Towards Optimal Distributed Edge Coloring with Fewer Colors

Manuel Jakob · Yannic Maus · **Florian Schager**

30. October 2025

TU Graz

FWF Österreichischer Wissenschaftsfonds

This research was funded in whole or in part by the Austrian Science Fund (FWF) [10.55776/P36280] & [10.55776/I6915].

Towards Optimal Distributed Edge Coloring with one fewer color

Manuel Jakob · Yannic Maus · **Florian Schager**

30. October 2025

TU Graz

FWF Österreichischer Wissenschaftsfonds

This research was funded in whole or in part by the Austrian Science Fund (FWF) [10.55776/P36280] & [10.55776/I6915].

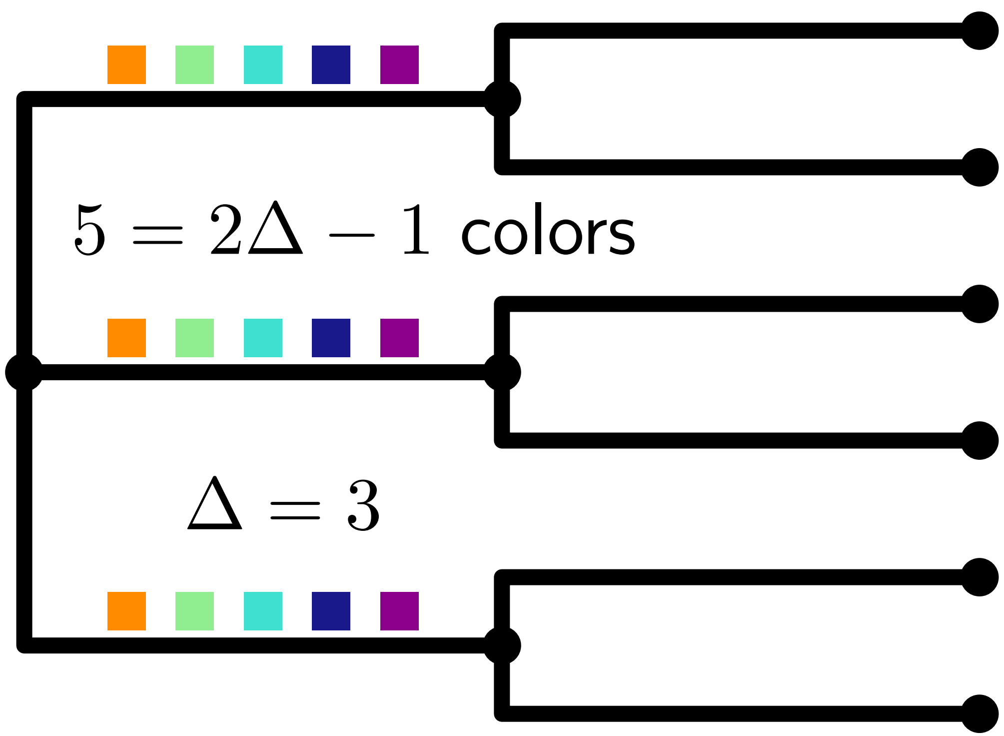
Contributions

Motivation	huge difference between greedy and non-greedy problems	

Contributions

Motivation

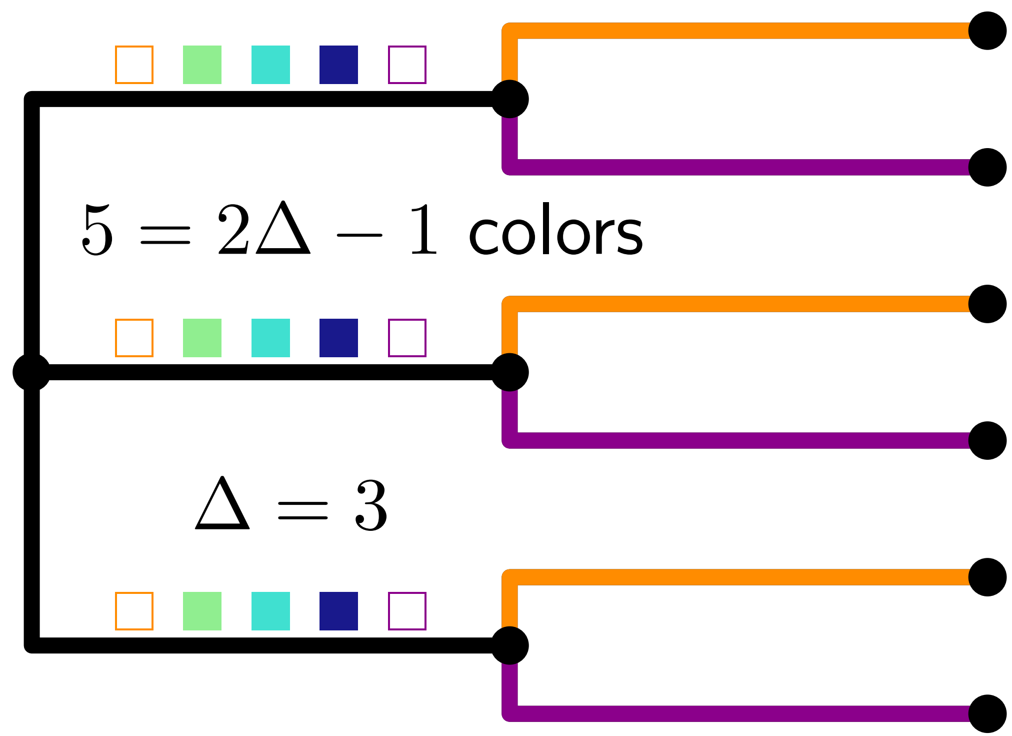
huge difference between greedy and non-greedy problems



Goal: assign colors to edges; adjacent edges receive different colors

Contributions

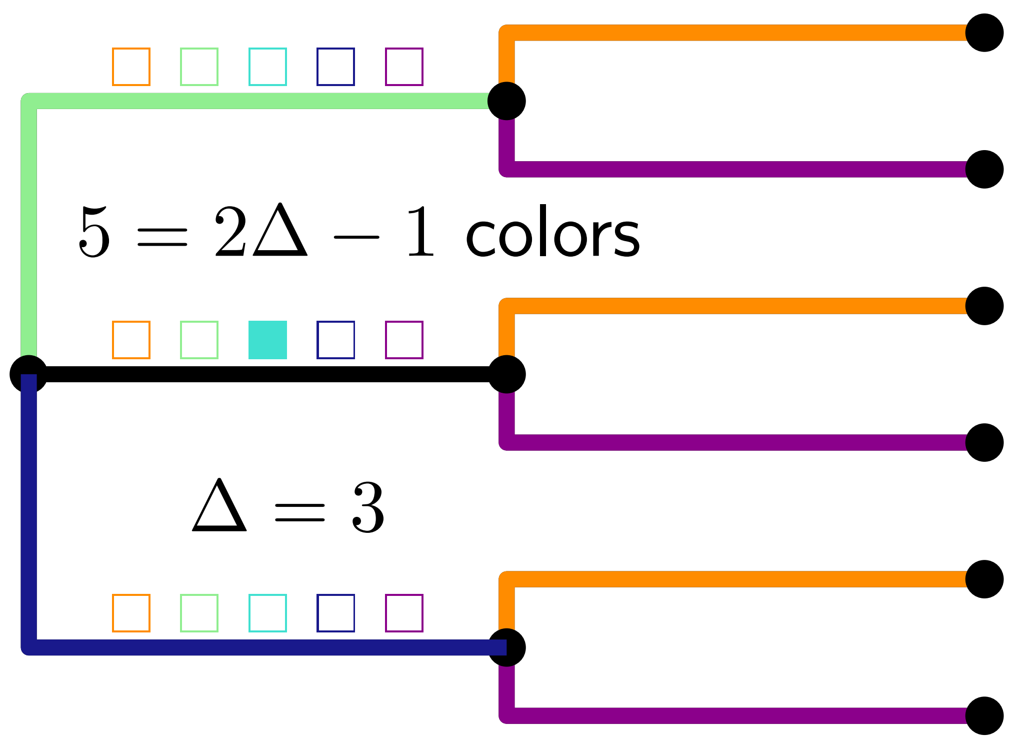
Motivation	huge difference between greedy and non-greedy problems	



Goal: assign colors to edges; adjacent edges receive different colors

Contributions

Motivation	huge difference between greedy and non-greedy problems	



Goal: assign colors to edges; adjacent edges receive different colors

Contributions

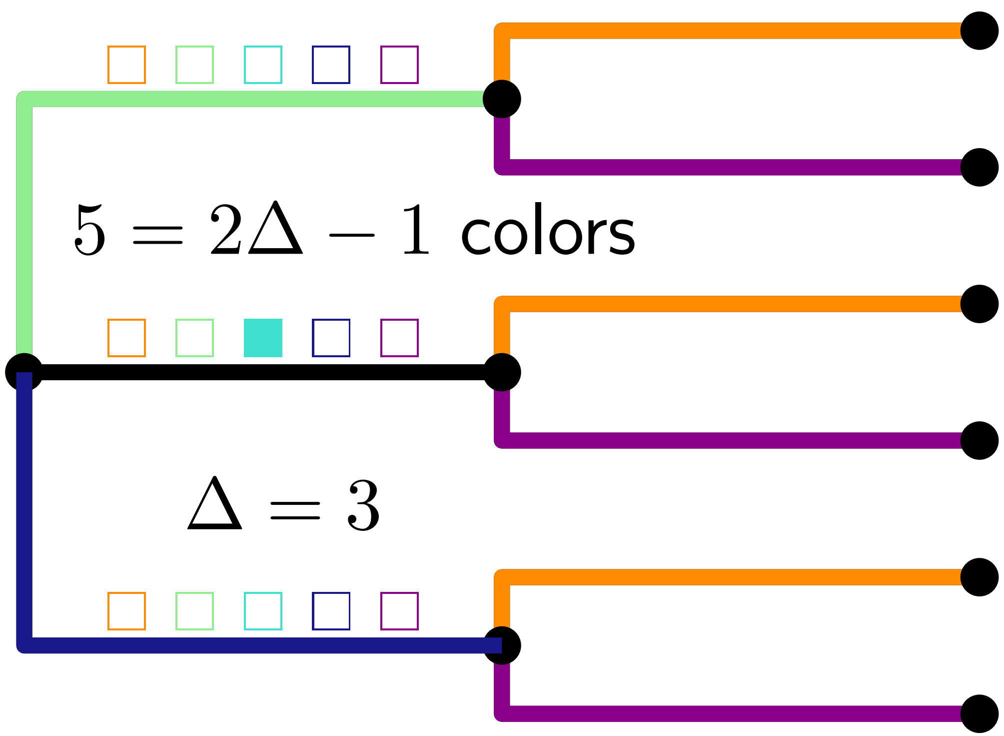
Motivation

huge difference between greedy and non-greedy problems

$2\Delta - 1$

EASY

$\mathcal{O}(\log^\star n)$



Goal: assign colors to edges; adjacent edges receive different colors

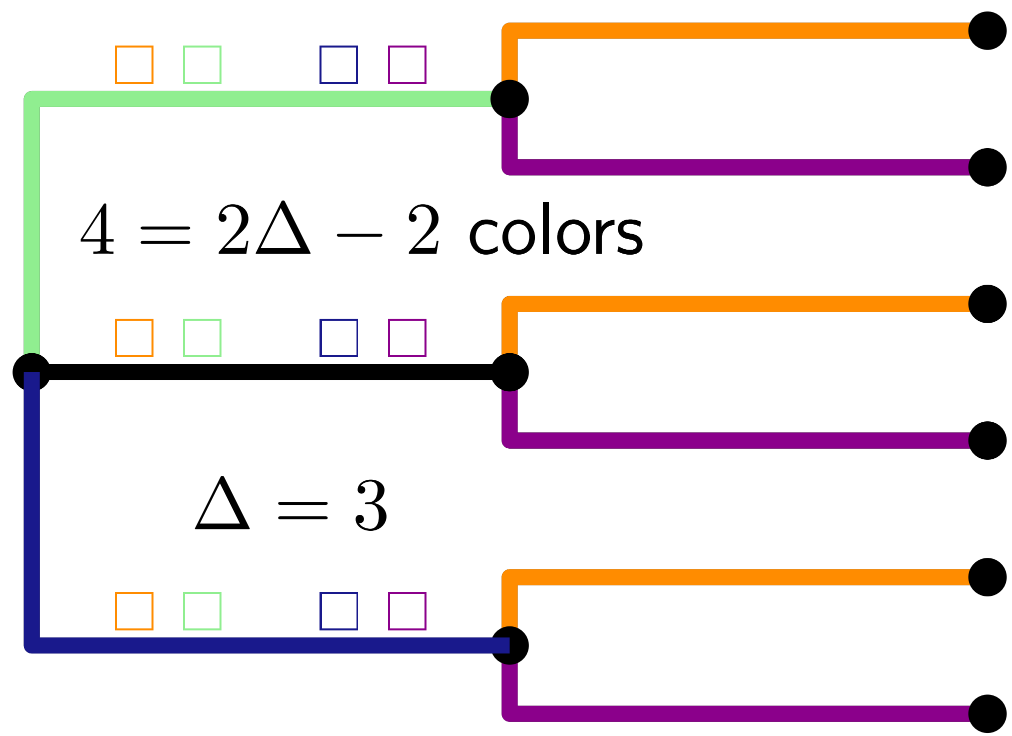
Contributions

Motivation	huge difference between greedy and non-greedy problems	

$2\Delta - 1$

EASY

$\mathcal{O}(\log^\star n)$



$2\Delta - 2$

HARD

$\Omega(\log n)$

Goal: assign colors to edges; adjacent edges receive different colors

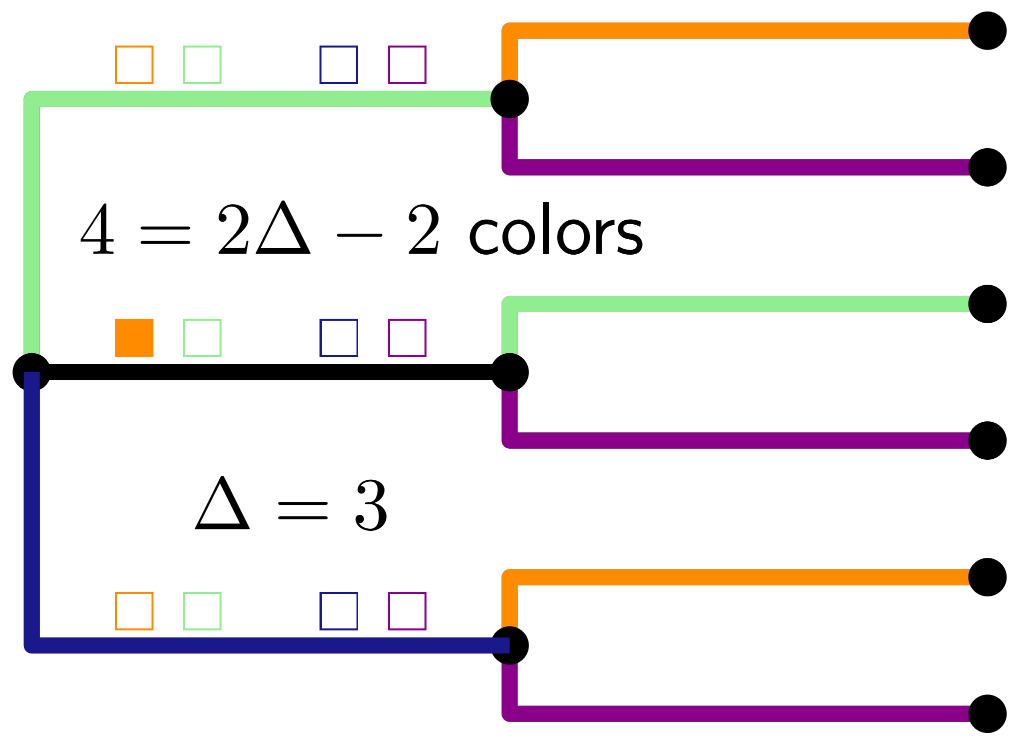
Contributions

Motivation	huge difference between greedy and non-greedy problems	

$2\Delta - 1$

EASY

$\mathcal{O}(\log^\star n)$



$2\Delta - 2$

HARD

$\Omega(\log n)$

Goal: assign colors to edges; adjacent edges receive different colors

Contributions

Motivation

huge difference between greedy and non-greedy problems

LOCAL

$2\Delta - 1$	$2\Delta - 2$
EASY	**HARD**
$\mathcal{O}(\log^\star n)$	$\Omega(\log n)$

Technique

Question: When can we extend partial $(2\Delta - 2)$-colorings?

extendable?

Contributions

Motivation

huge difference between greedy and non-greedy problems

LOCAL

$2\Delta - 1$	$2\Delta - 2$
EASY	**HARD**
$\mathcal{O}(\log^\star n)$	$\Omega(\log n)$

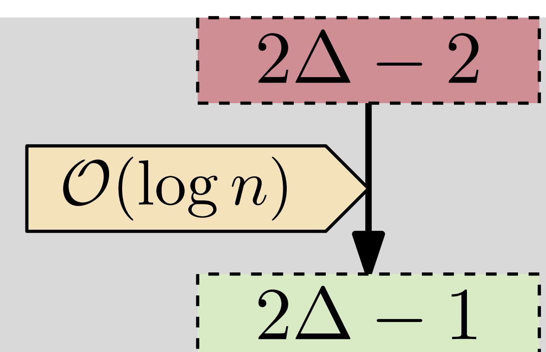
Technique

Question: When can we extend partial $(2\Delta - 2)$-colorings?

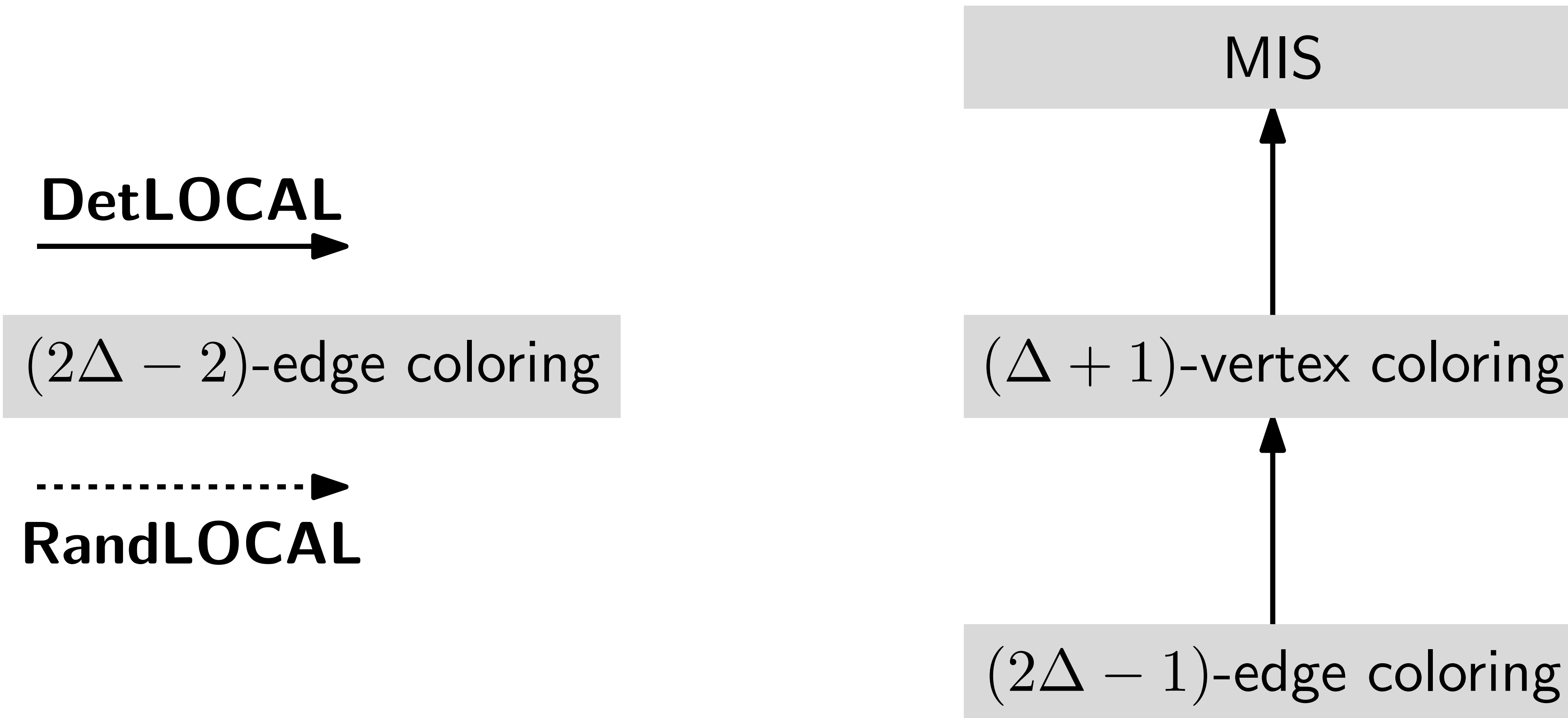
extendable?

Result

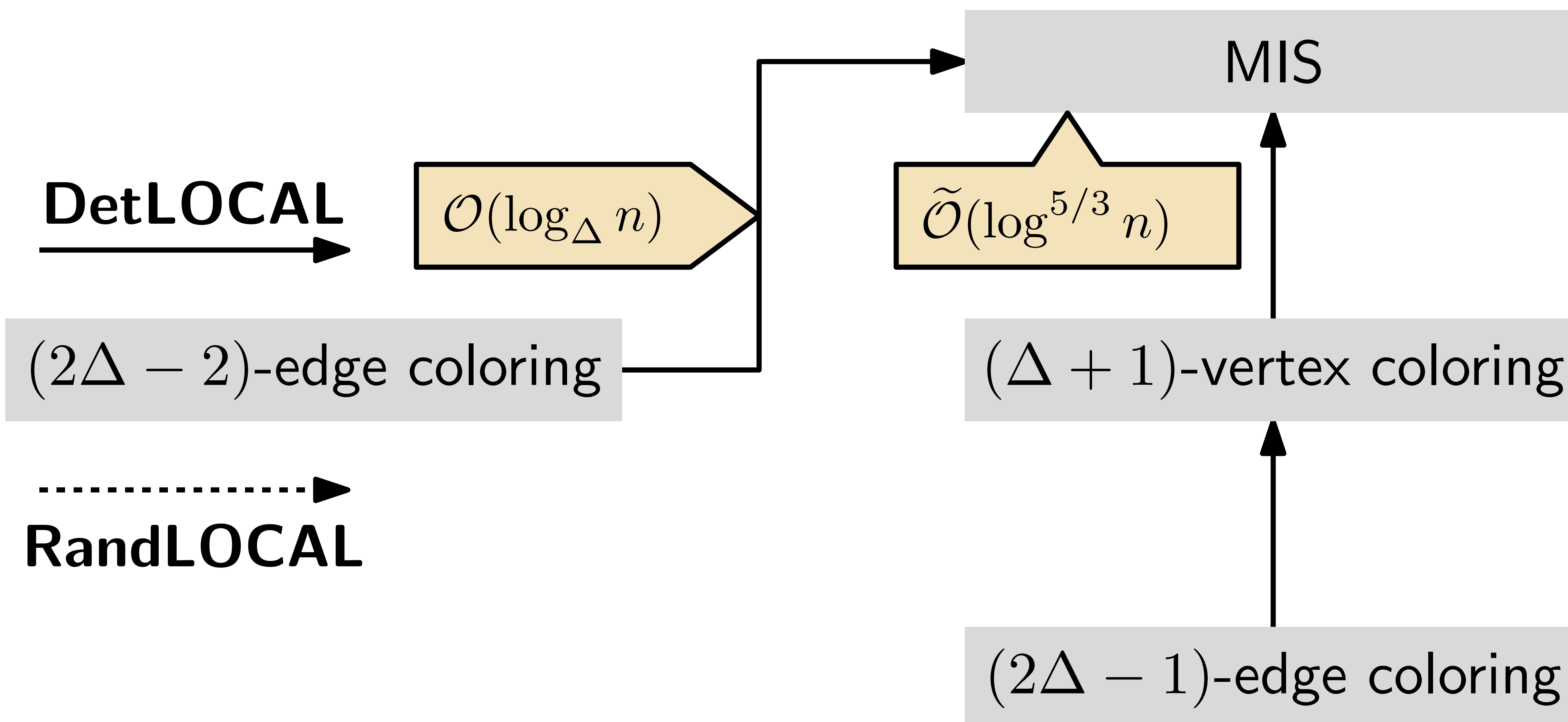
optimal reduction from non-greedy edge coloring to greedy edge coloring



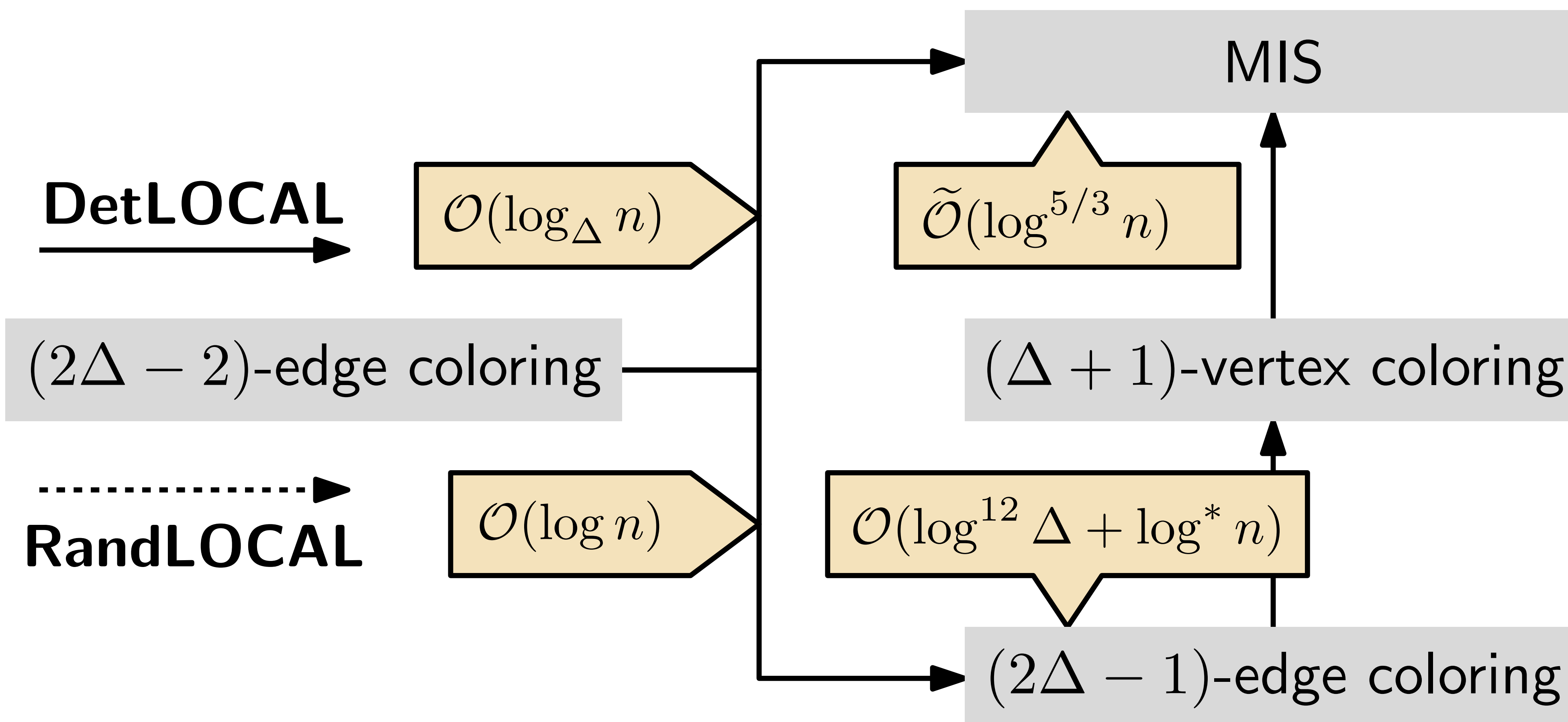
Reductions


MIS
DetLOCAL
$(2\Delta - 2)$-edge coloring
$(\Delta + 1)$-vertex coloring
RandLOCAL
$(2\Delta - 1)$-edge coloring

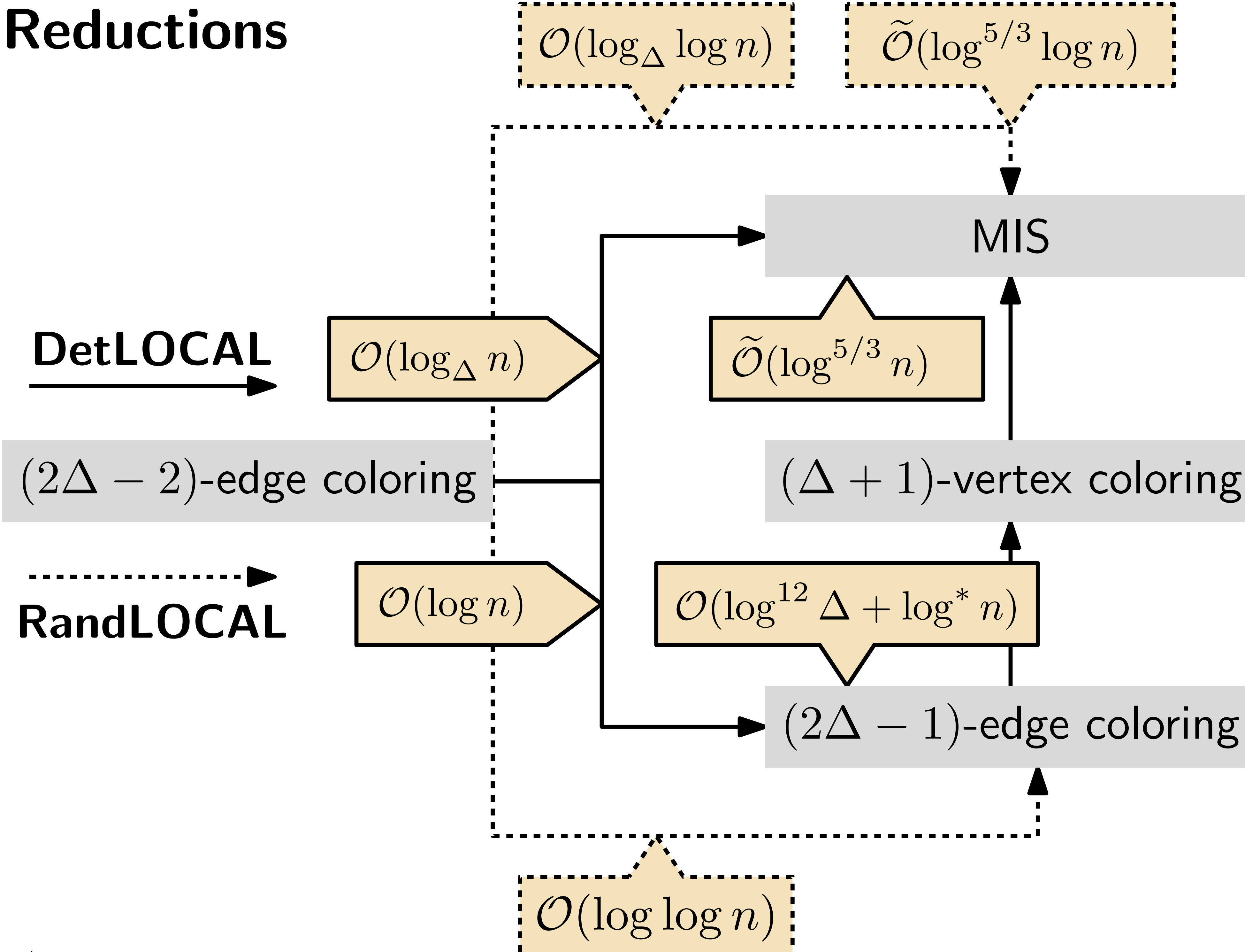
Reductions


MIS
DetLOCAL
$\mathcal{O}(\log_\Delta n)$
$\widetilde{\mathcal{O}}(\log^{5/3} n)$
$(2\Delta - 2)$-edge coloring
$(\Delta + 1)$-vertex coloring
RandLOCAL
$(2\Delta - 1)$-edge coloring

Reductions

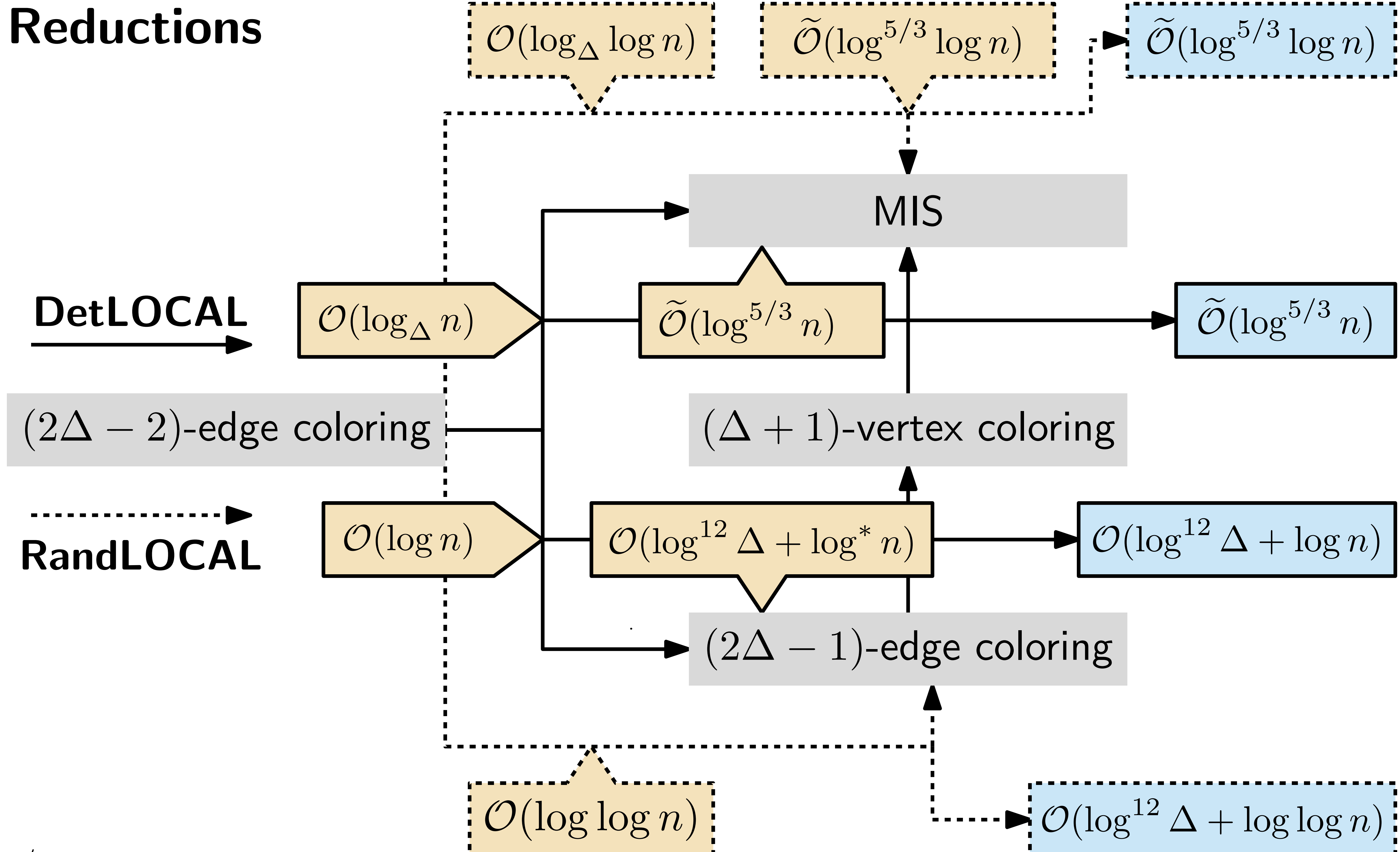

MIS
DetLOCAL
$\mathcal{O}(\log_\Delta n)$
$\widetilde{\mathcal{O}}(\log^{5/3} n)$
$(2\Delta - 2)$-edge coloring
$(\Delta + 1)$-vertex coloring
$\mathcal{O}(\log n)$
$\mathcal{O}(\log^{12} \Delta + \log^* n)$
RandLOCAL
$(2\Delta - 1)$-edge coloring

Reductions



$\mathcal{O}(\log_\Delta \log n)$
$\widetilde{\mathcal{O}}(\log^{5/3} \log n)$
MIS
DetLOCAL
$\mathcal{O}(\log_\Delta n)$
$\widetilde{\mathcal{O}}(\log^{5/3} n)$
$(2\Delta - 2)$-edge coloring
$(\Delta + 1)$-vertex coloring
RandLOCAL
$\mathcal{O}(\log n)$
$\mathcal{O}(\log^{12} \Delta + \log^* n)$
$(2\Delta - 1)$-edge coloring
$\mathcal{O}(\log \log n)$

Reductions



DetLOCAL
RandLOCAL
$\mathcal{O}(\log_\Delta \log n)$
$\widetilde{\mathcal{O}}(\log^{5/3} \log n)$
$\widetilde{\mathcal{O}}(\log^{5/3} \log n)$
MIS
$\mathcal{O}(\log_\Delta n)$
$\widetilde{\mathcal{O}}(\log^{5/3} n)$
$\widetilde{\mathcal{O}}(\log^{5/3} n)$
$(2\Delta - 2)$-edge coloring
$(\Delta + 1)$-vertex coloring
$\mathcal{O}(\log n)$
$\mathcal{O}(\log^{12} \Delta + \log^* n)$
$\mathcal{O}(\log^{12} \Delta + \log n)$
$(2\Delta - 1)$-edge coloring
$\mathcal{O}(\log \log n)$
$\mathcal{O}(\log^{12} \Delta + \log \log n)$

Runtimes

deterministic | randomized

Δ-regime	NEW	OLD	Sources
general graphs			
$\Delta \leq 2^{\log^{1/12} n}$			
$\Delta \leq 2^{\log^{1/12} \log n}$			

Runtimes

deterministic | randomized

Δ-regime	NEW	OLD	Sources
general graphs	$\widetilde{\mathcal{O}}(\log^{5/3} n)$	$\widetilde{\mathcal{O}}(\log^{19/9} n)$	[1,2,3]
	$\widetilde{\mathcal{O}}(\log^{5/3} \log n)$	$\widetilde{\mathcal{O}}(\log^{8/3} \log n)$	[1]
$\Delta \leq 2^{\log^{1/12} n}$			
$\Delta \leq 2^{\log^{1/12} \log n}$			

1: [Brandt, Bourreau & Nolin, STOC'25]

2: [Ghaffari, Kuhn, FOCS'21]

3: [Ghaffari, Grunau, FOCS'24]

Runtimes

deterministic	randomized

Δ-regime	**NEW**	**OLD**	**Sources**
general graphs	$\widetilde{\mathcal{O}}(\log^{5/3} n)$	$\widetilde{\mathcal{O}}(\log^{19/9} n)$	[1,2,3]
	$\widetilde{\mathcal{O}}(\log^{5/3} \log n)$	$\widetilde{\mathcal{O}}(\log^{8/3} \log n)$	[1]
$\Delta \leq 2^{\log^{1/12} n}$	$\mathcal{O}(\log n)$	$\widetilde{\mathcal{O}}(\log^{7/6} n)$	[1, 4]
$\Delta \leq 2^{\log^{1/12} \log n}$	$\mathcal{O}(\log \log n)$	$\mathcal{O}(\log^{7/6} \log n \cdot \log^{\star} n)$	[1, 4]

1: [Brandt, Bourreau & Nolin, STOC'25]

2: [Ghaffari, Kuhn, FOCS'21]

3: [Ghaffari, Grunau, FOCS'24]

4: [FGGKR, SODA'23]

Runtimes

deterministic | randomized

Δ-regime	NEW	OLD	Sources
general graphs (deterministic)	$\widetilde{\mathcal{O}}(\log^{5/3} n)$	$\widetilde{\mathcal{O}}(\log^{19/9} n)$	[1,2,3]
general graphs (randomized)	$\widetilde{\mathcal{O}}(\log^{5/3} \log n)$	$\widetilde{\mathcal{O}}(\log^{8/3} \log n)$	[1]
$\Delta \leq 2^{\log^{1/12} n}$	$\mathcal{O}(\log n)$	$\widetilde{\mathcal{O}}(\log^{7/6} n)$	[1, 4]
$\Delta \leq 2^{\log^{1/12} \log n}$	$\mathcal{O}(\log \log n)$	$\mathcal{O}(\log^{7/6} \log n \cdot \log^\star n)$	[1, 4]

1: [Brandt, Bourreau & Nolin, STOC'25]

2: [Ghaffari, Kuhn, FOCS'21]

3: [Ghaffari, Grunau, FOCS'24]

4: [FGGKR, SODA'23]

After publication: [Brandt, Bourreau & Nolin, SODA'26]

$\mathcal{O}(\log_\Delta n)$ reduction from Δ-vertex coloring to MIS

Contributions

Motivation

huge difference between greedy and non-greedy problems

$2\Delta - 2$	$2\Delta - 1$
$\Omega(\log n)$	$\mathcal{O}(\log^\star n)$
hard	**easy**

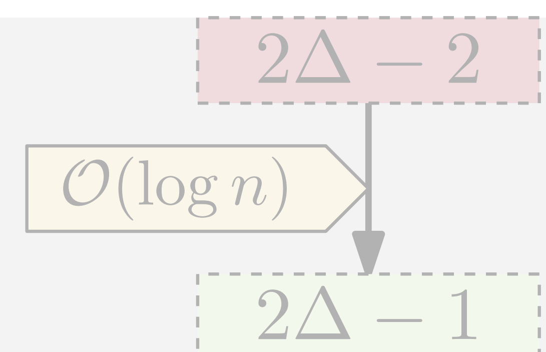
Technique

Question: When can we extend partial $(2\Delta - 2)$-colorings?

extendable?

Result

optimal reduction from greedy edge coloring to non-greedy edge coloring



When can we extend a partial coloring to an uncolored star?

Lemma 1 (reaching for the stars)

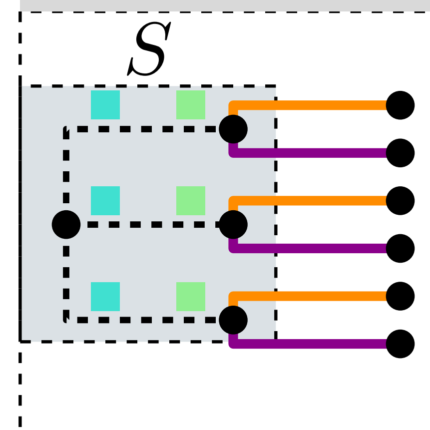
- a graph $G = (V, E)$ and a partial $(2\Delta - 2)$-edge coloring φ
- an uncolored star graph $S \subseteq G$

When can we extend a partial coloring to an uncolored star?

Lemma 1 (reaching for the stars)

- a graph $G = (V, E)$ and a partial $(2\Delta - 2)$-edge coloring φ
- an uncolored star graph $S \subseteq G$

not extendable:

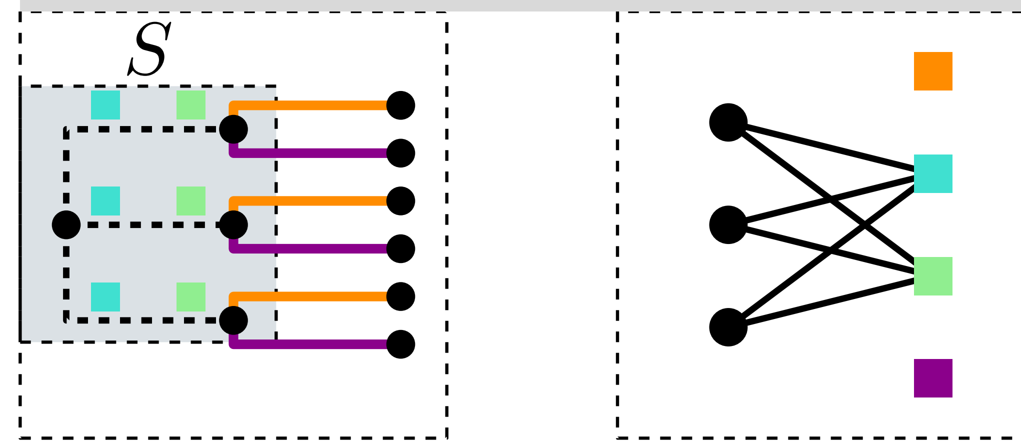


When can we extend a partial coloring to an uncolored star?

Lemma 1 (reaching for the stars)

- a graph $G = (V, E)$ and a partial $(2\Delta - 2)$-edge coloring φ
- an uncolored star graph $S \subseteq G$

not extendable:

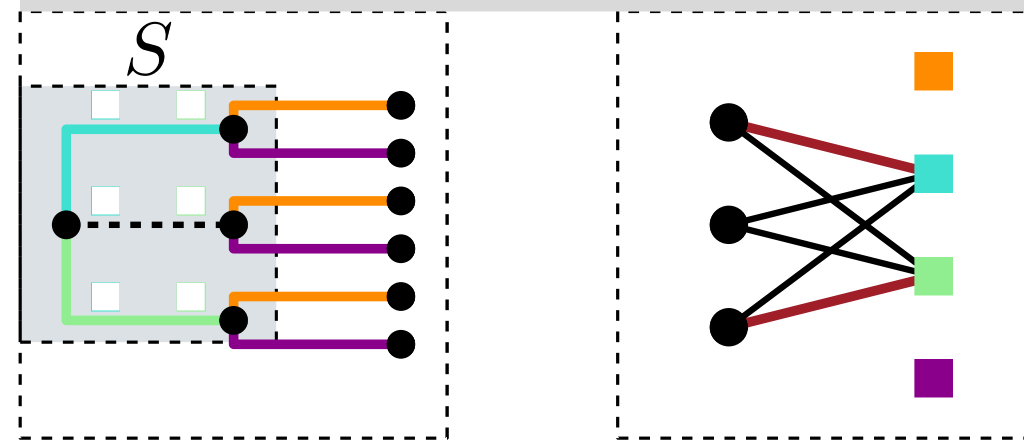


When can we extend a partial coloring to an uncolored star?

Lemma 1 (reaching for the stars)

- a graph $G = (V, E)$ and a partial $(2\Delta - 2)$-edge coloring φ
- an uncolored star graph $S \subseteq G$

not extendable:

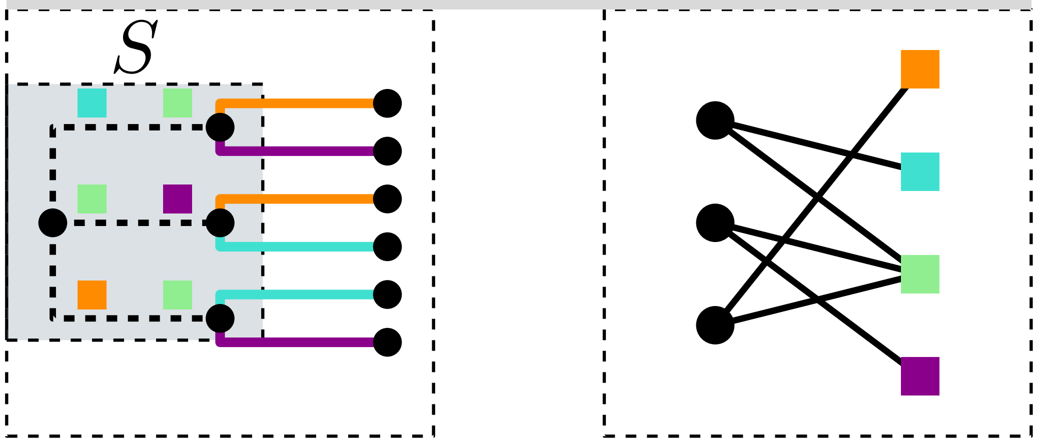


When can we extend a partial coloring to an uncolored star?

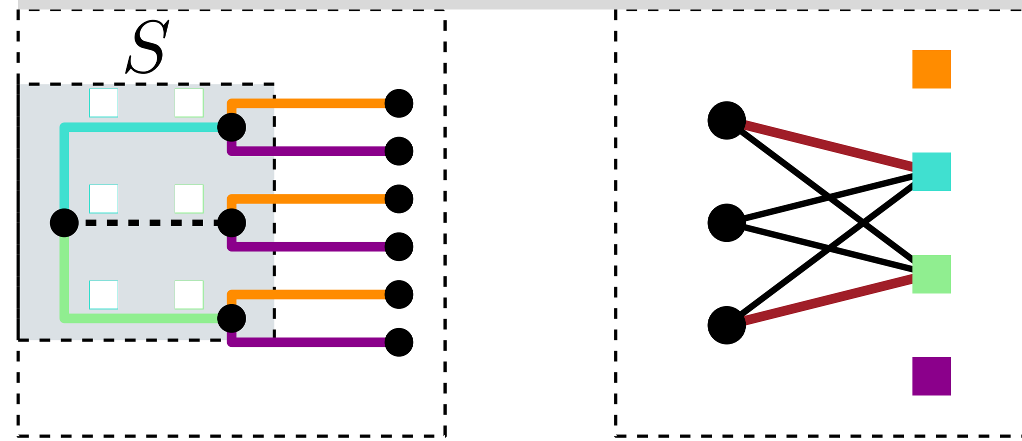
Lemma 1 (reaching for the stars)

- a graph $G = (V, E)$ and a partial $(2\Delta - 2)$-edge coloring φ
- an uncolored star graph $S \subseteq G$

extendable:



not extendable:

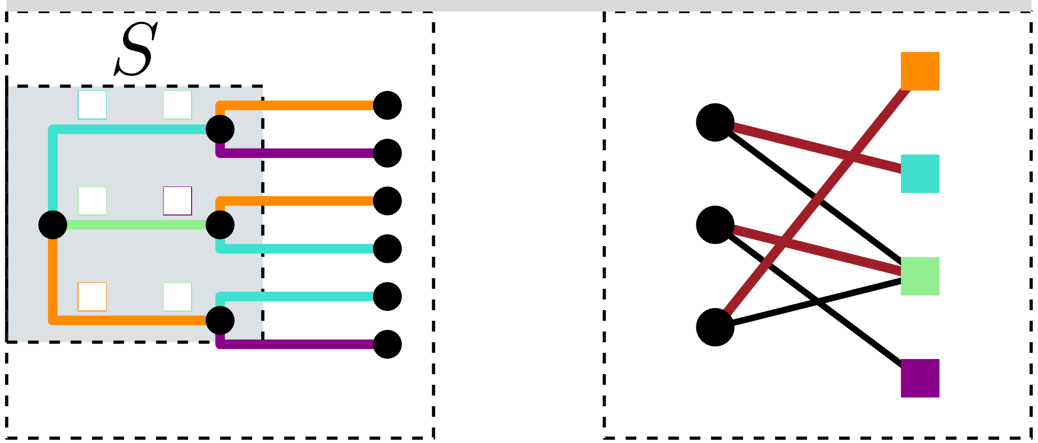


When can we extend a partial coloring to an uncolored star?

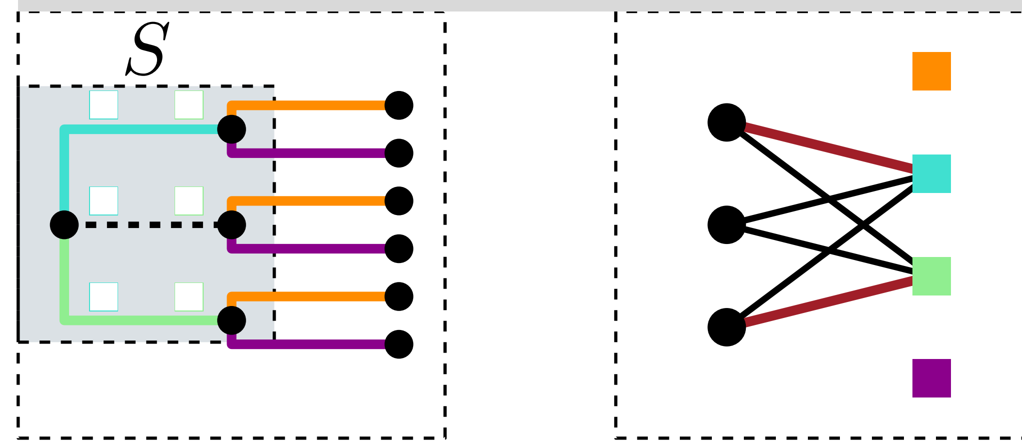
Lemma 1 (reaching for the stars)

- a graph $G = (V, E)$ and a partial $(2\Delta - 2)$-edge coloring φ
- an uncolored star graph $S \subseteq G$

extendable:



not extendable:

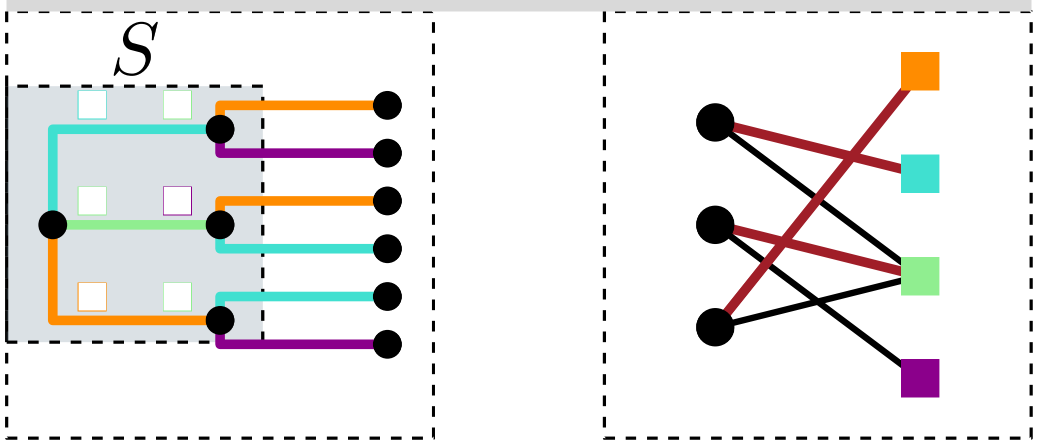


When can we extend a partial coloring to an uncolored star?

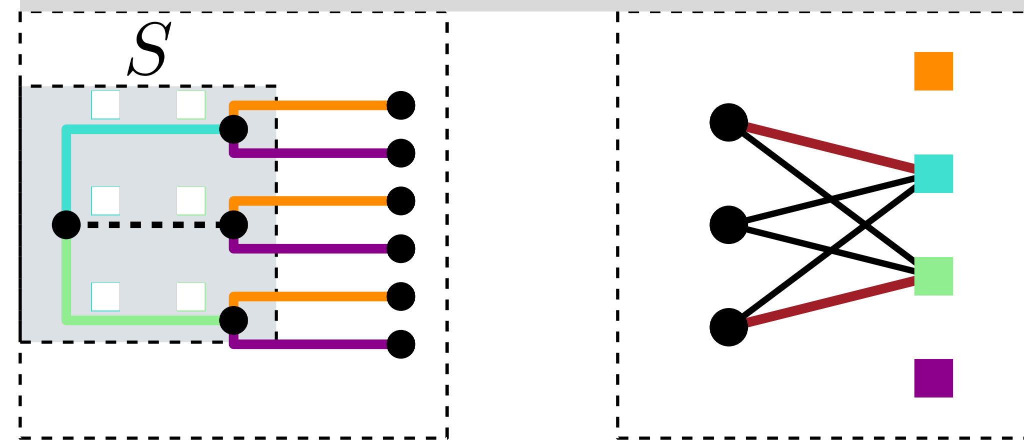
Lemma 1 (reaching for the stars)

- a graph $G = (V, E)$ and a partial $(2\Delta - 2)$-edge coloring φ
- an uncolored star graph $S \subseteq G$

extendable:



not extendable:



Hall's theorem: $\exists\, U$-saturating matching $\iff$ $|N(S)| \geq |S|$ for all $S \subseteq U$

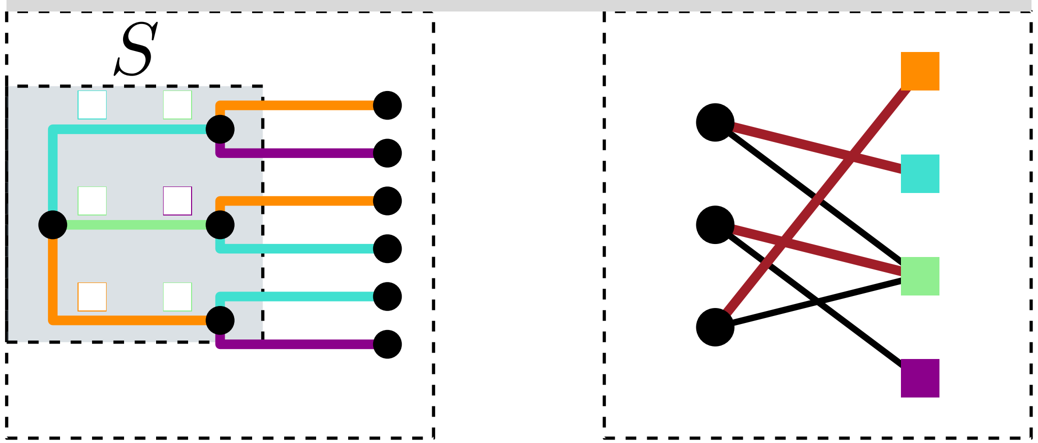
When can we extend a partial coloring to an uncolored star?

Lemma 1 (reaching for the stars)

- a graph $G = (V, E)$ and a partial $(2\Delta - 2)$-edge coloring φ
- an uncolored star graph $S \subseteq G$

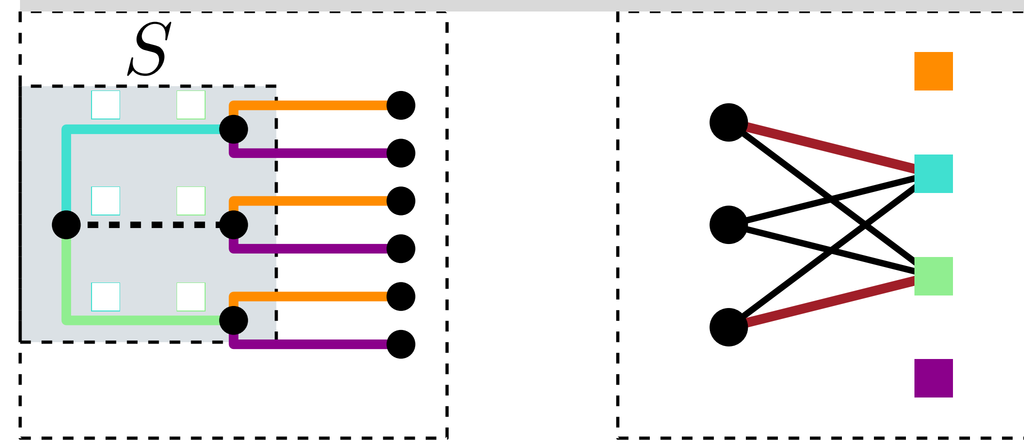
If $|\varphi(N_E(V_S))| \geq \Delta$, then we can extend φ to S.

extendable:



$\varphi(N_E(V_S)) = \{$ ■ , ■ , ■ $\}$

not extendable:



$\varphi(N_E(V_S)) = \{$ ■ , ■ $\}$

Hall's theorem: $\exists\, U$-saturating matching $\iff$ $|N(S)| \geq |S|$ for all $S \subseteq U$

When can we extend a partial coloring to an uncolored tree?

Lemma 2 (colorful leaves make any tree happy)

- Δ-regular graph G, rooted tree $T \ni r$
- $\varphi : E \setminus E_T \to [2\Delta - 2]$ partial edge coloring
- $V_k := \{v \in V_T : \mathrm{dist}(v, r) = k\}$

When can we extend a partial coloring to an uncolored tree?

Lemma 2 (colorful leaves make any tree happy)

- Δ-regular graph G, rooted tree $T \ni r$
- $\varphi : E \setminus E_T \to [2\Delta - 2]$ partial edge coloring
- $V_k := \{v \in V_T : \mathrm{dist}(v, r) = k\}$

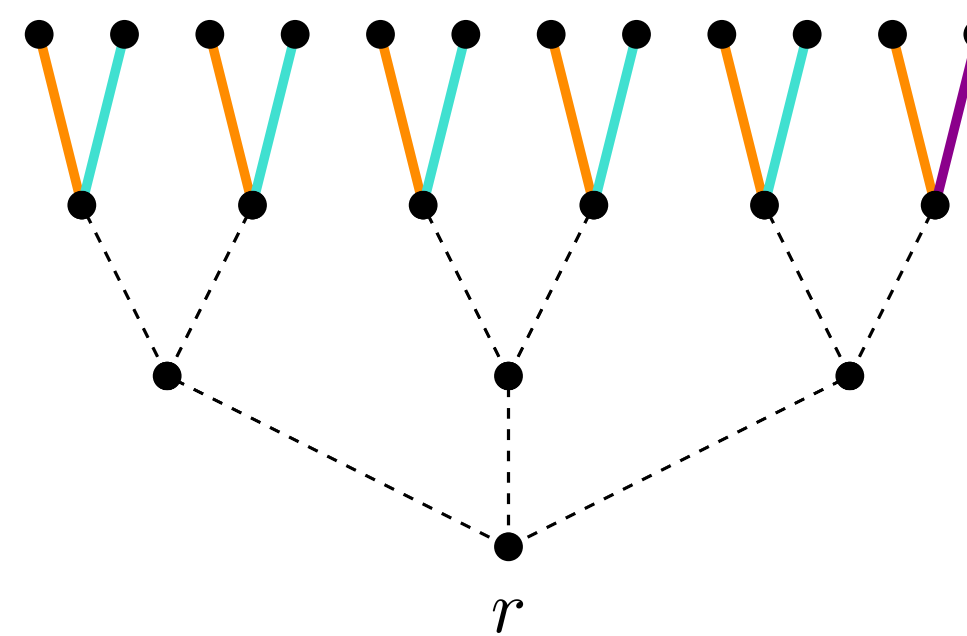
If $\exists k \in \mathbb{N} : |\varphi(N_E(V_k))| \geq \Delta$, then we can extend φ to T.

When can we extend a partial coloring to an uncolored tree?

Lemma 2 (colorful leaves make any tree happy)

- Δ-regular graph G, rooted tree $T \ni r$
- $\varphi : E \setminus E_T \to [2\Delta - 2]$ partial edge coloring
- $V_k := \{v \in V_T : \mathrm{dist}(v, r) = k\}$

If $\exists k \in \mathbb{N} : |\varphi(N_E(V_k))| \geq \Delta$, then we can extend φ to T.



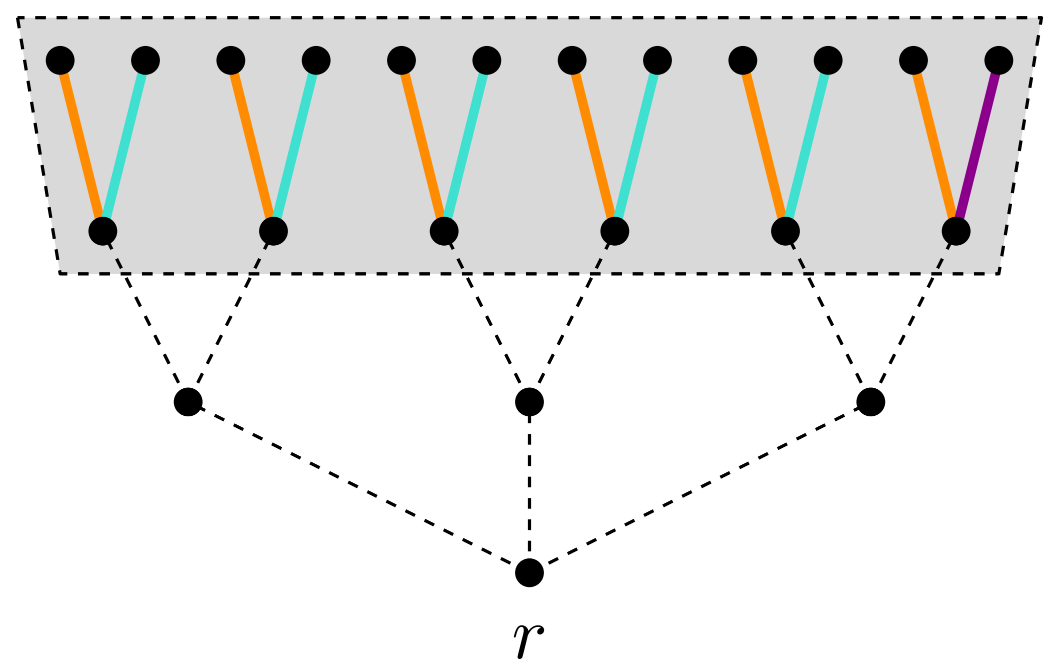
When can we extend a partial coloring to an uncolored tree?

Lemma 2 (colorful leaves make any tree happy)

- Δ-regular graph G, rooted tree $T \ni r$
- $\varphi : E \setminus E_T \to [2\Delta - 2]$ partial edge coloring
- $V_k := \{v \in V_T : \mathrm{dist}(v, r) = k\}$

If $\exists k \in \mathbb{N} : |\varphi(N_E(V_k))| \geq \Delta$, then we can extend φ to T.

$|\varphi(N_E(V_2))| \geq 3$ ✓



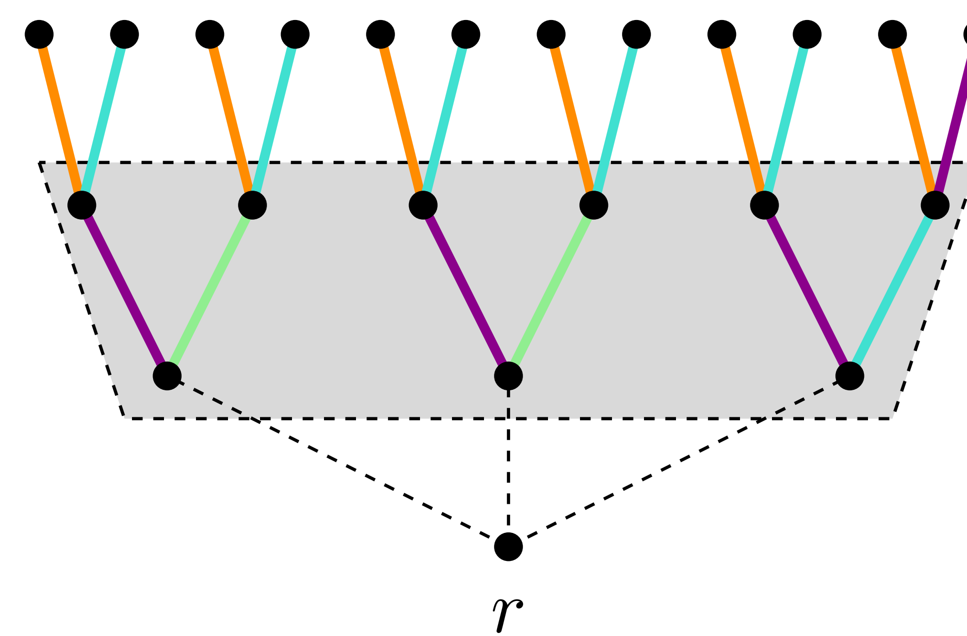
When can we extend a partial coloring to an uncolored tree?

Lemma 2 (colorful leaves make any tree happy)

- Δ-regular graph G, rooted tree $T \ni r$
- $\varphi : E \setminus E_T \to [2\Delta - 2]$ partial edge coloring
- $V_k := \{v \in V_T : \mathrm{dist}(v, r) = k\}$

If $\exists k \in \mathbb{N} : |\varphi(N_E(V_k))| \geq \Delta$, then we can extend φ to T.

$|\varphi(N_E(V_1))| \geq 3$ ✓



When can we extend a partial coloring to an uncolored tree?

Lemma 2 (colorful leaves make any tree happy)

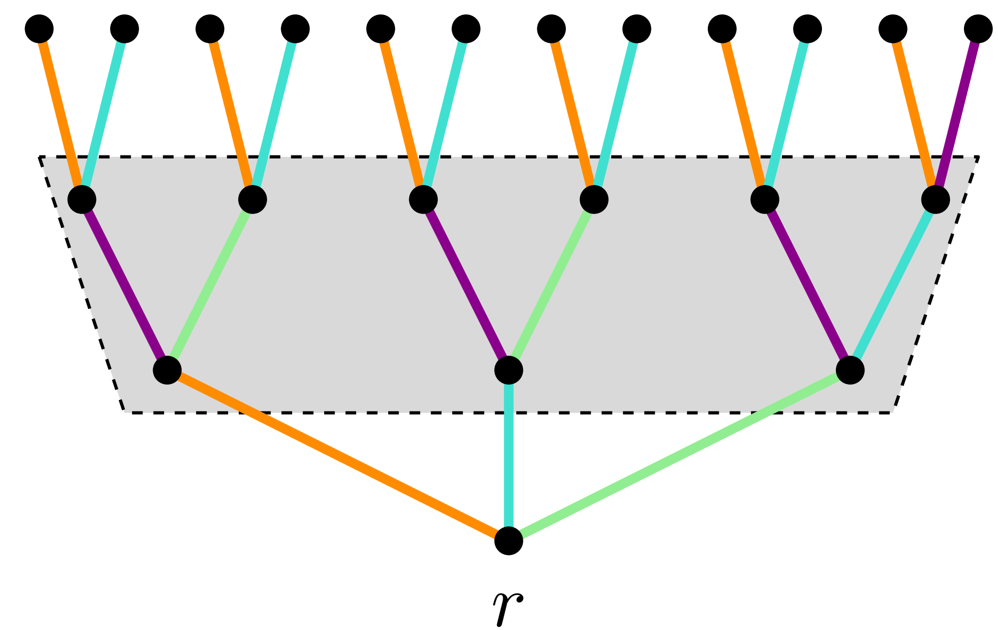
- Δ-regular graph G, rooted tree $T \ni r$
- $\varphi : E \setminus E_T \to [2\Delta - 2]$ partial edge coloring
- $V_k := \{v \in V_T : \mathrm{dist}(v, r) = k\}$

If $\exists k \in \mathbb{N} : |\varphi(N_E(V_k))| \geq \Delta$**, then we can extend** φ **to** T**.**

$|\varphi(N_E(V_1))| \geq 3$ ✓

Lemma 1 ↓

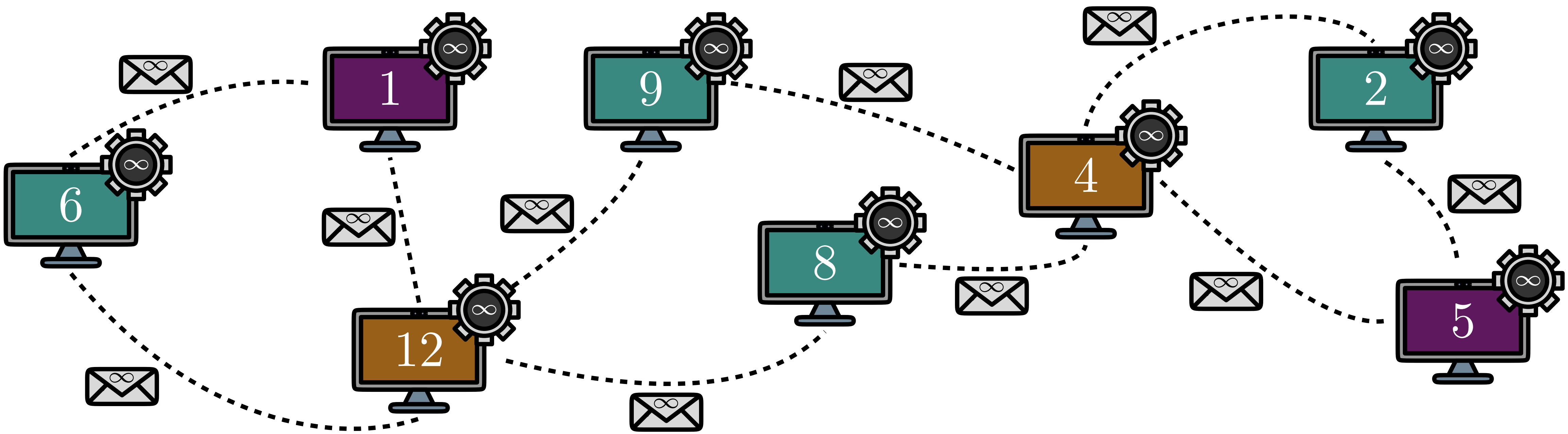
φ is extendable to T ✓



Model of computation

Definition (LOCAL model) **[Linial, 1992]**

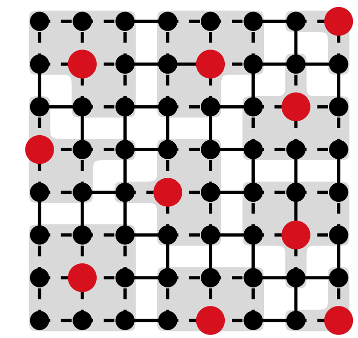
- communication network: **undirected graph**
- nodes have **unique IDs**
- communication happens in **synchronous rounds**
- message size and local computation is unlimited
- time complexity: **# of synchronous rounds**



High level overview of our algorithm

Phase 1: Partition vertices into clusters

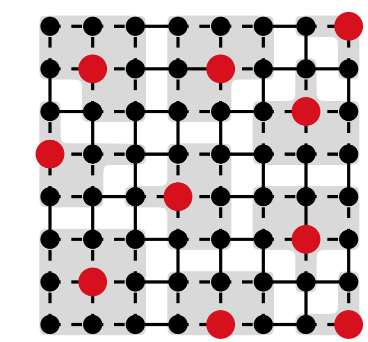
- compute MIS on power graph G^k
- every vertex joins the cluster of its closest MIS-node
- compute $(2\Delta - 3)$-edge coloring of intercluster edges



High level overview of our algorithm

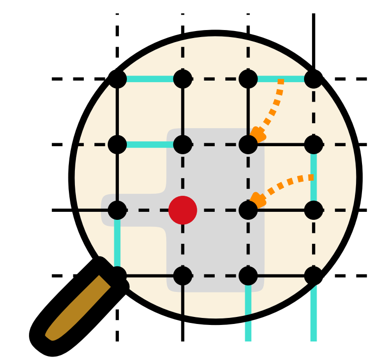
Phase 1: Partition vertices into clusters

- compute MIS on power graph G^k
- every vertex joins the cluster of its closest MIS-node
- compute $(2\Delta - 3)$-edge coloring of intercluster edges



Phase 2: Assign two exclusive edges to each cluster

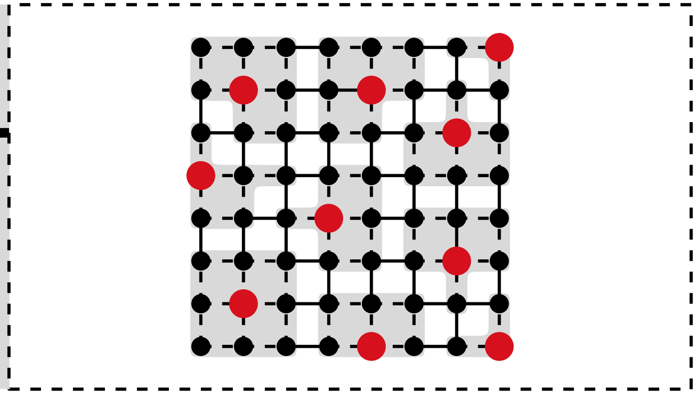
- compute maximal matching of intercluster edges
- modify the matching via hypergraph sinkless orientation



High level overview of our algorithm

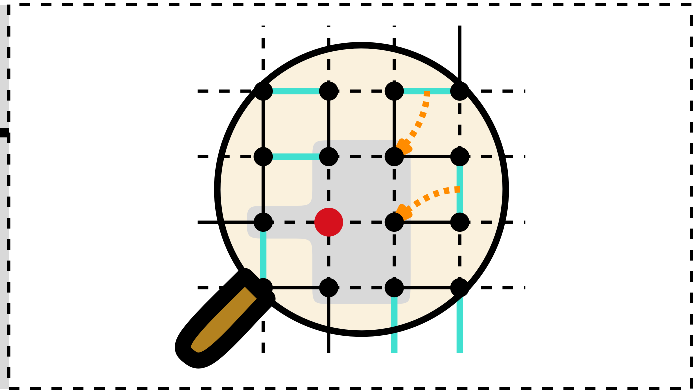
Phase 1: Partition vertices into clusters

- compute MIS on power graph G^k
- every vertex joins the cluster of its closest MIS-node
- compute $(2\Delta - 3)$-edge coloring of intercluster edges



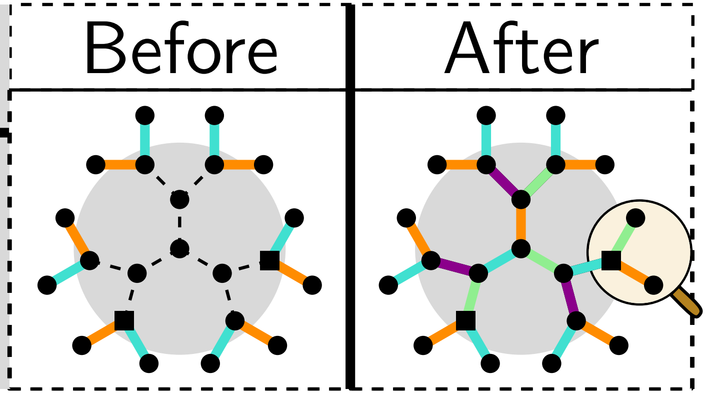
Phase 2: Assign two exclusive edges to each cluster

- compute maximal matching of intercluster edges
- modify the matching via hypergraph sinkless orientation



Phase 3: Switch colors and complete the coloring

- adapt colors of assigned edges in order to
- complete the coloring on the intracluster edges



Phase 1: Partition vertices into clusters

Goal: symmetry breaking + every cluster gets sufficiently many vertices

Technique:

- compute a maximal independent set $\mathcal{I}$ on G^8
- every vertex joins the cluster of its closest node in $\mathcal{I}$
- compute greedy edge coloring of intercluster edges

Result:

- every cluster $C \in \mathcal{C}$ has $\mathrm{diam}(C) \leq 8$
- for every $v \in \mathcal{I} : N^4(v) \subseteq C(v)$
- intercluster edges are colored with just $2\Delta - 3$ colors

Runtime: $T_{\mathrm{MIS}}(n, \mathrm{poly}(\Delta)) + T_{2\Delta-1}(n) = T_{\mathrm{MIS}}(\Delta^2 \cdot n, \mathrm{poly}(\Delta))$

Phase 1: Partition vertices into clusters

Goal: symmetry breaking + every cluster gets sufficiently many vertices

Technique:

- compute a maximal independent set $\mathcal{I}$ on G^8
- every vertex joins the cluster of its closest node in $\mathcal{I}$
- compute greedy edge coloring of intercluster edges

Standard techniques

Result:

- every cluster $C \in \mathcal{C}$ has $\mathrm{diam}(C) \leq 8$
- for every $v \in \mathcal{I} : N^4(v) \subseteq C(v)$
- intercluster edges are colored with just $2\Delta - 3$ colors

Runtime: $T_{\mathrm{MIS}}(n, \mathrm{poly}(\Delta)) + T_{2\Delta-1}(n) = T_{\mathrm{MIS}}(\Delta^2 \cdot n, \mathrm{poly}(\Delta))$

Phase 2: Assign two exclusive edges to each cluster

Goal: get each cluster exclusive access to change colors of two edges

Phase 2: Assign two exclusive edges to each cluster

Goal: get each cluster exclusive access to change colors of two edges

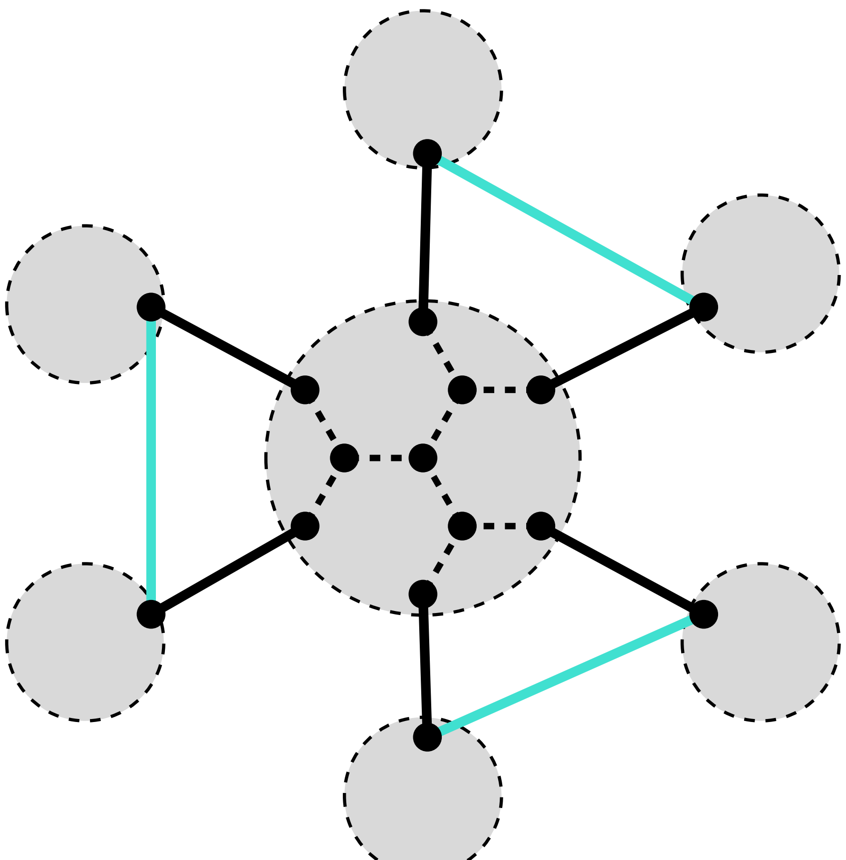
Technique:

- compute a maximal matching of the colored edges
- assign two maximal edges to each cluster via HSO

Phase 2: Assign two exclusive edges to each cluster

Goal: get each cluster exclusive access to change colors of two edges

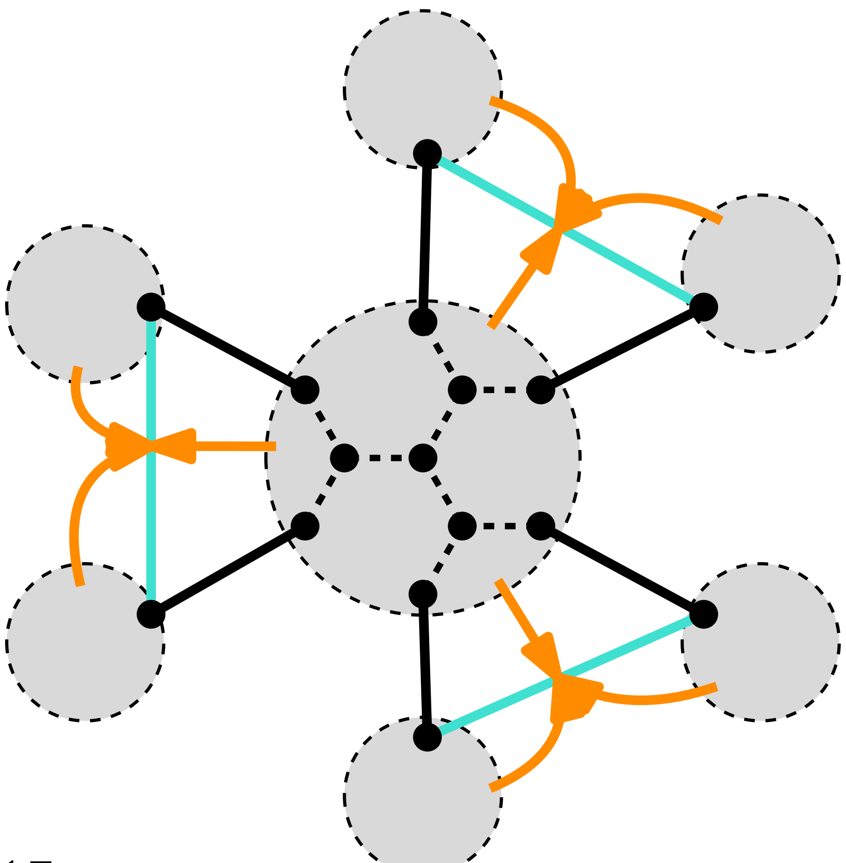
Technique:
- compute a maximal matching of the colored edges
- assign two maximal edges to each cluster via HSO



Phase 2: Assign two exclusive edges to each cluster

Goal: get each cluster exclusive access to change colors of two edges

Technique:
- compute a maximal matching of the colored edges
- assign two maximal edges to each cluster via HSO

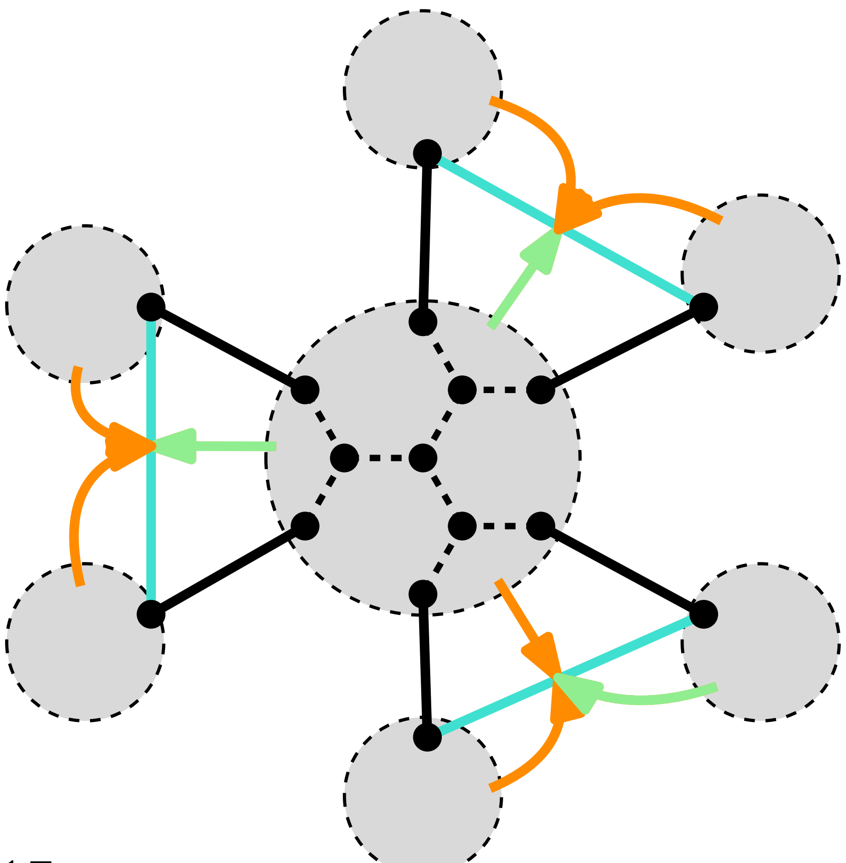


- each cluster sends requests to all matching edges in its 2-hop neighborhood
- each cluster sends at least $\delta := 2\Delta^2$ requests
- each edge in the matching receives at most $r := 2\Delta$ requests

Phase 2: Assign two exclusive edges to each cluster

Goal: get each cluster exclusive access to change colors of two edges

Technique:
- compute a maximal matching of the colored edges
- assign two maximal edges to each cluster via HSO



- each cluster sends requests to all matching edges in its 2-hop neighborhood
- each cluster sends at least $\delta := 2\Delta^2$ requests
- each edge in the matching receives at most $r := 2\Delta$ requests

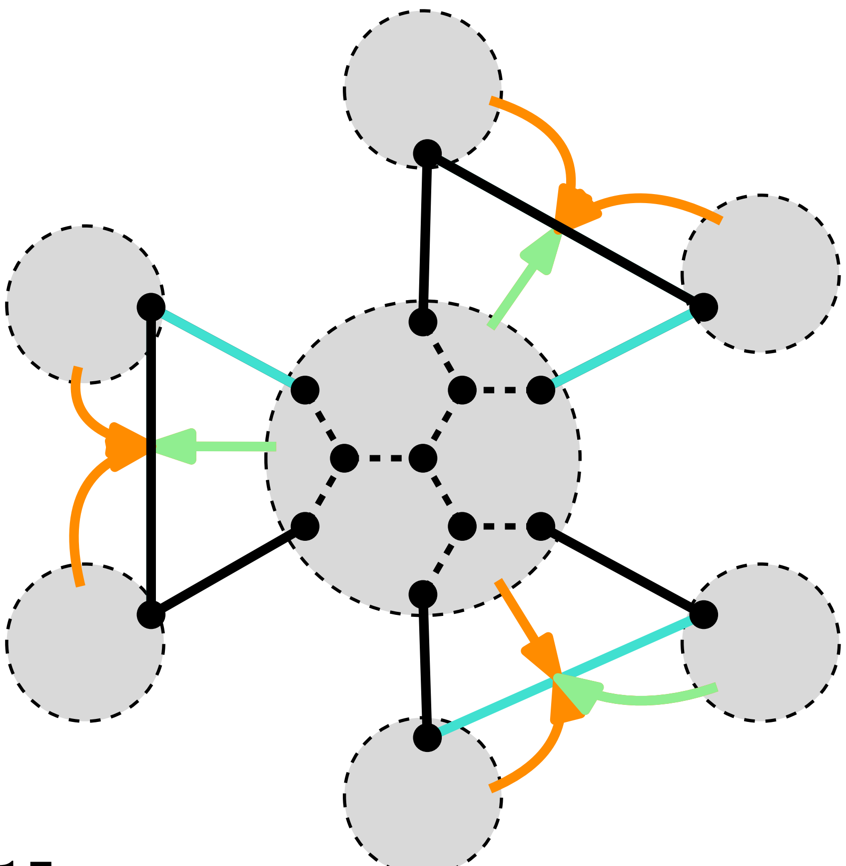
↓

efficiently solvable via HSO

Phase 2: Assign two exclusive edges to each cluster

Goal: get each cluster exclusive access to change colors of two edges

Technique:
- compute a maximal matching of the colored edges
- assign two maximal edges to each cluster via HSO



- each cluster sends requests to all matching edges in its 2-hop neighborhood
- each cluster sends at least $\delta := 2\Delta^2$ requests
- each edge in the matching receives at most $r := 2\Delta$ requests

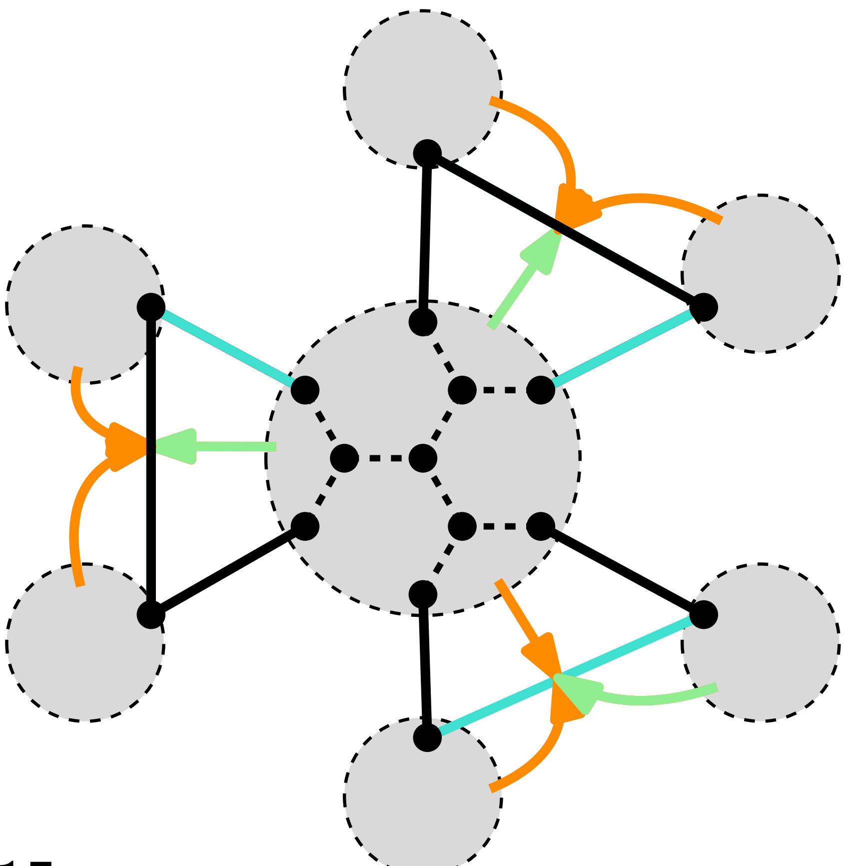
↓

efficiently solvable via HSO

Phase 2: Assign two exclusive edges to each cluster

Goal: get each cluster exclusive access to change colors of two edges

Technique:
- compute a maximal matching of the colored edges
- assign two maximal edges to each cluster via HSO



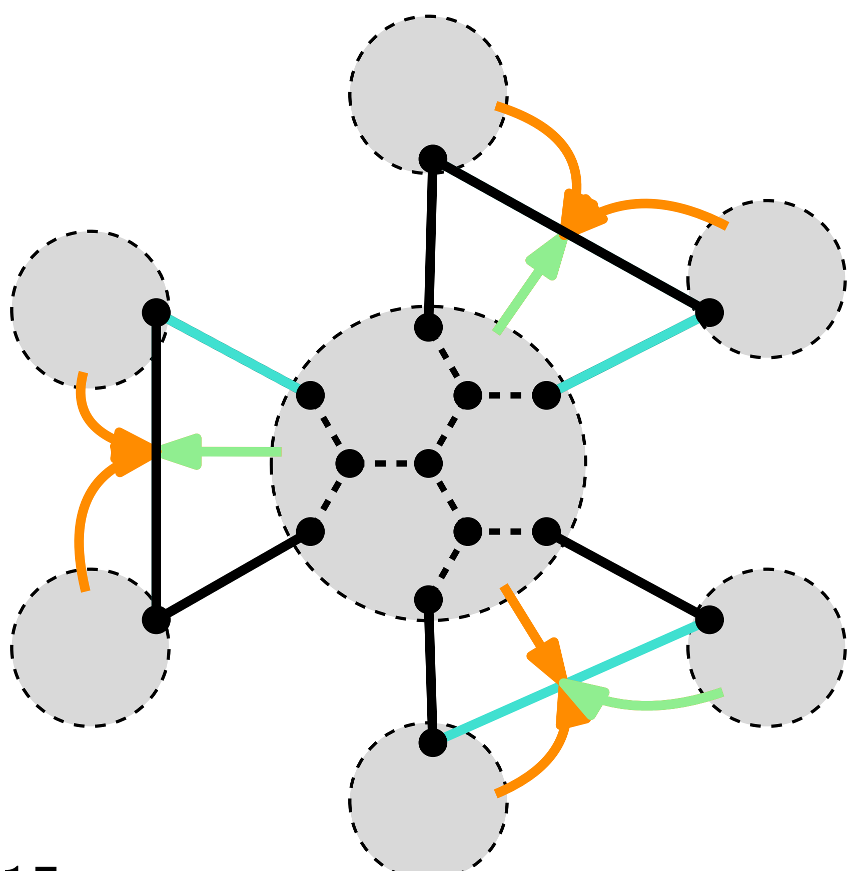
Result:
each cluster gets exclusive access to two edges in its 2-hop neighborhood

Phase 2: Assign two exclusive edges to each cluster

Goal: get each cluster exclusive access to change colors of two edges

Technique:

- compute a maximal matching of the colored edges
- assign two maximal edges to each cluster via HSO



Result:

each cluster gets exclusive access to two edges in its 2-hop neighborhood

Runtime:

$$T_{\mathrm{MM}}(n) + T_{\mathrm{HSO}}(n, 2\Delta^2, 2\Delta)$$
$$=$$
$$T_{\mathrm{MIS}}(\Delta \cdot n, 2\Delta - 2) + \mathcal{O}(\log_\Delta n)$$

Phase 3: Switch colors in order to complete the coloring

Goal:	extend the coloring inside the clusters

Phase 3: Switch colors in order to complete the coloring

Goal: extend the coloring inside the clusters

Technique:

- move assigned edges to the immediate neighborhood of the cluster
- change colors of assigned edges to satisfy:

Lemma 2: If $\exists k \in \mathbb{N} : |\varphi(N_E(V_k))| \geq \Delta$, then we can extend φ to T

Phase 3: Switch colors in order to complete the coloring

Goal: extend the coloring inside the clusters

Technique:
- move assigned edges to the immediate neighborhood of the cluster
- change colors of assigned edges to satisfy:

Lemma 2: If $\exists k \in \mathbb{N} : |\varphi(N_E(V_k))| \geq \Delta$, then we can extend φ to T

Result: each cluster can now independently extend the coloring

Phase 3: Switch colors in order to complete the coloring

Goal: extend the coloring inside the clusters

Technique:
- move assigned edges to the immediate neighborhood of the cluster
- change colors of assigned edges to satisfy:

Lemma 2: If $\exists k \in \mathbb{N} : |\varphi(N_E(V_k))| \geq \Delta$, then we can extend φ to T

Result: each cluster can now independently extend the coloring

Runtime: $\mathcal{O}(1)$

High level overview of our algorithm

Phase 1: Partition vertices into clusters	$T_{\mathrm{MIS}}(n)$

High level overview of our algorithm

Phase 1: Partition vertices into clusters	$T_{\mathrm{MIS}}(n)$
Phase 2: Assign two exclusive edges to each cluster	$T_{\mathrm{MM}}(n) + \mathcal{O}(\log_{\Delta} n)$

High level overview of our algorithm

Phase 1: Partition vertices into clusters	$T_{\mathrm{MIS}}(n)$
Phase 2: Assign two exclusive edges to each cluster	$T_{\mathrm{MM}}(n) + \mathcal{O}(\log_\Delta n)$
Phase 3: Switch colors and complete the coloring	$\mathcal{O}(1)$

High level overview of our algorithm

Phase 1: Partition vertices into clusters	$T_{\mathrm{MIS}}(n)$
Phase 2: Assign two exclusive edges to each cluster	$T_{\mathrm{MM}}(n) + \mathcal{O}(\log_{\Delta} n)$
Phase 3: Switch colors and complete the coloring	$\mathcal{O}(1)$

Reduction from $2\Delta - 2$-edge coloring to MIS	$T_{\mathrm{MIS}}(n) + \mathcal{O}(\log_{\Delta} n)$

Reduction to greedy edge coloring

Phase 1: Partition vertices into clusters $\mathcal{O}(\log \Delta) + T_{2\Delta-1}(n)$

- ~~compute MIS on power graph G^k~~
- compute a $\mathcal{O}(\log \Delta)$-ruling set on G^8

Reduction to greedy edge coloring

Phase 1: Partition vertices into clusters $\mathcal{O}(\log \Delta) + T_{2\Delta-1}(n)$

- ~~compute MIS on power graph G^k~~
- compute a $\mathcal{O}(\log \Delta)$-ruling set on G^8

Phase 2: Assign two exclusive edges to each cluster $\mathcal{O}(\log \Delta \cdot \log_\Delta n)$

- ~~compute maximal matching of intercluster edges~~
- compute 2-edge ruling set of intercluster edges

Reduction to greedy edge coloring

Phase 1: Partition vertices into clusters $\mathcal{O}(\log \Delta) + T_{2\Delta-1}(n)$

- ~~compute MIS on power graph G^k~~
- compute a $\mathcal{O}(\log \Delta)$-ruling set on G^8

Phase 2: Assign two exclusive edges to each cluster $\mathcal{O}(\log \Delta \cdot \log_\Delta n)$

- ~~compute maximal matching of intercluster edges~~
- compute 2-edge ruling set of intercluster edges

Phase 3: Switch colors and complete the coloring $\mathcal{O}(1)$

- nothing changes

Reduction to greedy edge coloring

Phase 1: Partition vertices into clusters	$\mathcal{O}(\log\Delta) + T_{2\Delta-1}(n)$

- ~~compute MIS on power graph G^k~~
- compute a $\mathcal{O}(\log\Delta)$-ruling set on G^8

Phase 2: Assign two exclusive edges to each cluster	$\mathcal{O}(\log\Delta \cdot \log_\Delta n)$

- ~~compute maximal matching of intercluster edges~~
- compute 2-edge ruling set of intercluster edges

Phase 3: Switch colors and complete the coloring	$\mathcal{O}(1)$

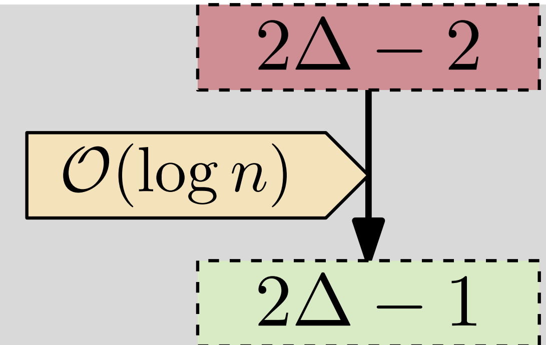
- nothing changes

Reduction from $2\Delta - 2$ to $2\Delta - 1$-edge coloring	$T_{2\Delta-1}(n) + \mathcal{O}(\log n)$

Wrapping up

Result

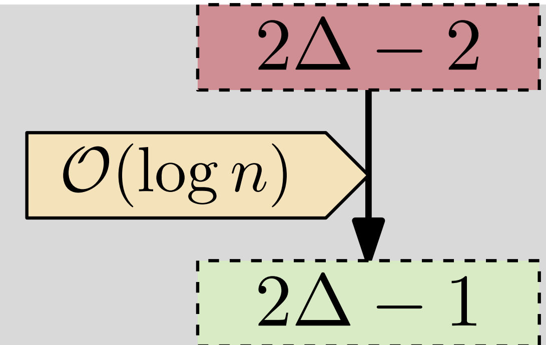
optimal reduction from greedy edge coloring to non-greedy edge coloring



Wrapping up

Result

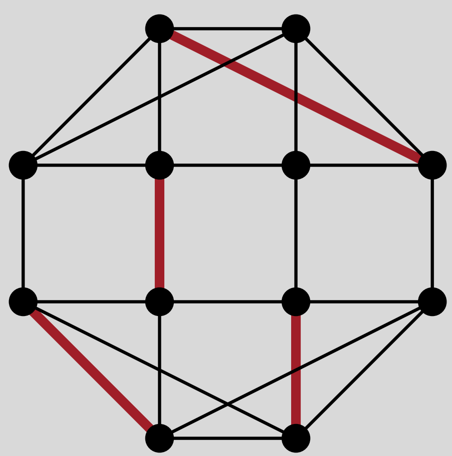
optimal reduction from greedy edge coloring to non-greedy edge coloring



Open problem

Given

4-regular graph $G = (V, E)$

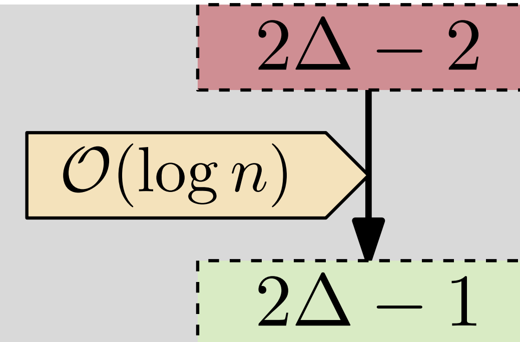


matching $M \subseteq E$

Wrapping up

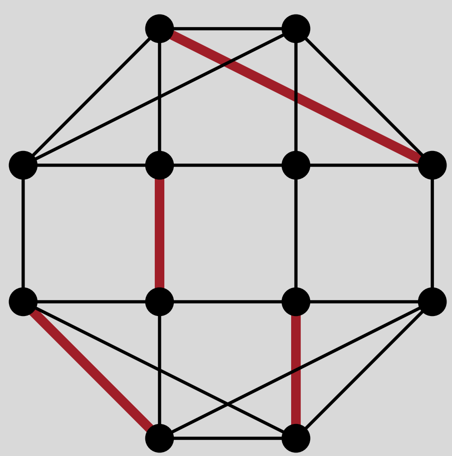
Result

optimal reduction from greedy edge coloring to non-greedy edge coloring

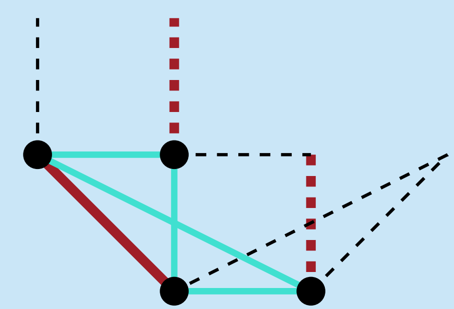


Open problem

Given	**Goal**
4-regular graph $G = (V, E)$	find small-diameter subgraph H



matching $M \subseteq E$



such that H can still be properly 5-colored at the end