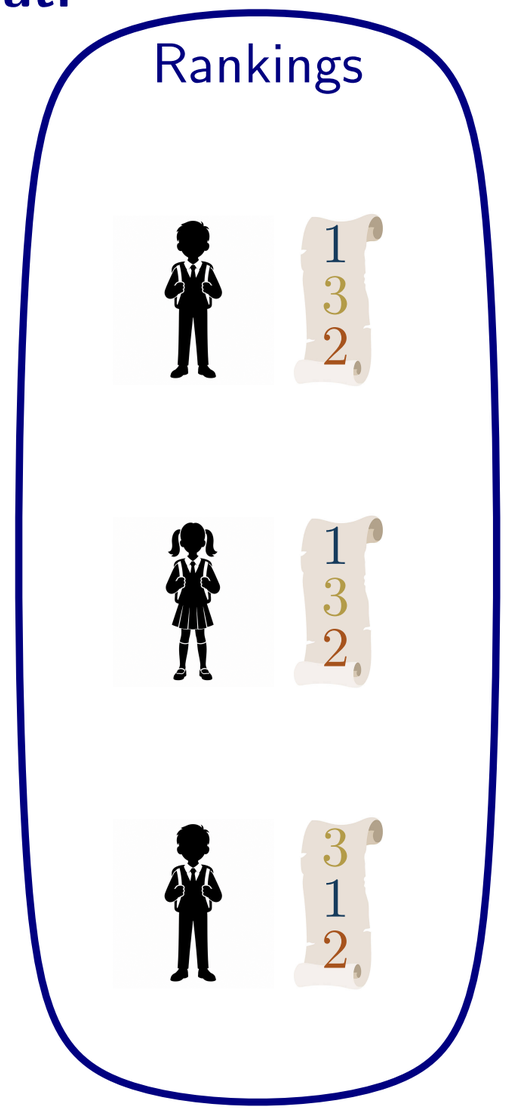


Robustness

With arbitrary advice:
what cost guarantee?

Motivation – Predictions

Input:







Two goals for predictions



Consistency

With correct advice:
how close to optimum?



Robustness

With arbitrary advice:
what cost guarantee?

Goal: optimal with perfect advice, safe even when the advice fails.

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

Problem Definition

Instance:

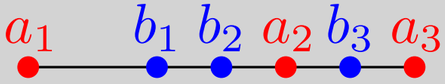
n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a metric instance

d_σ : 

here: a line

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1
a_2 —— b_2
a_3 —— b_3

matching

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

Goal: Compute matching M a_1 —— b_1, a_2 —— b_3, a_3 —— b_2 optimizing some objective.

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

Goal: Compute matching M

a_1 —— b_1
a_2 —— b_3
a_3 —— b_2

optimizing some objective.

top_k cost $\widehat{=}$ sum of k largest distances.

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

Goal: Compute matching M

a_1 —— b_1
a_2 —— b_3
a_3 —— b_2

optimizing some objective.

top_k cost $\widehat{=}$ sum of k largest distances.

Theorem [F-R,G,L,R,V,2025]**:** There exists a deterministic **mechanism** that is given **only the rankings** and computes a 3**-approximation** on line instances **for all** k. This is **best possible** for line instances.

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

Goal: Compute matching M

a_1 —— b_1
a_2 —— b_3
a_3 —— b_2

optimizing some objective.

top_k cost $\widehat{=}$ sum of k largest distances.

Theorem [F-R,G,L,R,V,2025]**:** There exists a deterministic **mechanism** that is given **only the rankings** and computes a **3-approximation** on line instances **for all** k. This is **best possible** for line instances.

?

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

top_k cost $\widehat{=}$ sum of k largest distances.

Goal: Compute matching M

a_1 —— b_1
a_2 —— b_3
a_3 —— b_2

optimizing some objective.

Theorem [F-R,G,L,R,V,2025]**:** There exists a deterministic **mechanism** that is given **only the rankings** and computes a **3-approximation** on line instances **for all** k. This is **best possible** for line instances.

?

$$\text{distortion}_k(M,\sigma) := \sup_{d_\sigma} \frac{\text{top}_k(M,d_\sigma)}{\text{OPT}_k(d_\sigma)}$$

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

top_k cost $\widehat{=}$ sum of k largest distances.

Goal: Compute matching M

a_1 —— b_1
a_2 —— b_3
a_3 —— b_2

optimizing some objective.

Theorem [F-R,G,L,R,V,2025]**:** There exists a deterministic **mechanism** that is given **only the rankings** and computes a **3-approximation** on line instances **for all** k. This is **best possible** for line instances.

?

$$\text{distortion}_k(M,\sigma) := \sup_{d_\sigma} \frac{\text{top}_k(M,d_\sigma)}{\text{OPT}_k(d_\sigma)}$$

all possible line instances agreeing with σ

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

$$\text{distortion}_k(M,\sigma) := \sup_{d_\sigma} \frac{\text{top}_k(M,d_\sigma)}{\text{OPT}_k(d_\sigma)}$$

Goal: Compute matching M a_1 —— b_1, a_2 —— b_3, a_3 —— b_2 with small distortion.

Theorem [F-R,G,L,R,V,2025]**:** There exists a deterministic **mechanism** that is given **only the rankings** and computes a matching M that has **distortion** 3 on line instances **for all** k. This is **best possible** for line instances.

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

$$\text{distortion}_k(M,\sigma) := \sup_{d_\sigma} \frac{\text{top}_k(M,d_\sigma)}{\text{OPT}_k(d_\sigma)}$$

Goal: Compute matching M a_1 —— b_1, a_2 —— b_3, a_3 —— b_2 with small distortion.

Theorem [F-R,G,L,R,V,2025]**:** There exists a deterministic **mechanism** that is given **only the rankings** and computes a matching M that has **distortion** 3 on line instances **for all** k. This is **best possible** for line instances.

Can we really not do better?



[F-R,G,L,R,V,2025]

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

$$\text{distortion}_k(M,\sigma) := \sup_{d_\sigma} \frac{\text{top}_k(M,d_\sigma)}{\text{OPT}_k(d_\sigma)}$$

Goal: Compute matching M a_1 —— b_1, a_2 —— b_3, a_3 —— b_2 with small distortion.

Theorem [F-R,G,L,R,V,2025]**:** There exists a deterministic **mechanism** that is given **only the rankings** and computes a matching M that has **distortion** 3 on line instances **for all** k. This is **best possible** for line instances.

Can we really not do better?



[F-R,G,L,R,V,2025]



What if we use predictions?

Problem Definition

Instance:

n agents
n items

ordinal ranking

σ :

a_1	a_2	a_3
b_1	b_3	b_3
b_2	b_2	b_2
b_3	b_1	b_1

based on a line

+

prediction

$\widehat{M}$: a_1 —— b_1, a_2 —— b_2, a_3 —— b_3

matching

$$\text{distortion}_k(M,\sigma) := \sup_{d_\sigma} \frac{\text{top}_k(M,d_\sigma)}{\text{OPT}_k(d_\sigma)}$$

Goal: Compute matching M a_1 —— b_1, a_2 —— b_3, a_3 —— b_2 with small distortion.

Theorem: There exists a deterministic **mechanism** that computes a matching M with

$$\textbf{distortion}_k(M,\sigma) = \min\{3, \textbf{distortion}_k(\widehat{M},\sigma)\}$$

on **line instances for all k**.

prediction = 

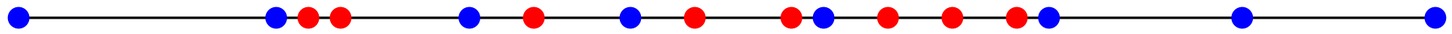


prediction = 



Understanding Optimum Solutions

Lemma [F-R,G,L,R,V]**:** On a known line, left-to-right matching minimizes top$_k$ cost for all k.



Understanding Optimum Solutions

Lemma [F-R,G,L,R,V]**:** On a known line, left-to-right matching minimizes top$_k$ cost for all k.



Understanding Optimum Solutions

Lemma [F-R,G,L,R,V]**:** On a known line, left-to-right matching minimizes top$_k$ cost for all k.



Understanding Optimum Solutions

Lemma [F-R,G,L,R,V]**:** On a known line, left-to-right matching minimizes top$_k$ cost for all k.



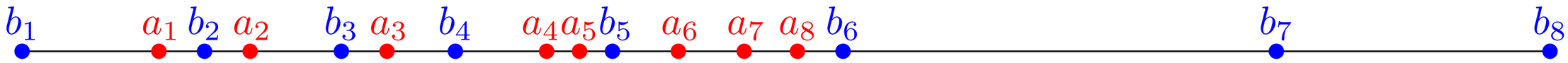
Understanding Optimum Solutions

Lemma [F-R,G,L,R,V]**:** On a known line, left-to-right matching minimizes top_k cost for all k.



Goal: Restore ordering.

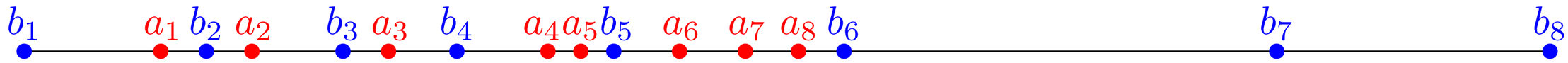
Restoring Ordering



a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1



Restoring Ordering

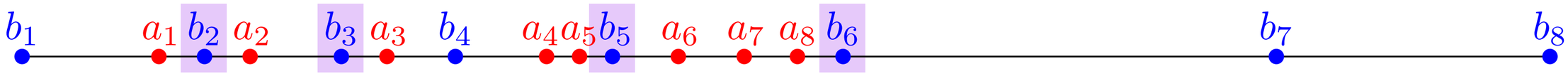


a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

① Find outermost top items b_ℓ and b_r.



Restoring Ordering



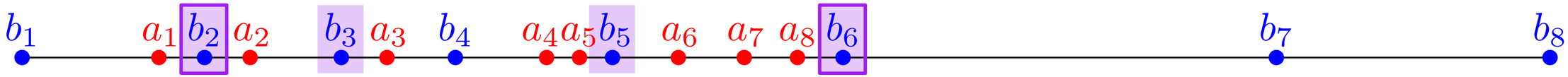
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

① Find outermost top items b_ℓ and b_r.

Consider all top items.



Restoring Ordering



a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

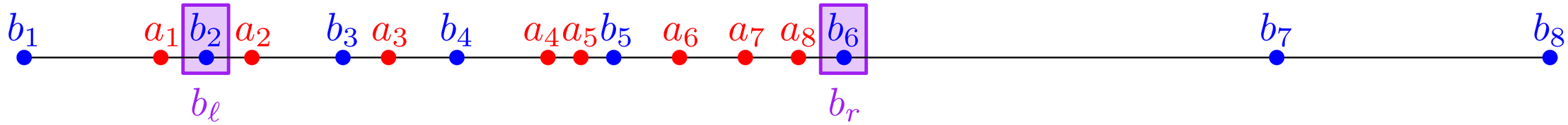
① Find outermost top items b_ℓ and b_r.

Consider all top items.

outermost $\widehat{=}$ bottom row of top items



Restoring Ordering

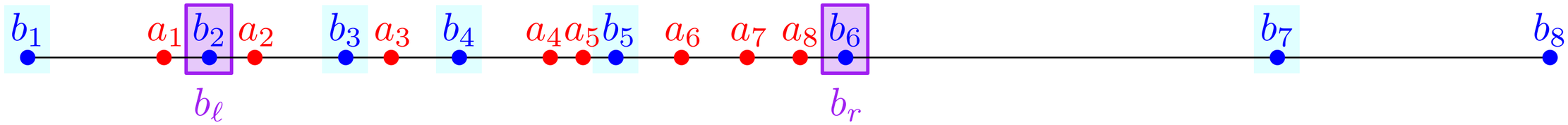


a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

① Find outermost top items b_ℓ and b_r.



Restoring Ordering



a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

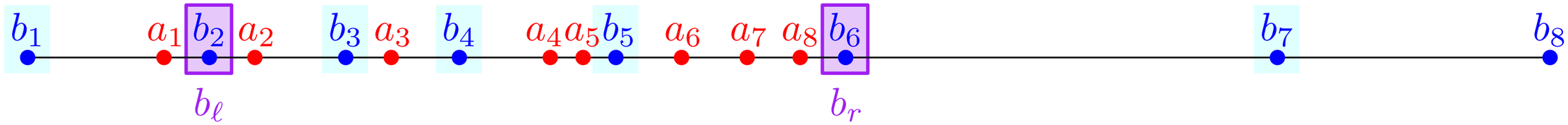
① Find outermost top items b_ℓ and b_r.

② Find sortable items.

$$B_{\text{in}} = \{b \in B : b \succeq_a b_\ell \text{ or } b \succeq_a b_r \text{ for some } a \in A\}$$



Restoring Ordering



a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

① Find outermost top items b_ℓ and b_r.

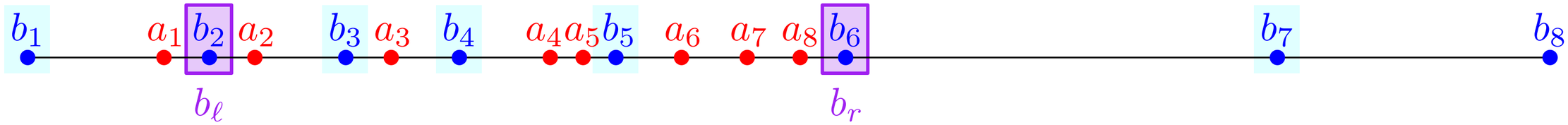
② Find sortable items.

$$B_{\text{in}} = \{b \in B : b \succeq_a b_\ell \text{ or } b \succeq_a b_r \text{ for some } a \in A\}$$

$$b_1 \rightarrow b_2 \rightarrow b_3 \rightarrow b_4 \rightarrow b_5 \rightarrow b_6 \rightarrow b_7$$



Restoring Ordering



a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

① Find outermost top items b_ℓ and b_r.

② Find sortable items.

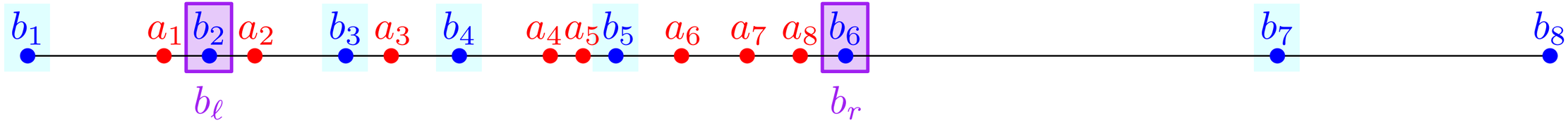
$$B_{\text{in}} = \{b \in B : b \succeq_a b_\ell \text{ or } b \succeq_a b_r \text{ for some } a \in A\}$$

$$b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$$

③ Find weak agent ordering.



Restoring Ordering



a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

① Find outermost top items b_ℓ and b_r.

② Find sortable items.

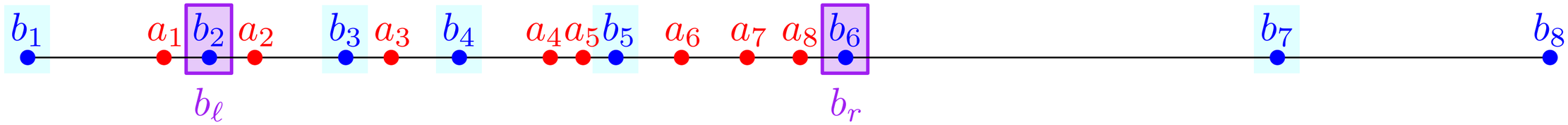
$$B_{\text{in}} = \{b \in B : b \succeq_a b_\ell \text{ or } b \succeq_a b_r \text{ for some } a \in A\}$$

$$b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$$

$$\begin{matrix} a_1 \\ a_2 \end{matrix} \to a_3 \longrightarrow \begin{matrix} a_4 \\ a_5 \\ a_6 \end{matrix} \to \begin{matrix} a_7 \\ a_8 \end{matrix}$$

③ Find weak agent ordering.

Restoring Ordering



a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

① Find outermost top items b_ℓ and b_r.

② Find sortable items.

$$B_{\text{in}} = \{b \in B : b \succeq_a b_\ell \text{ or } b \succeq_a b_r \text{ for some } a \in A\}$$

$$b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$$

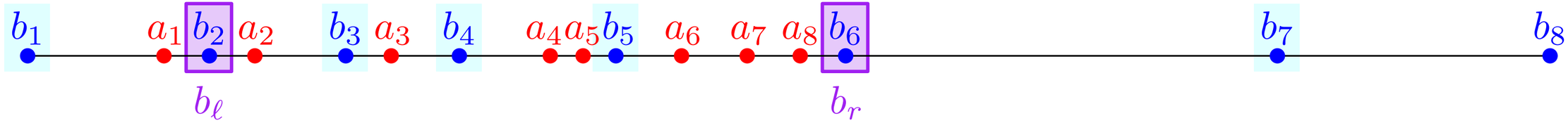
$$\begin{matrix} a_1 \\ a_2 \end{matrix} \to a_3 \longrightarrow \begin{matrix} a_4 \\ a_5 \\ a_6 \end{matrix} \to \begin{matrix} a_7 \\ a_8 \end{matrix}$$

③ Find weak agent ordering.

Not sorted: **Outliers:** $B_{\text{out}} = B \setminus B_{\text{in}}$



Restoring Ordering



a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
b_2	b_2	b_3	b_5	b_5	b_5	b_6	b_6
b_1	b_3	b_4	b_4	b_4	b_6	b_5	b_5
b_3	b_4	b_2	b_3	b_3	b_4	b_4	b_4
b_4	b_1	b_5	b_6	b_6	b_3	b_3	b_3
b_5	b_5	b_1	b_2	b_2	b_2	b_7	b_7
b_6	b_6	b_6	b_1	b_1	b_7	b_2	b_2
b_7	b_7	b_7	b_7	b_7	b_1	b_1	b_8
b_8	b_8	b_8	b_8	b_8	b_8	b_8	b_1

① Find outermost top items b_ℓ and b_r.

② Find sortable items.

$$B_{\text{in}} = \{b \in B : b \succeq_a b_\ell \text{ or } b \succeq_a b_r \text{ for some } a \in A\}$$

$$b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$$

$$\begin{matrix} a_1 \\ a_2 \end{matrix} \to a_3 \longrightarrow \begin{matrix} a_4 \\ a_5 \\ a_6 \end{matrix} \to \begin{matrix} a_7 \\ a_8 \end{matrix}$$

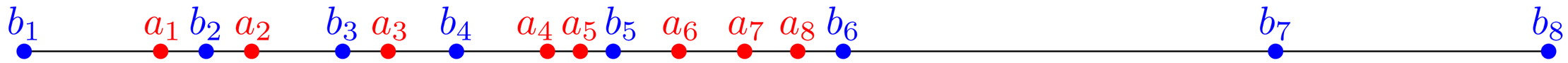
③ Find weak agent ordering.

Not sorted: **Outliers:** $B_{\text{out}} = B \setminus B_{\text{in}}$



Known reconstruction [F-R,G,L,R,V,2025].

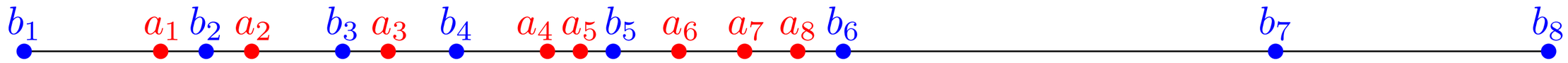
Matching with Reserved Outliers



Algorithm: Given: Preferences σ.



Matching with Reserved Outliers



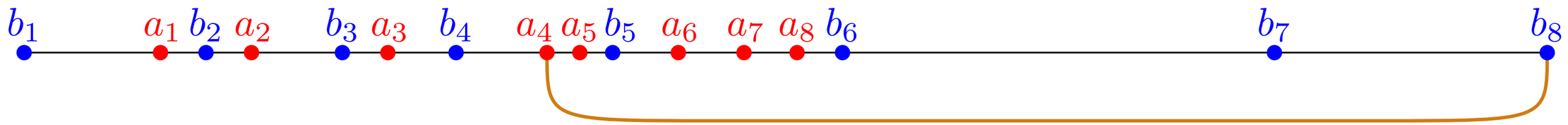
Algorithm: Given: Preferences σ.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \rightarrow b_2 \rightarrow b_3 \rightarrow b_4 \rightarrow b_5 \rightarrow b_6 \rightarrow b_7$



Matching with Reserved Outliers



Algorithm: Given: Preferences σ.

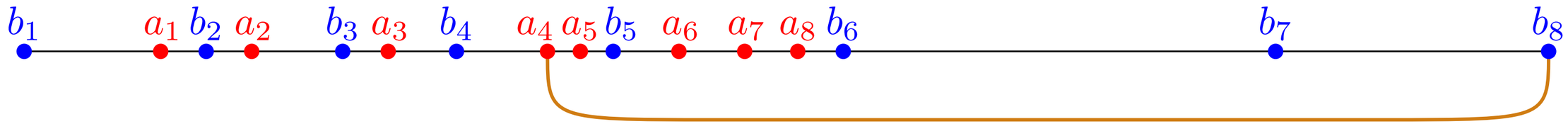
① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \rightarrow b_2 \rightarrow b_3 \rightarrow b_4 \rightarrow b_5 \rightarrow b_6 \rightarrow b_7$

② Assign outliers arbitrarily.



Matching with Reserved Outliers



Algorithm: Given: Preferences σ.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $a_1, a_2 \to a_3 \to a_5, a_6 \to a_7, a_8$

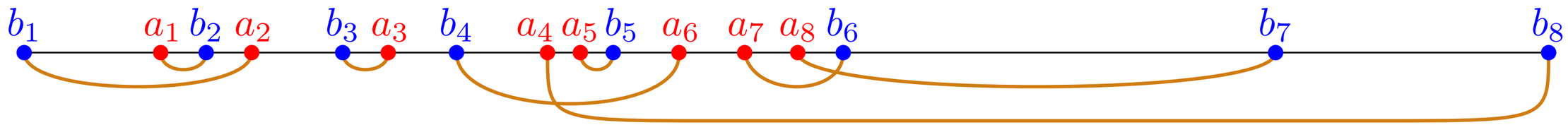
② Assign outliers arbitrarily.

③ Weakly order remaining agents

Agents with same top item: Break ties arbitrarily.



Matching with Reserved Outliers



Algorithm: Given: Preferences σ.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad \begin{matrix} a_1 \\ a_2 \end{matrix} \to a_3 \longrightarrow \begin{matrix} a_5 \\ a_6 \end{matrix} \to \begin{matrix} a_7 \\ a_8 \end{matrix}$

② Assign outliers arbitrarily.

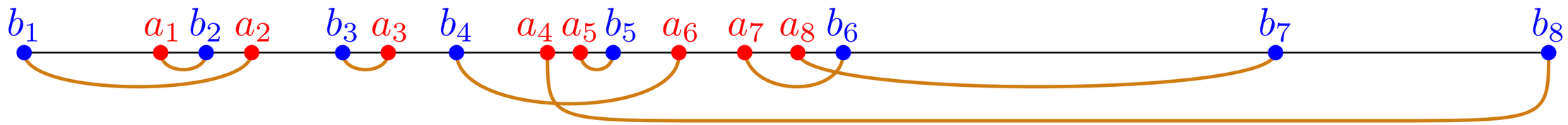
③ Weakly order remaining agents

Agents with same top item: Break ties arbitrarily.

④ Match greedily left to right $\to M$



Matching with Reserved Outliers



Algorithm: Given: Preferences σ.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad a_1, a_2 \to a_3 \longrightarrow a_5, a_6 \to a_7, a_8$

② Assign outliers arbitrarily.

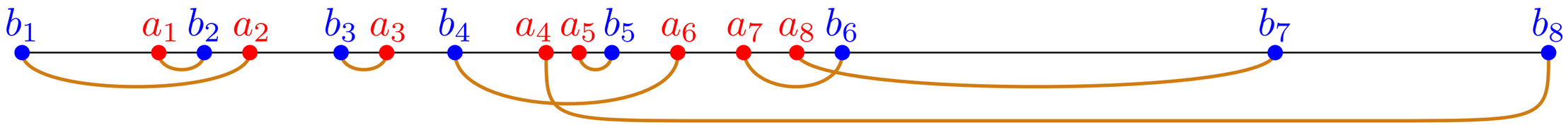
③ Weakly order remaining agents

Agents with same top item: Break ties arbitrarily.

④ Match greedily left to right $\to M$

Lemma: $\text{distortion}_k(M, \sigma) \leq 3$.

Matching with Reserved Outliers



Algorithm: Given: Preferences σ.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad \begin{matrix} a_1 \\ a_2 \end{matrix} \to a_3 \longrightarrow \begin{matrix} a_5 \\ a_6 \end{matrix} \to \begin{matrix} a_7 \\ a_8 \end{matrix}$

Any choice works!

② Assign outliers arbitrarily.

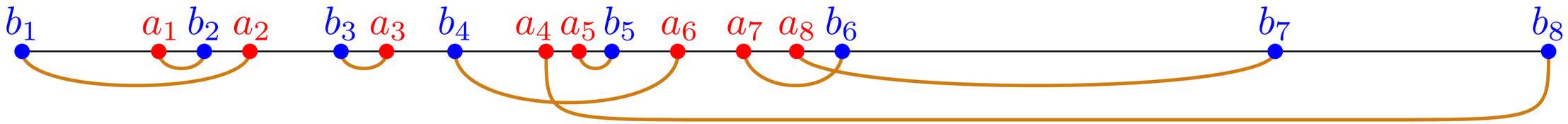
③ Weakly order remaining agents

Agents with same top item: Break ties arbitrarily.

④ Match greedily left to right $\to M$

Lemma: $\text{distortion}_k(M, \sigma) \leq 3$.

Matching with Reserved Outliers



Algorithm: Given: Preferences σ.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad \begin{matrix} a_1 \\ a_2 \end{matrix} \to a_3 \longrightarrow \begin{matrix} a_5 \\ a_6 \end{matrix} \to \begin{matrix} a_7 \\ a_8 \end{matrix}$

Any choice works! new family of matchings

② Assign outliers arbitrarily.

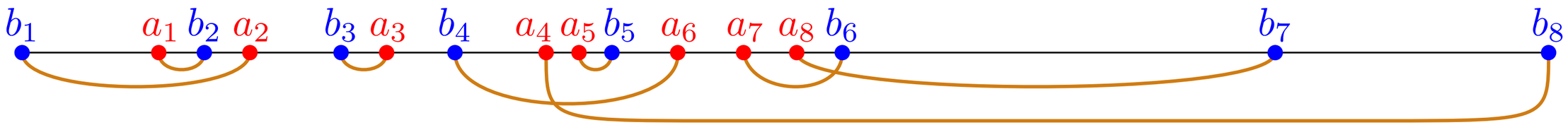
③ Weakly order remaining agents

Agents with same top item: Break ties arbitrarily.

④ Match greedily left to right $\to M$

Lemma: $\text{distortion}_k(M, \sigma) \leq 3$.

Matching with Reserved Outliers



Algorithm: Given: Preferences σ.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad \begin{matrix} a_1 \\ a_2 \end{matrix} \to a_3 \longrightarrow \begin{matrix} a_5 \\ a_6 \end{matrix} \to \begin{matrix} a_7 \\ a_8 \end{matrix}$

Any choice works! new family of matchings

② Assign outliers arbitrarily.

③ Weakly order remaining agents

Agents with same top item: Break ties arbitrarily.

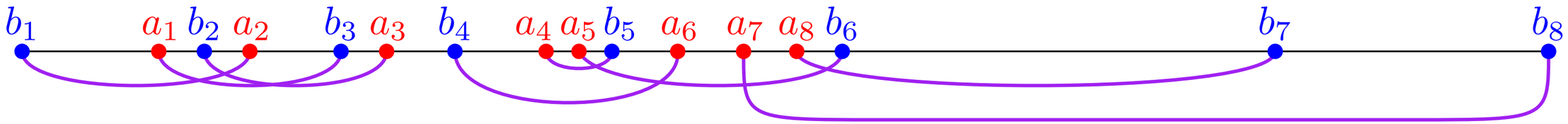
④ Match greedily left to right $\to M$

Lemma: $\text{distortion}_k(M, \sigma) \leq 3$.

Let's use the prediction!



Matching with Reserved Outliers



Algorithm: Given: Preferences σ, prediction $\widehat{M}$.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad \begin{matrix} a_1 \\ a_2 \end{matrix} \to a_3 \longrightarrow \begin{matrix} a_5 \\ a_6 \end{matrix} \to \begin{matrix} a_7 \\ a_8 \end{matrix}$

② Assign outliers arbitrarily.

③ Weakly order remaining agents

Agents with same top item: Break ties arbitrarily.

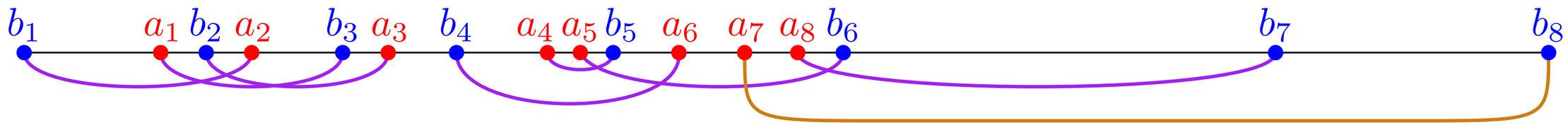
④ Match greedily left to right $\to M$

Lemma: $\text{distortion}_k(M, \sigma) \leq 3$.



Let's use the prediction!

Matching with Reserved Outliers



Algorithm: Given: Preferences σ, prediction $\widehat{M}$.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad \begin{matrix} a_1 \\ a_2 \end{matrix} \to a_3 \longrightarrow \begin{matrix} a_5 \\ a_6 \end{matrix} \to \begin{matrix} a_7 \\ a_8 \end{matrix}$

② Assign outliers according to $\widehat{M}$.

③ Weakly order remaining agents

Agents with same top item: Break ties arbitrarily.

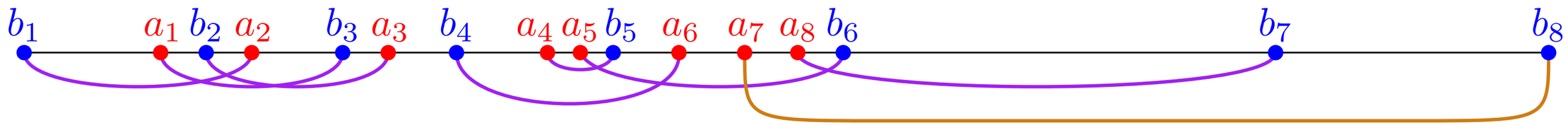
④ Match greedily left to right $\to M$

Lemma: $\text{distortion}_k(M, \sigma) \leq 3$.



Let's use the prediction!

Matching with Reserved Outliers



Algorithm: Given: Preferences σ, prediction $\widehat{M}$.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad a_1, a_2 \to a_3 \to a_4, a_5, a_6 \to a_8$

② Assign outliers according to $\widehat{M}$.

③ Weakly order remaining agents

Agents with same top item: Break ties arbitrarily.

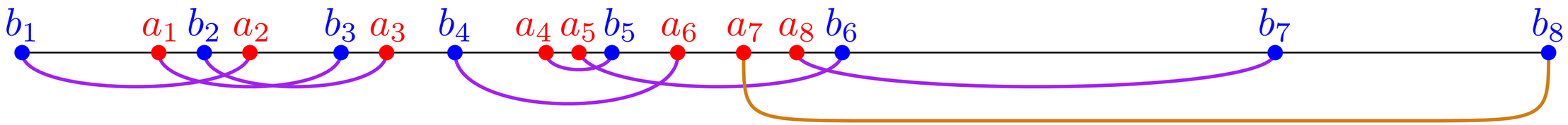
④ Match greedily left to right $\to M$

Lemma: $\text{distortion}_k(M, \sigma) \leq 3$.



Let's use the prediction!

Matching with Reserved Outliers



Algorithm: Given: Preferences σ, prediction $\widehat{M}$.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad a_2 \to a_1 \to a_3 \to a_6 \to a_4 \to a_5 \to a_8$

② Assign outliers according to $\widehat{M}$.

③ Strongly order remaining agents

Break remaining ties using the predicted item order.

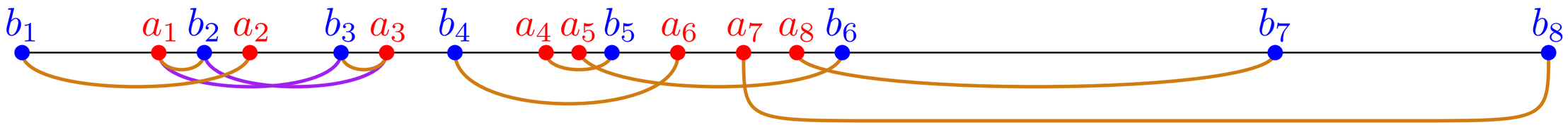
④ Match greedily left to right $\to M$

Lemma: $\text{distortion}_k(M, \sigma) \leq 3$.



Let's use the prediction!

Matching with Reserved Outliers



Algorithm: Given: Preferences σ, prediction $\widehat{M}$.

① Compute outliers and the ordering on B_{in}.

$B_{\text{out}} = \{b_8\}$ $\quad b_1 \to b_2 \to b_3 \to b_4 \to b_5 \to b_6 \to b_7$

Unassigned agents: $\quad a_2 \to a_1 \to a_3 \to a_6 \to a_4 \to a_5 \to a_8$

② Assign outliers according to $\widehat{M}$.

③ Strongly order remaining agents

Break remaining ties using the predicted item order.

④ Match greedily left to right $\to M$

Lemma: $\text{distortion}_k(M, \sigma) \leq \min\{\text{distortion}_k(\widehat{M}, \sigma), 3\}$.



Let's use the prediction!

Take Home Message



Predictions for matching

A proof of concept
for ordinal matching
on the line.

Take Home Message



Predictions for matching

A proof of concept
for ordinal matching
on the line.



New family of matchings

Use prediction for:

- Assignment of Outliers
- order remaining agents

Take Home Message



Predictions for matching

A proof of concept
for ordinal matching
on the line.



New family of matchings

Use prediction for:

- Assignment of Outliers
- order remaining agents



1-consistency

3-robustness

Perfect advice: optimal.

Any advice: distortion ≤ 3.

Take Home Message



Predictions for matching

A proof of concept
for ordinal matching
on the line.



New family of matchings

Use prediction for:

- Assignment of Outliers
- order remaining agents

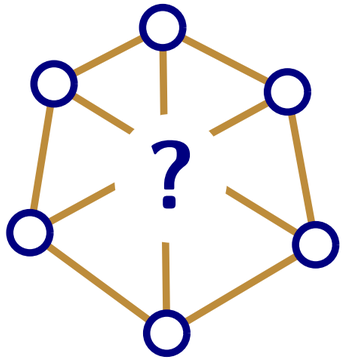


1-consistency

3-robustness

Perfect advice: optimal.

Any advice: distortion ≤ 3.



Open questions

- general metrics
- other forms of prediction
- other cost objectives

Take Home Message



Predictions for matching

A proof of concept
for ordinal matching
on the line.



New family of matchings

Use prediction for:

- Assignment of Outliers
- order remaining agents

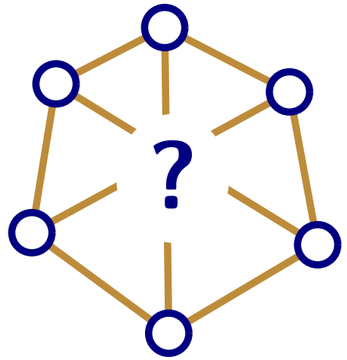


1-consistency

3-robustness

Perfect advice: optimal.

Any advice: distortion ≤ 3.



Open questions

- general metrics
- other forms of prediction
- other cost objectives

Thank you, questions?