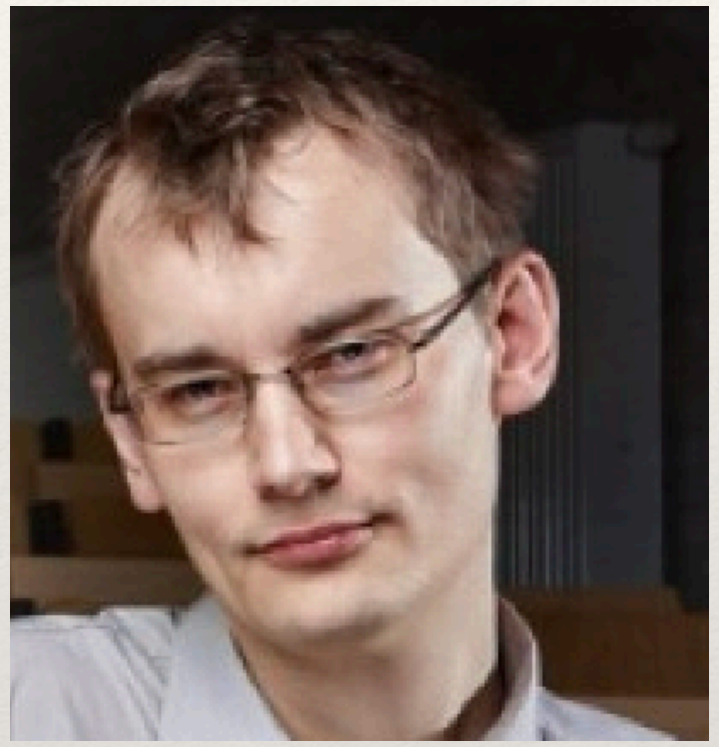
Online Disjoint Set Covers
(Randomization is Not Necessary)

Marcin Bieńkowski
University of Wrocław

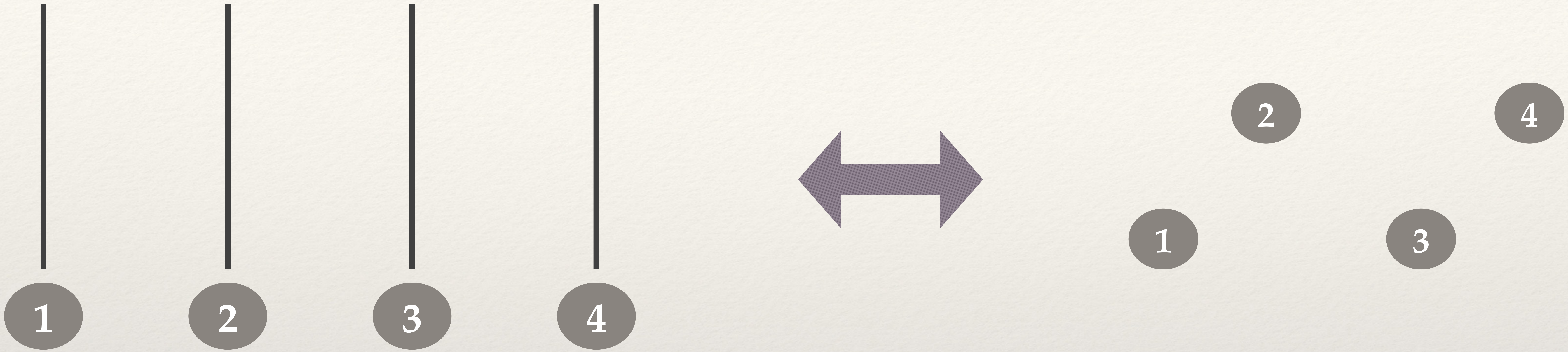


Jarek Byrka
University of Wrocław

Łukasz Jeż
University of Wrocław

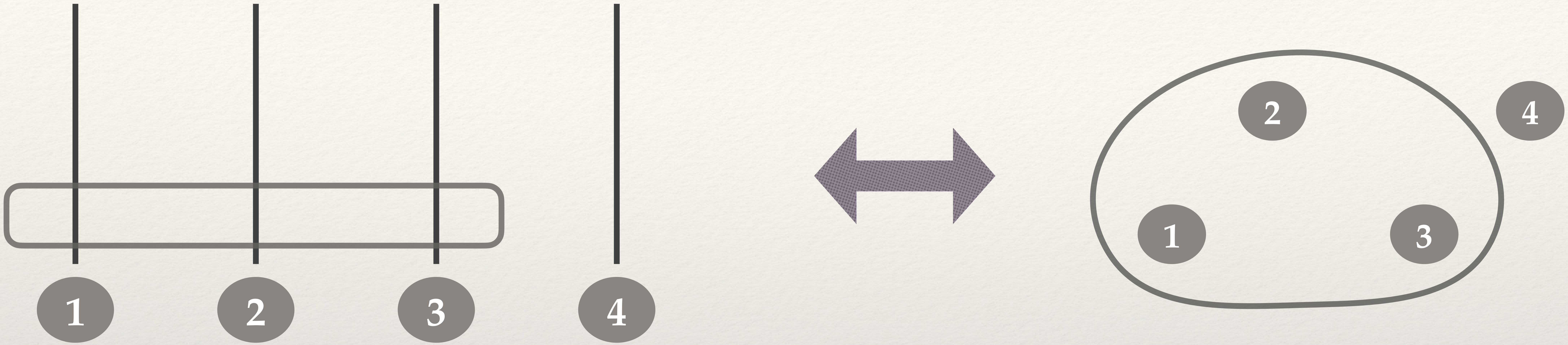
(STACS 2025)

Online Disjoint Set Covers



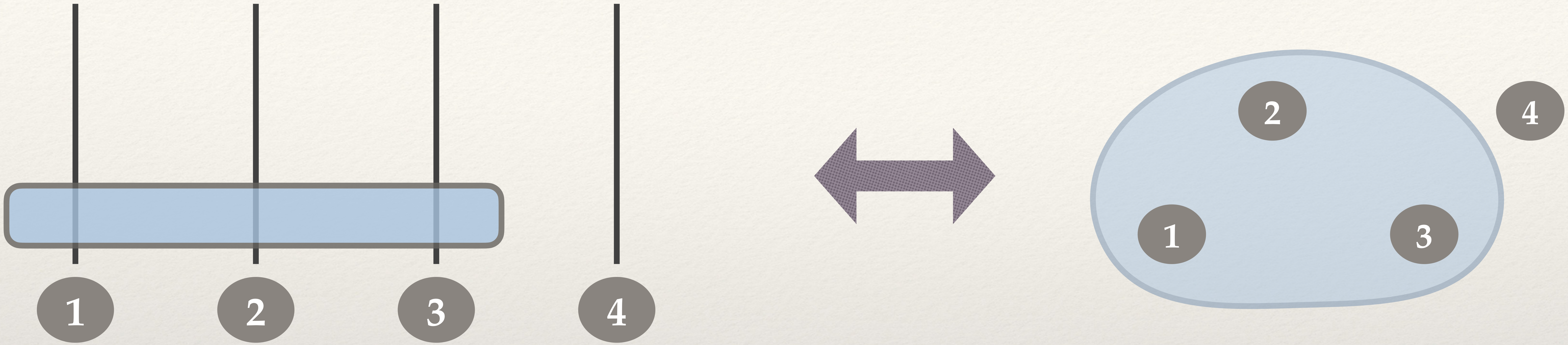
- n elements
- Sequence of sets (**appear online**), each has to be colored
- Set S colored with color $x \implies$ all nodes from S ***collect color x***
- **Gain** = numbers of "**valid**" colors = colors collected by all nodes

Online Disjoint Set Covers



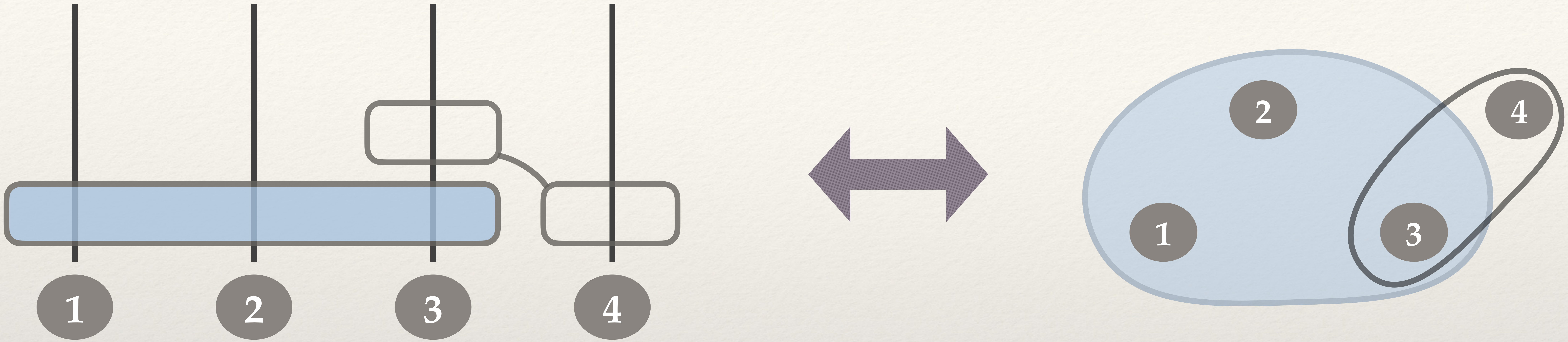
- n elements
- Sequence of sets (**appear online**), each has to be colored
- Set S colored with color $x \implies$ all nodes from S ***collect color x***
- **Gain** = numbers of "**valid**" colors = colors collected by all nodes

Online Disjoint Set Covers



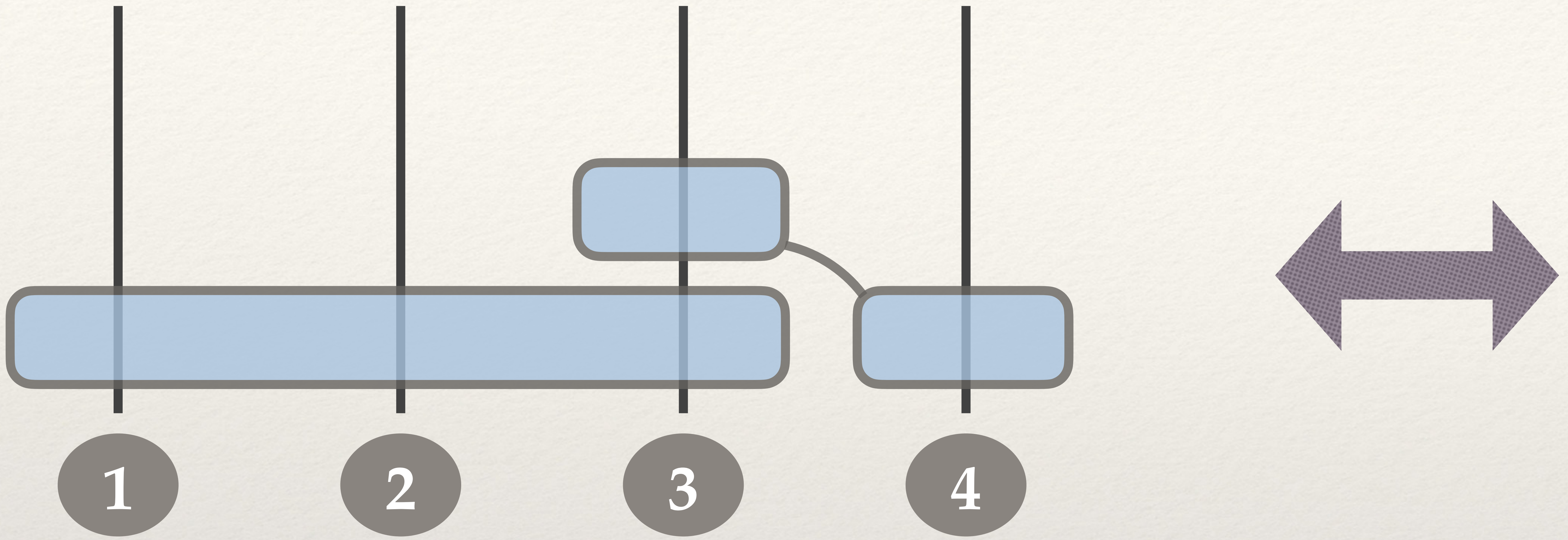
- n elements
- Sequence of sets (**appear online**), each has to be colored
- Set S colored with color $x \Longrightarrow$ all nodes from S ***collect color x***
- **Gain** = numbers of "**valid**" colors = colors collected by all nodes

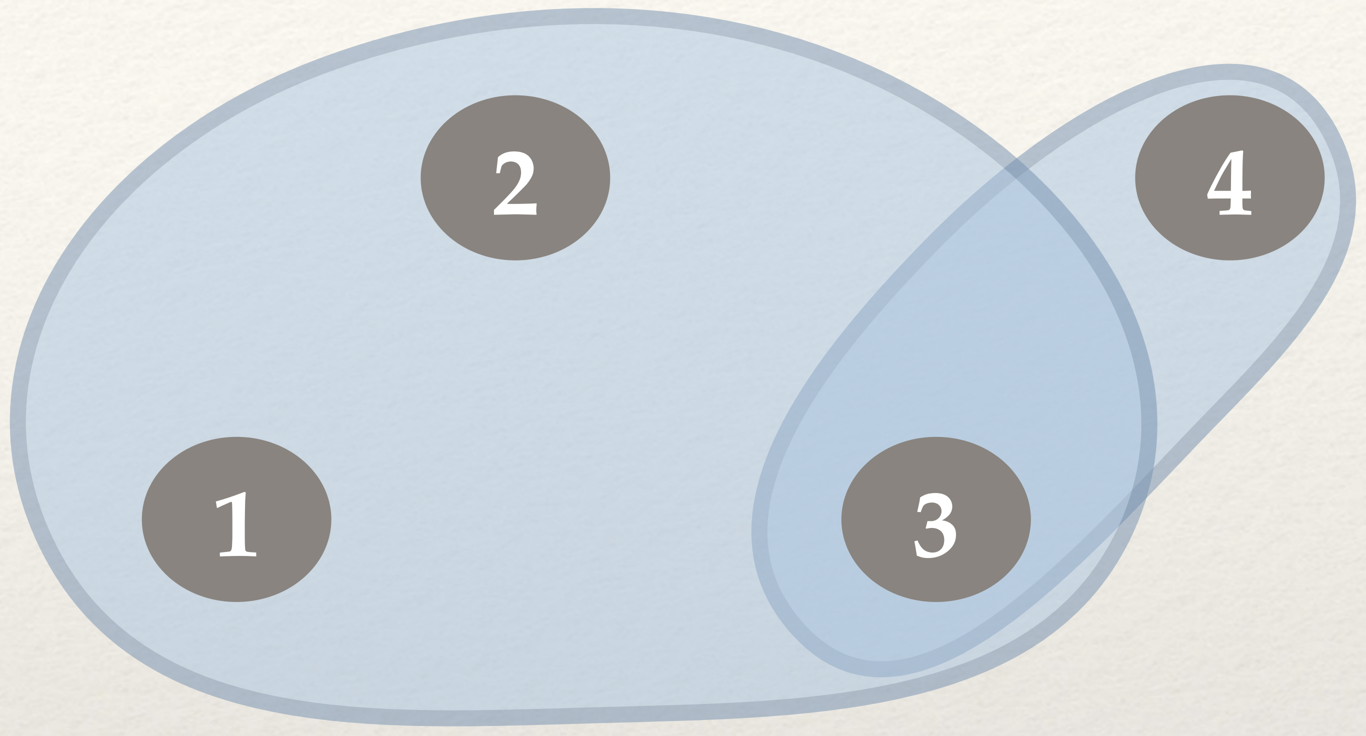
Online Disjoint Set Covers



- n elements
- Sequence of sets (**appear online**), each has to be colored
- Set S colored with color x $\Longrightarrow$ all nodes from S ***collect color x***
- **Gain** = numbers of "**valid**" colors = colors collected by all nodes

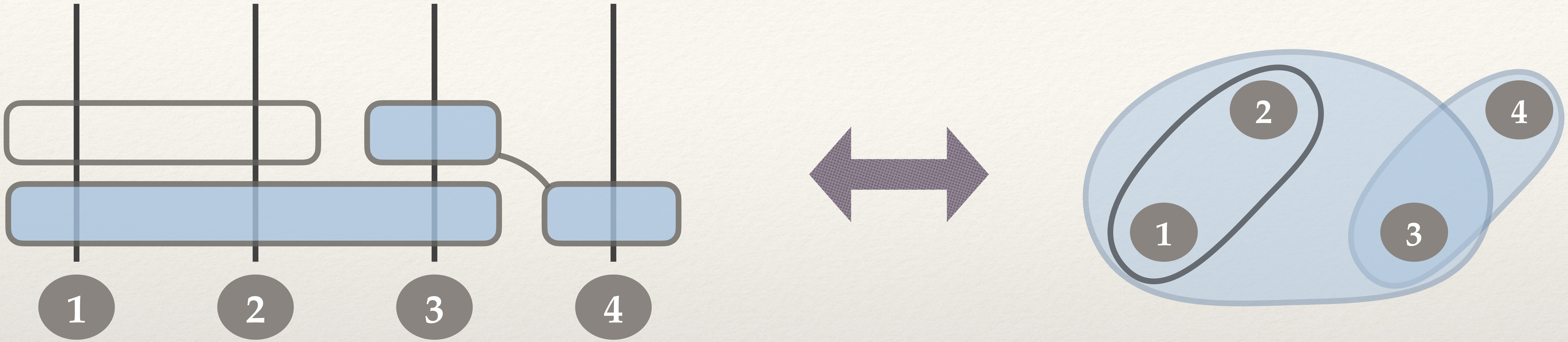
Online Disjoint Set Covers





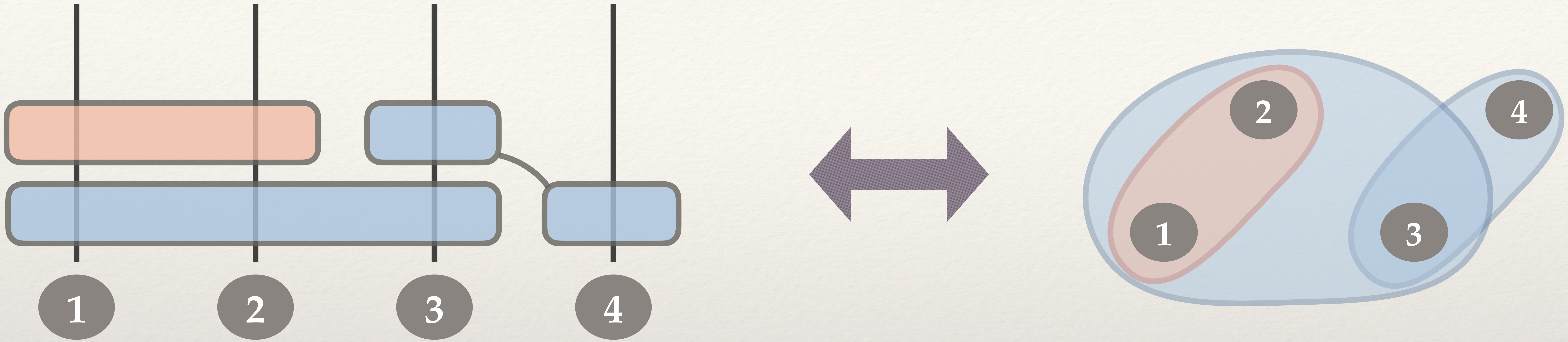
- n elements
- Sequence of sets (**appear online**), each has to be colored
- Set S colored with color $x \implies$ all nodes from S ***collect color x***
- **Gain** = numbers of "**valid**" colors = colors collected by all nodes

Online Disjoint Set Covers



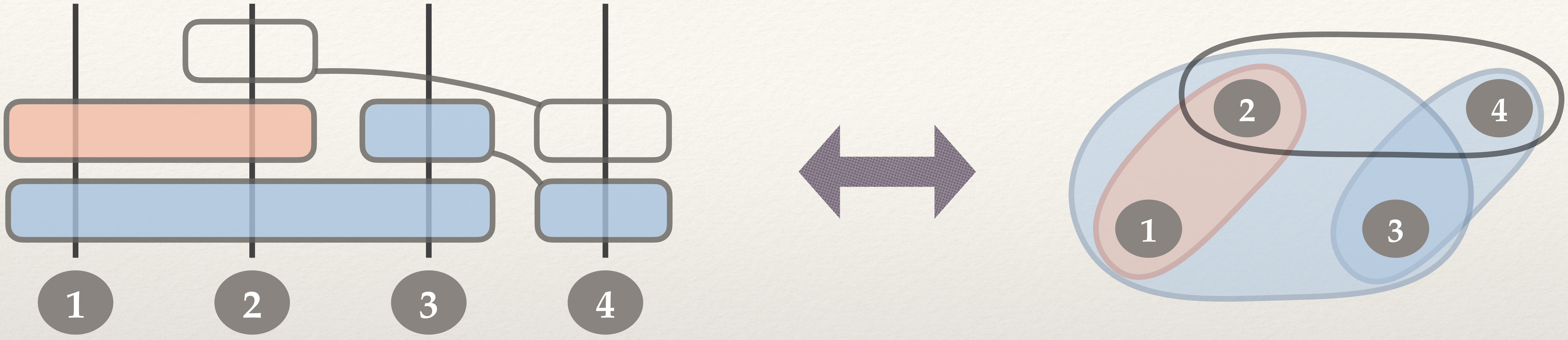
- n elements
- Sequence of sets (**appear online**), each has to be colored
- Set S colored with color $\boldsymbol{x} \Longrightarrow$ all nodes from S ***collect color x***
- **Gain** = numbers of "**valid**" colors = colors collected by all nodes

Online Disjoint Set Covers



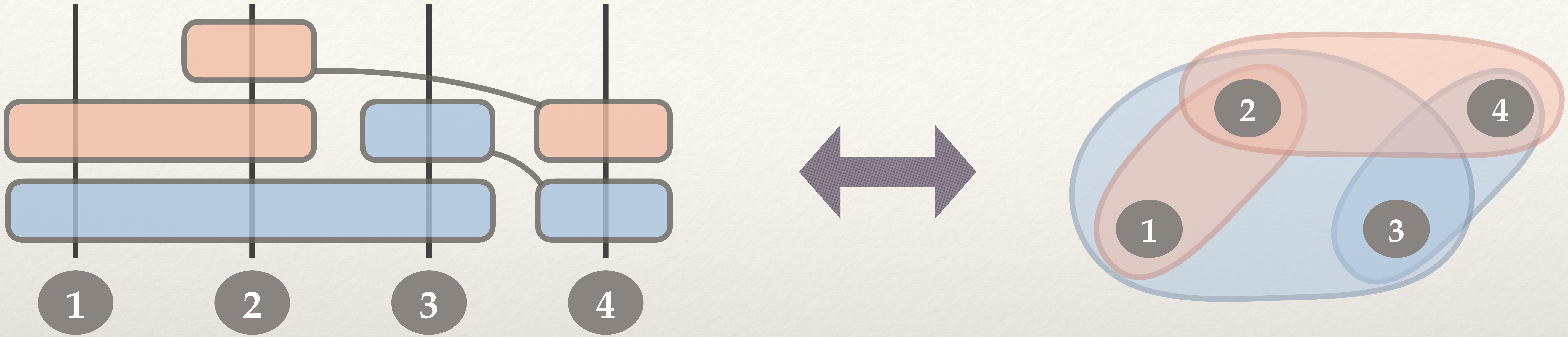
- n elements
- Sequence of sets (**appear online**), each has to be colored
- Set S colored with color $\boldsymbol{x} \Longrightarrow$ all nodes from S ***collect color x***
- **Gain** = numbers of "**valid**" colors = colors collected by all nodes

Online Disjoint Set Covers



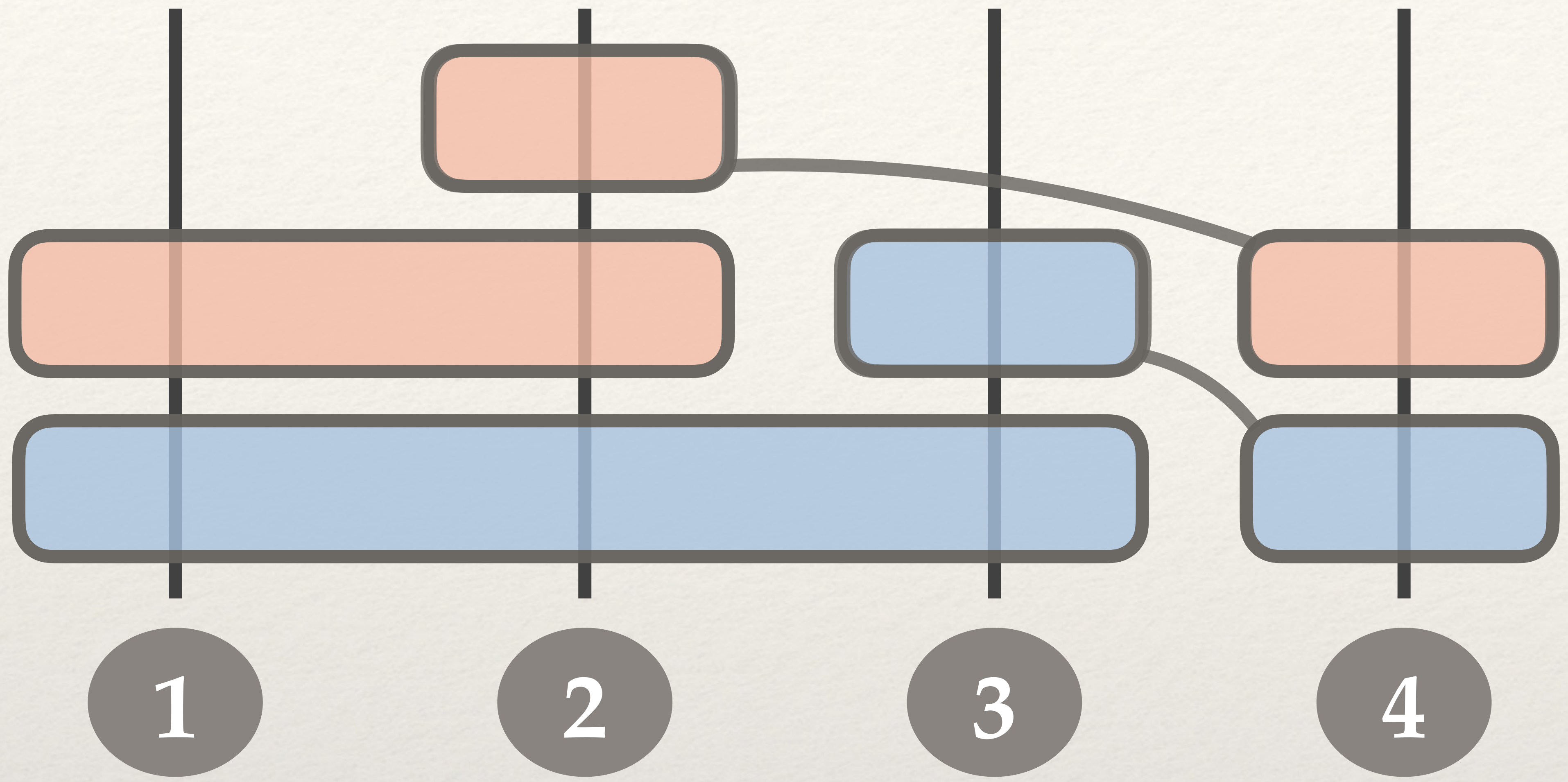
- n elements
- Sequence of sets (**appear online**), each has to be colored
- Set S colored with color $x \implies$ all nodes from S ***collect color x***
- **Gain** = numbers of "**valid**" colors = colors collected by all nodes

Online Disjoint Set Covers

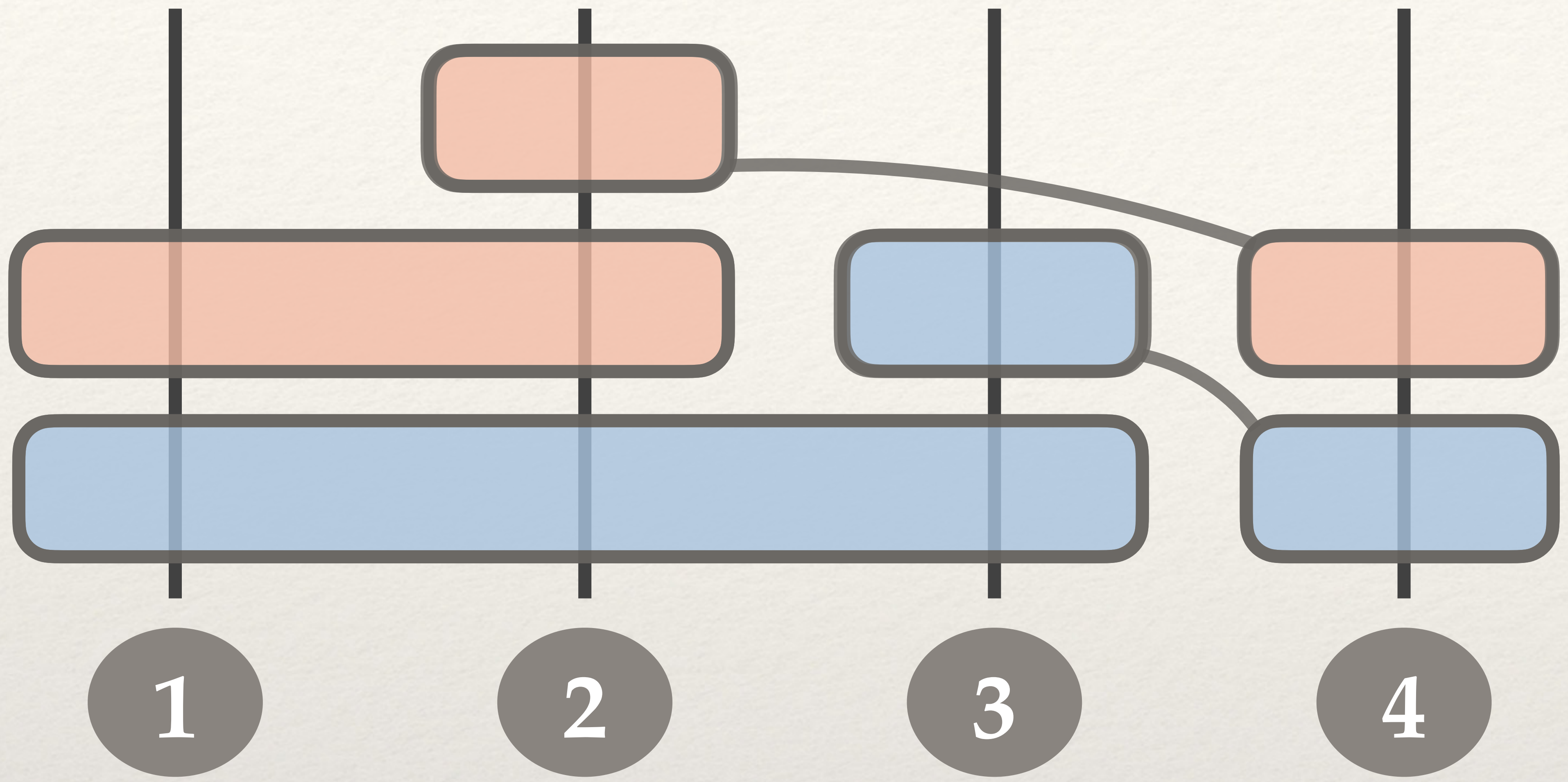


- n elements
- Sequence of sets (**appear online**), each has to be colored
- Set S colored with color $\boldsymbol{x}$ $\Longrightarrow$ all nodes from S ***collect color x***
- **Gain** = numbers of "**valid**" colors = colors collected by all nodes

Competitive ratio

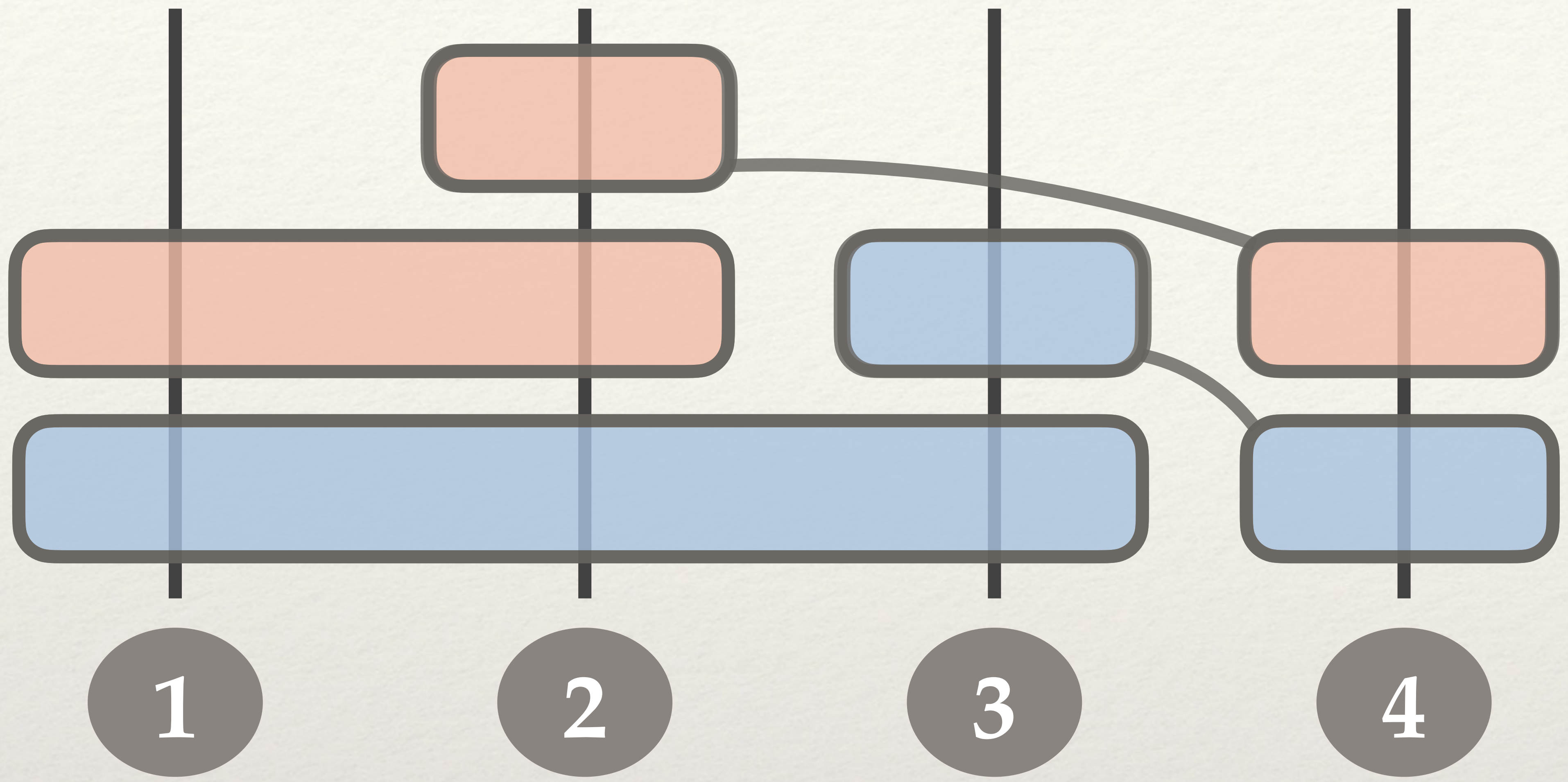


Competitive ratio

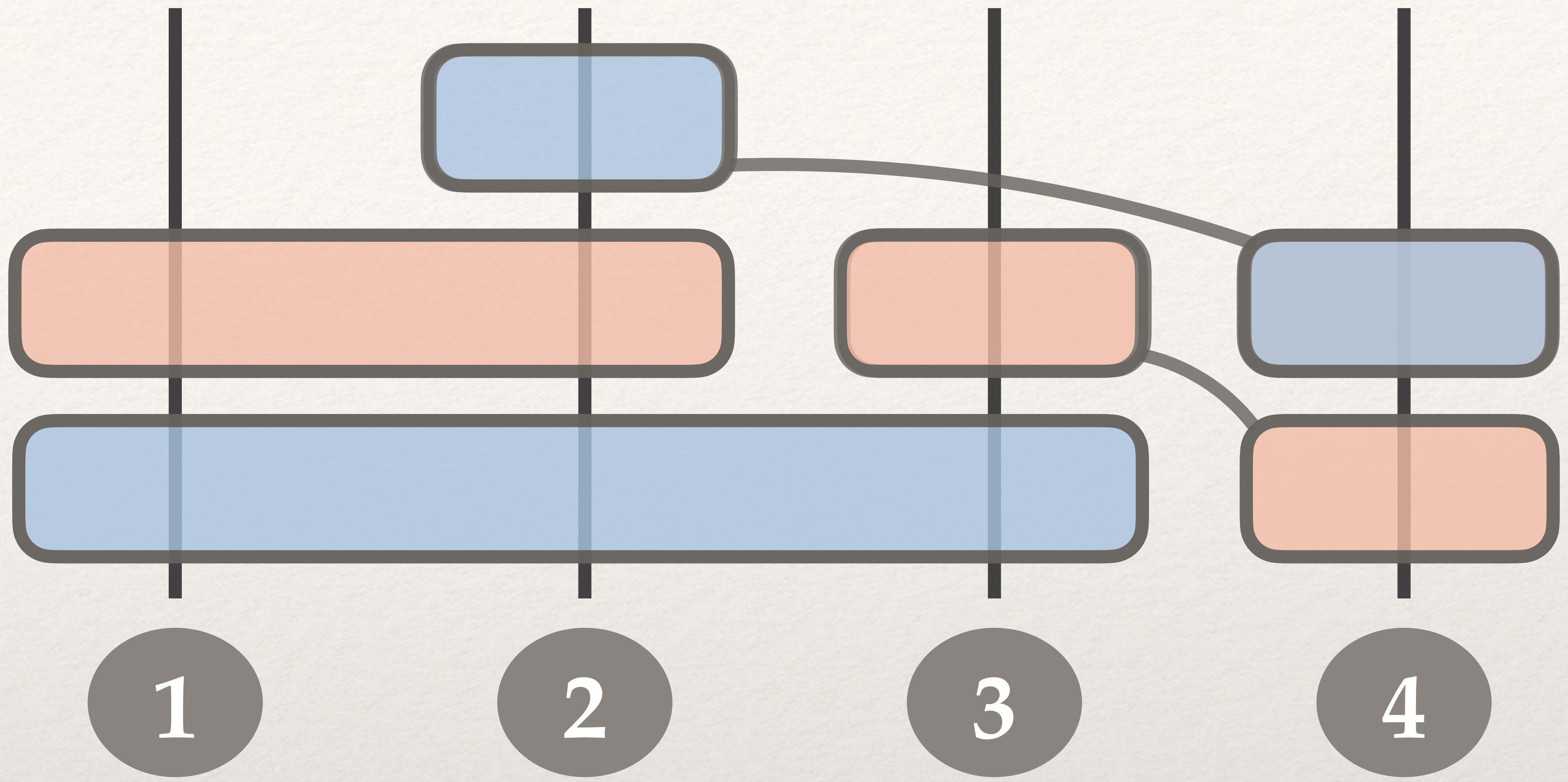


$$\mathrm{ALG}(I) = 1$$

Competitive ratio

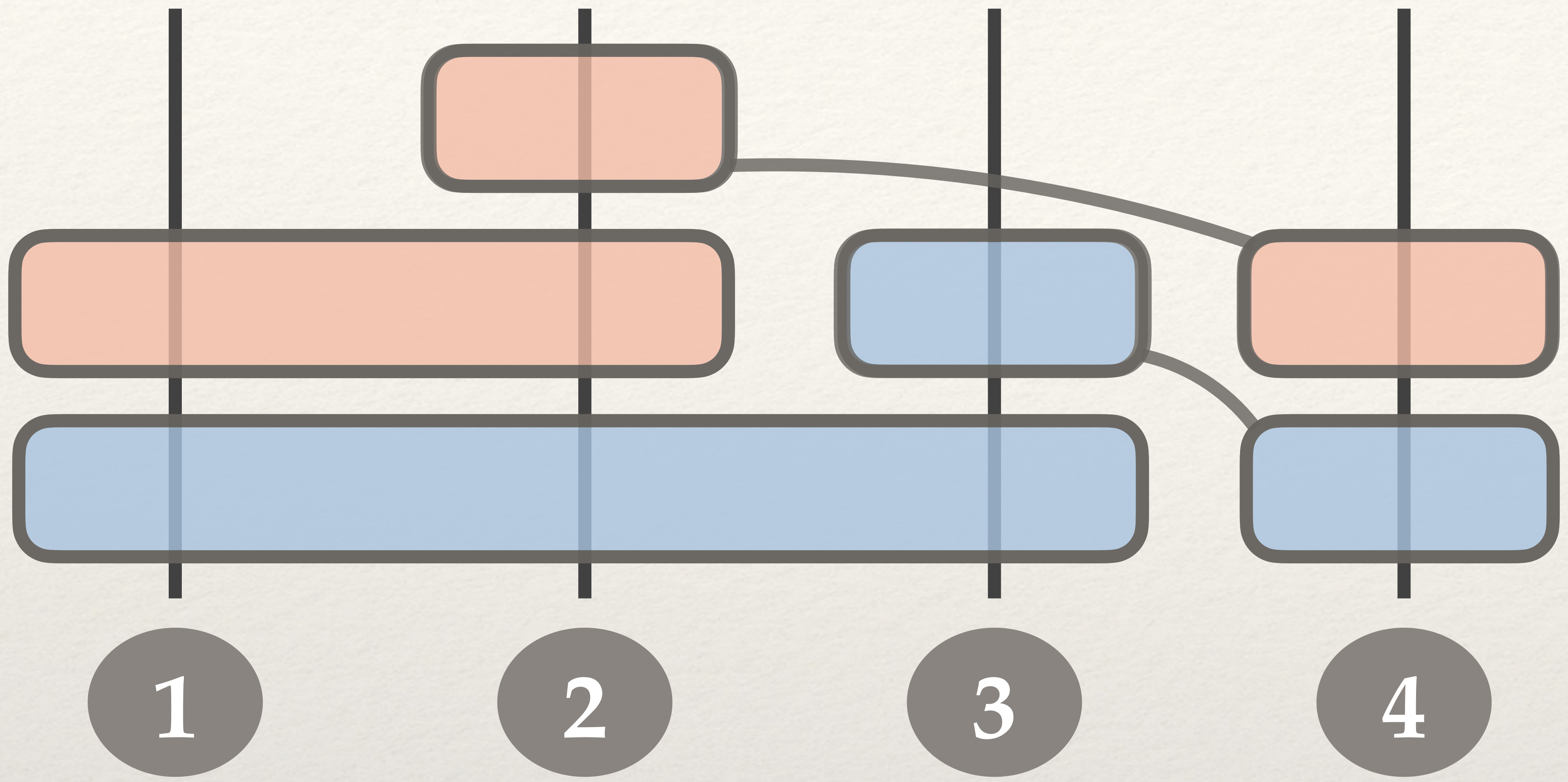


$\mathrm{ALG}(I) = 1$

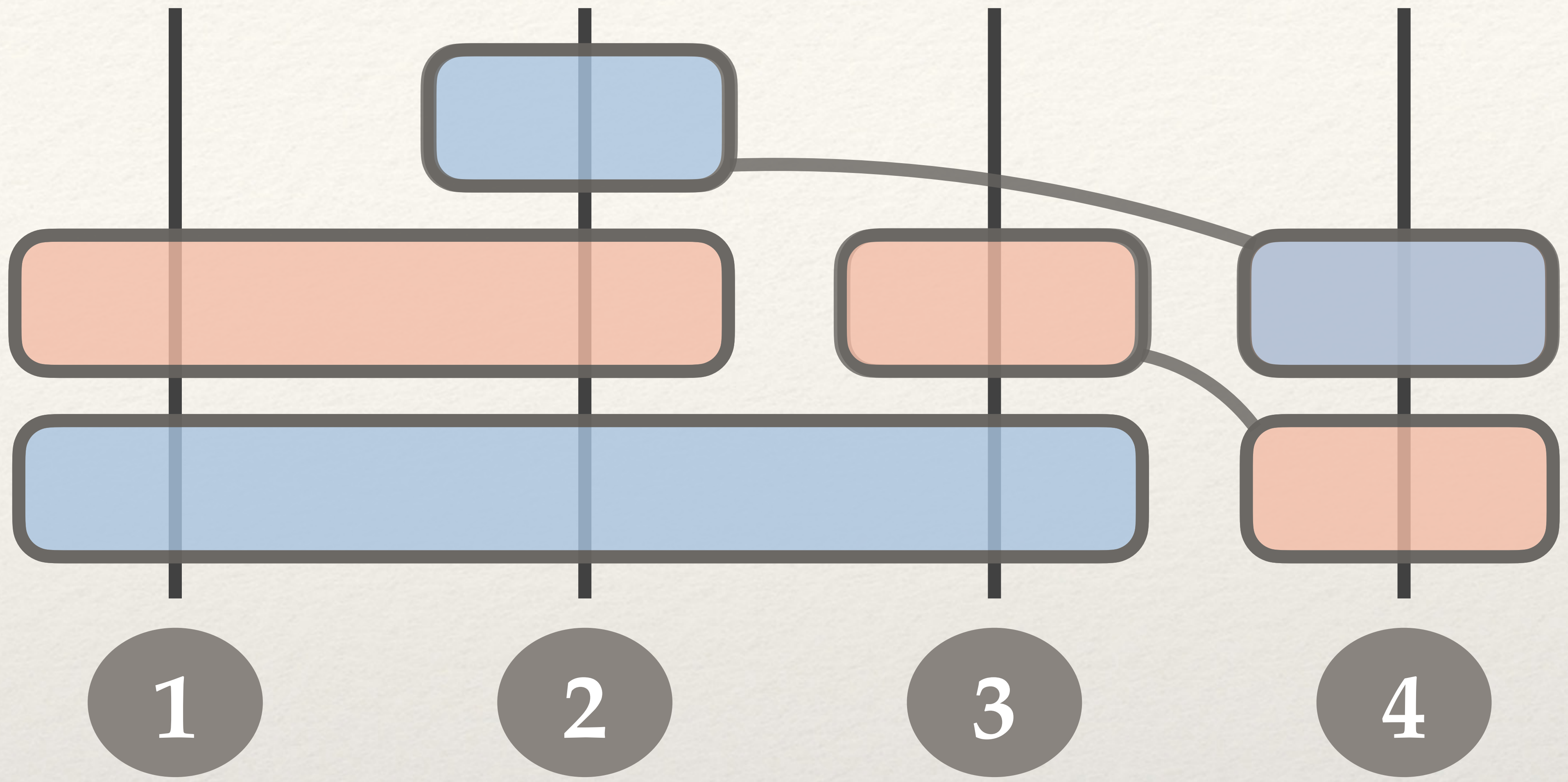


$\mathrm{OPT}(I) = 2$

Competitive ratio



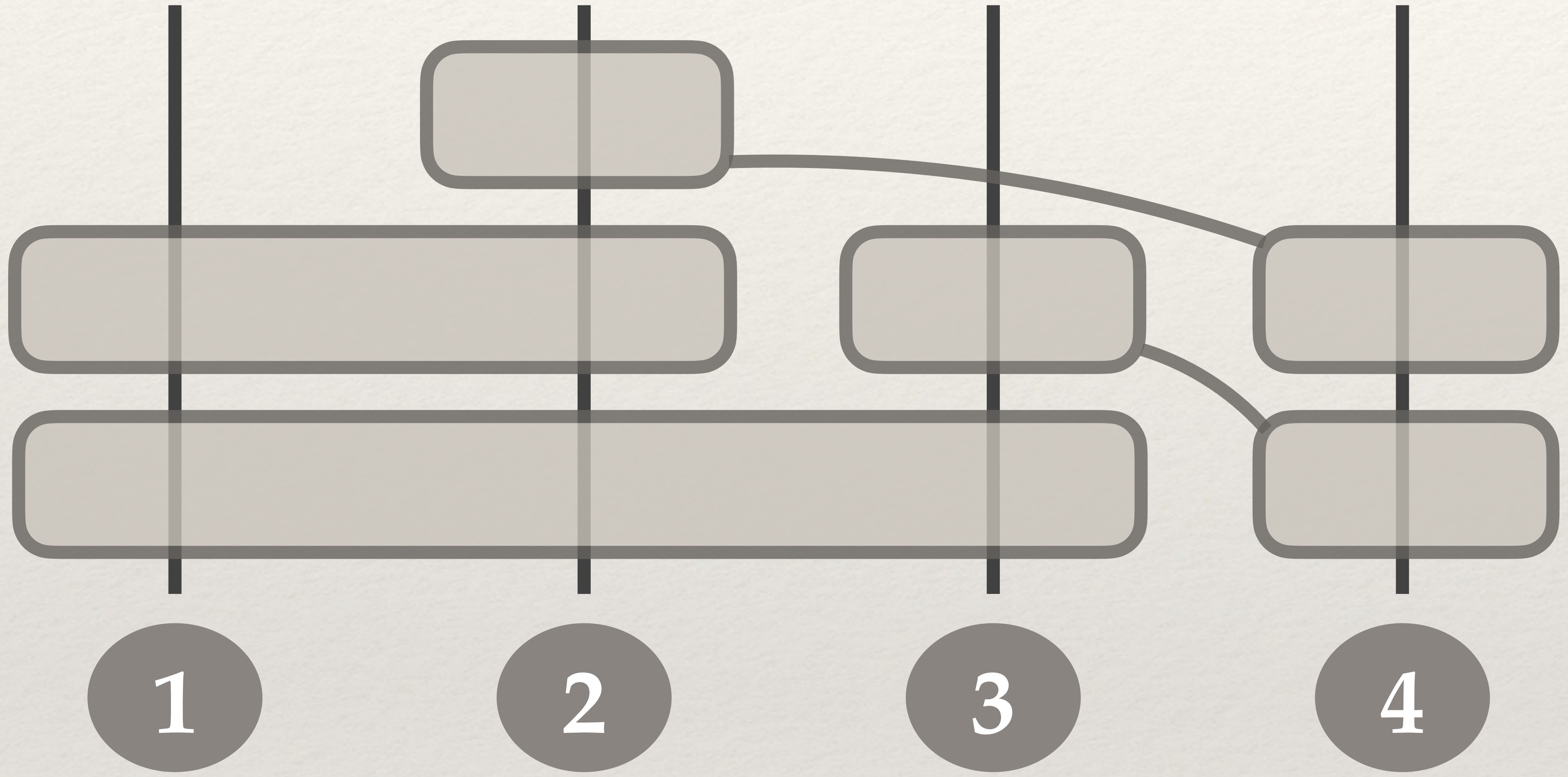
$\mathrm{ALG}(I) = 1$



$\mathrm{OPT}(I) = 2$

Competitive ratio $= \mathrm{OPT}(I) \;/\; \mathrm{ALG}(I) = 2$

OPT ≤ min-degree



- **Gain of OPT** = colors collected by all nodes in OPT's solution
- Gain of OPT ≤ min-degree
- We will compare ALG to min-degree instead of OPT

Results

	Lower bound	Upper bound
randomized algorithms	$O(\log n / \log\log n)$ [1]	$O(\log^2 n)$ [1]
deterministic algorithms	$O(\log n / \log\log n)$ [1]	$O(n)$ [1]

[1] Emek, Goldbraikh, Kantor '19

Results

	Lower bound	Upper bound
randomized algorithms	$O(\log n / \log\log n)$ [1]	$O(\log^2 n)$ [1]
deterministic algorithms	$O(\log n / \log\log n)$ [1]	$O(n)$ [1]
		$O(\log^2 n)$

Our result

[1] Emek, Goldbraikh, Kantor '19

Results

	Lower bound	Upper bound
randomized algorithms	$O(\log n / \log\log n)$ [1]	$O(\log^2 n)$ [1]
deterministic algorithms	$O(\log n / \log\log n)$ [1]	$O(n)$ [1]
		$O(\log^2 n)$

Our result

[1] Emek, Goldbraikh, Kantor '19

Offline case: O(log n)-apx exists, not possible to improve it unless NP $\subseteq$ DTIME($n^{\log\log n}$)

The Plan

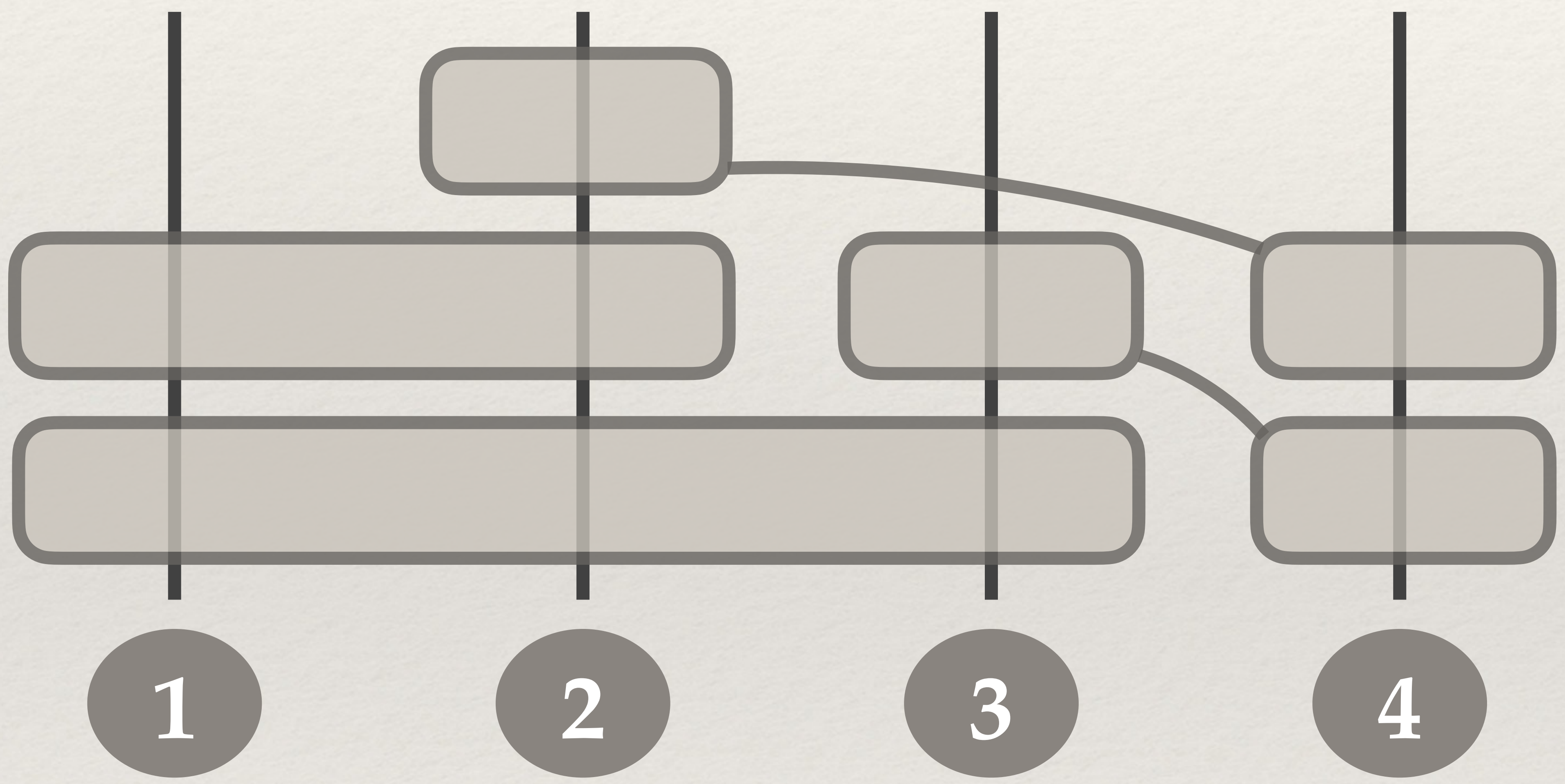
- Problem definition ✅
- Previous results ✅
- **Min-degree known a priori**
  - **➜ randomized O(log n)-competitive solution**
  - ➜ deterministic O(log n)-competitive solution
- Challenges for unknown min-degree

The Plan

- Problem definition ✅
- Previous results ✅
- **Min-degree known a priori**
  - ➜ **randomized O(log n)-competitive solution**
  - ➜ deterministic O(log n)-competitive solution
- Challenges for unknown min-degree

Our framework, but for an already known result

Known min-degree δ



Randomized algorithm [Pananjady, Bagaria, Vaze '15]

- Fix color palette $P = \{1, 2, \ldots, \Theta(\delta / \log n)\}$
- For each set: choose color u.a.r. from P

Analysis

- Each node gets all colors from P w.h.p.
- This holds for all nodes w.h.p.
- $\text{ALG} = |P| = \Theta(\delta / \log n) \geq \text{OPT} \ / \ \log n$

The Plan

- Problem definition ✅
- Previous results ✅
- **Min-degree known a priori**
  - ➜ randomized O(log n)-competitive solution ✅
  - **➜ deterministic O(log n)-competitive solution**

Our framework, but for an already known result

- Challenges for unknown min-degree

Known min-degree δ

Deterministic algorithm

- Fix color palette $P = \{1, 2, \ldots, \Theta(\delta / \log n)\}$
- For each set: choose color ~~u.a.r.~~ from P **(in a smart way)**

Node performance

- **How well is node *i* performing?**
- $\deg(i)$ = number of steps when node *i* collected colors
- $c(i)$ = #colors node *i* collected so far
- $$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

Node performance

- **How well is node *i* performing?**
- $\deg(i)$ = number of steps when node i collected colors
- $c(i)$ = #colors node i collected so far
- $$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

expected number of steps to gain c(i) colors

if colors are chosen u.a.r from P

Node performance

- **How well is node i performing?**
- $\deg(i)$ = number of steps when node i collected colors
- $c(i)$ = #colors node i collected so far
- $$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

expected number of steps to gain c(i) colors

if colors are chosen u.a.r from P

Think:

- **high $Z(i)$ = bad performance of node i**
- **goal: keep all $Z(i)$ small**

Node performance

- **How well is node *i* performing?**
- $\deg(i)$ = number of steps when node i collected colors
- $c(i)$ = #colors node i collected so far

expected number of steps to gain c(i) colors

if colors are chosen u.a.r from P

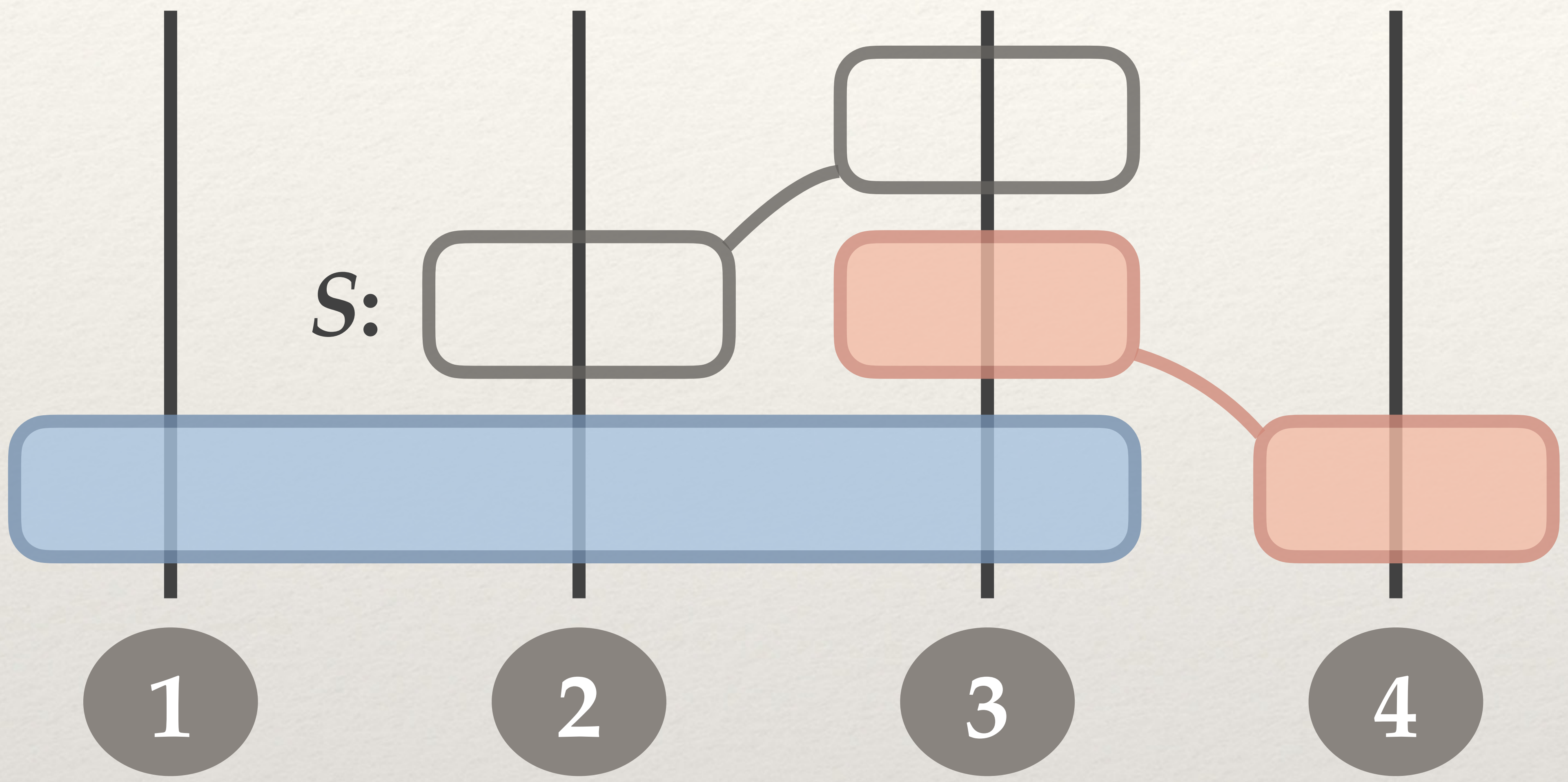
- $$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

- Initially, $Z(i) = 0$

Think:

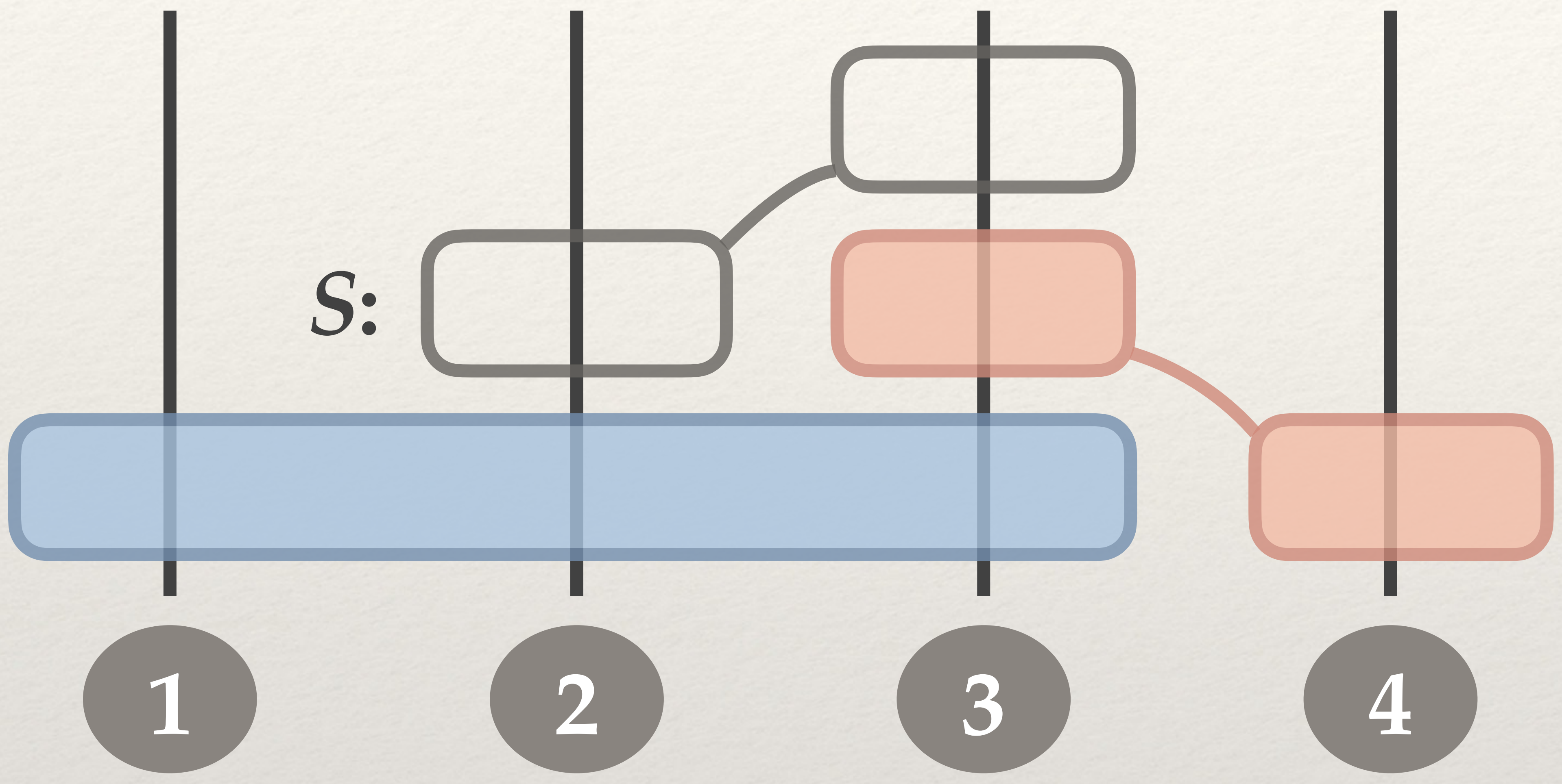
- **high $Z(i)$ = bad performance of node i**
- **goal: keep all $Z(i)$ small**

How Z(i) changes if colors are chosen randomly?



$$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

How Z(i) changes if colors are chosen randomly?



$$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

Case 1. i $\notin S$

- $\Delta \deg(i) = 0, \Delta c(i) = 0 \Longrightarrow \Delta Z(i) = 0$

Case 2. i $\in S$

- $\Delta \deg(i) = 1$
- $\Delta c(i) = \begin{cases} 1 & \text{with probability} \frac{|P| - c(i)}{|P|} \\ 0 & \text{otherwise} \end{cases}$
- $E[\Delta Z(i)] = 1 - 2 \cdot 1 = -1 < 0$

Choosing a color for set S: Applying probabilistic method (1)

Recall the goal: keep all Z(i) small

- Initially, $Z(i) = 0$

Choosing a color for set S: Applying probabilistic method (1)

Recall the goal: keep all Z(i) small

- Initially, $Z(i) = 0$
- If we choose color for S randomly, then:

Choosing a color for set S: Applying probabilistic method (1)

Recall the goal: keep all Z(i) small

- Initially, $Z(i) = 0$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***

Choosing a color for set *S*: Applying probabilistic method (1)

Recall the goal: keep all Z(*i*) small

- Initially, $Z(i) = 0$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - $\sum_{i=1}^{n} Z(i)$ decreases ***in expectation***

Choosing a color for set S: Applying probabilistic method (1)

Recall the goal: keep all Z(i) small

- Initially, $Z(i) = 0$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - $\sum_{i=1}^{n} Z(i)$ decreases ***in expectation***
- There ***exists a deterministic*** choice of color for S, s.t. $\sum_{i=1}^{n} Z(i)$ decreases.

Choosing a color for set S: Applying probabilistic method (1)

Recall the goal: keep all Z(i) small

- Initially, $Z(i) = 0$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - $\sum_{i=1}^{n} Z(i)$ decreases ***in expectation***
- There ***exists a deterministic*** choice of color for S, s.t. $\sum_{i=1}^{n} Z(i)$ decreases.
- But it does not imply small values of Z(i)'s for all nodes!
- **Dead end?**

Choosing a color for set S: Applying probabilistic method (1)

Recall the goal: keep all Z(i) small

- Initially, $Z(i) = 0$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - $\sum_{i=1}^{n} Z(i)$ decreases ***in expectation***
- There ***exists a deterministic*** choice of color for S, s.t. $\sum_{i=1}^{n} Z(i)$ decreases.

- But it does not imply small values of Z(i)'s for all nodes!
- **Dead end?**

Trick: replace $Z(i)$ by $\exp(Z(i) \,/\, |P|)$

Alon, Awerbuch, Azar, Buchbinder, Naor '03

Choosing a color for set *S*: Applying probabilistic method (2)

Recall the goal: keep all Z(*i*) small

- Initially, $Z(i) = 0$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - $\sum_{i=1}^{n} Z(i)$ decreases ***in expectation***
- There ***exists a det.*** choice of color for S, s.t. $\sum_{i=1}^{n} Z(i)$ decreases.

Choosing a color for set S: Applying probabilistic method (2)

Recall the goal: keep all Z(*i*) small

- Initially, ~~$Z(i) = 0$~~ $\exp\left(Z(i) \,/\, |P|\right) = 1$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - $\sum_{i=1}^{n} Z(i)$ decreases ***in expectation***
- There ***exists a det.*** choice of color for S, s.t. $\sum_{i=1}^{n} Z(i)$ decreases.

Choosing a color for set S: Applying probabilistic method (2)

Recall the goal: keep all Z(i) small

- Initially, ~~$Z(i) = 0$~~ $\exp\left(Z(i) / |P|\right) = 1$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - each $\exp\left(Z(i) / |P|\right)$ decreases ***in expectation***
  - $\sum_{i=1}^{n} Z(i)$ decreases ***in expectation***
- There ***exists a det.*** choice of color for S, s.t. $\sum_{i=1}^{n} Z(i)$ decreases.

Reverse Jensen's type inequality

Choosing a color for set S: Applying probabilistic method (2)

Recall the goal: keep all Z(i) small

- Initially, ~~$Z(i) = 0$~~ $\exp\left(Z(i) / |P|\right) = 1$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - each $\exp\left(Z(i) / |P|\right)$ decreases ***in expectation***
  - ~~$\sum_{i=1}^{n} Z(i)$~~ $\sum_{i=1}^{n} \exp\left(Z(i) / |P|\right)$ decreases ***in expectation***
- There ***exists a det.*** choice of color for S, s.t. $\sum_{i=1}^{n} Z(i)$ decreases.

Reverse Jensen's type inequality

Choosing a color for set S: Applying probabilistic method (2)

Recall the goal: keep all Z(i) small

- Initially, ~~$Z(i) = 0$~~ $\exp\left(Z(i) \,/\, |P|\right) = 1$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - each $\exp\left(Z(i) \,/\, |P|\right)$ decreases ***in expectation***
  - ~~$\sum_{i=1}^{n} Z(i)$~~ $\sum_{i=1}^{n} \exp\left(Z(i) \,/\, |P|\right)$ decreases ***in expectation***
- There ***exists a det.*** choice of color for S, s.t. ~~$\sum_{i=1}^{n} Z(i)$~~ $\sum_{i=1}^{n} \exp\left(Z(i) \,/\, |P|\right)$ decreases.

Reverse Jensen's type inequality

This is our algorithm!

Choosing a color for set S: Applying probabilistic method (2)

Recall the goal: keep all Z(i) small

- Initially, ~~$Z(i) = 0$~~ $\exp\left(Z(i)\,/\,|P|\right) = 1$
- If we choose color for S randomly, then:
  - each $Z(i)$ decreases ***in expectation***
  - each $\exp\left(Z(i)\,/\,|P|\right)$ decreases ***in expectation***
  - ~~$\sum_{i=1}^{n} Z(i)$~~ $\sum_{i=1}^{n} \exp\left(Z(i)\,/\,|P|\right)$ decreases ***in expectation***

Reverse Jensen's type inequality

- There ***exists a det.*** choice of color for S, s.t. ~~$\sum_{i=1}^{n} Z(i)$~~ $\sum_{i=1}^{n} \exp\left(Z(i)\,/\,|P|\right)$ decreases.

- $\Longrightarrow \sum_{i=1}^{n} \exp\left(Z(i)\,/\,|P|\right) \le n \;\Longrightarrow\; Z(i) \le |P| \cdot \ln n$ for each i

Guarantees of the algorithm

$$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

- For any node i, it always holds $Z(i) \leq |P| \cdot \ln n$

Guarantees of the algorithm

$$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

- For any node i, it always holds $Z(i) \leq |P| \cdot \ln n$

- **Suppose for a contradiction** that at the end of the execution $c(i) < |P|$

Guarantees of the algorithm

$$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

- For any node i, it always holds $Z(i) \leq |P| \cdot \ln n$

- **Suppose for a contradiction** that at the end of the execution $c(i) < |P|$

- Then, $\deg(i) \leq |P| \cdot \ln n + 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1} < \text{const} \cdot |P| \cdot \ln n = \delta$, **a contradiction.**

Guarantees of the algorithm

$$Z(i) \triangleq \deg(i) - 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1}$$

- For any node i, it always holds $Z(i) \leq |P| \cdot \ln n$

- **Suppose for a contradiction** that at the end of the execution $c(i) < |P|$

- Then, $\deg(i) \leq |P| \cdot \ln n + 2 \cdot \sum_{j=1}^{c(i)} \frac{|P|}{|P| - j + 1} < \text{const} \cdot |P| \cdot \ln n = \delta$, **a contradiction.**

Actually (ln n + ln |P|), but a modification of the algorithm can fix it

The Plan

- Problem definition ✅
- Previous results ✅
- Min-degree known a priori
  - ➜ randomized O(log n)-competitive solution ✅
  - ➜ deterministic O(log n)-competitive solution ✅
- **Challenges for unknown min-degree**

Problems with Unknown Min-degree

The presented technique requires the knowledge of min-degree:

- for choosing palette
- for down-scaling of $Z(i)$ for exp-function ← this is not a mere technicality!

Our approach:

- Each node has its own phase p (new phase when its degree doubles)
- **Main obstacle:** this results in **infinitely many** variables $Z(i, p)$
- We show that we can control **weighted averages** of $Z(i, p)$

Outlook

	Lower bound	Upper bound
randomized	$O(\log n / \log \log n)$ [1]	$O(\log^2 n)$ [1]
deterministic	$O(\log n / \log \log n)$ [1]	$O(\log^2 n)$

Open questions:

- Close the (randomized and deterministic) gaps
- What features of an online problem make randomization unnecessary?

Thank you!