Improved Deterministic Distributed Matching via Rounding

Manuela Fischer
ETH Zurich

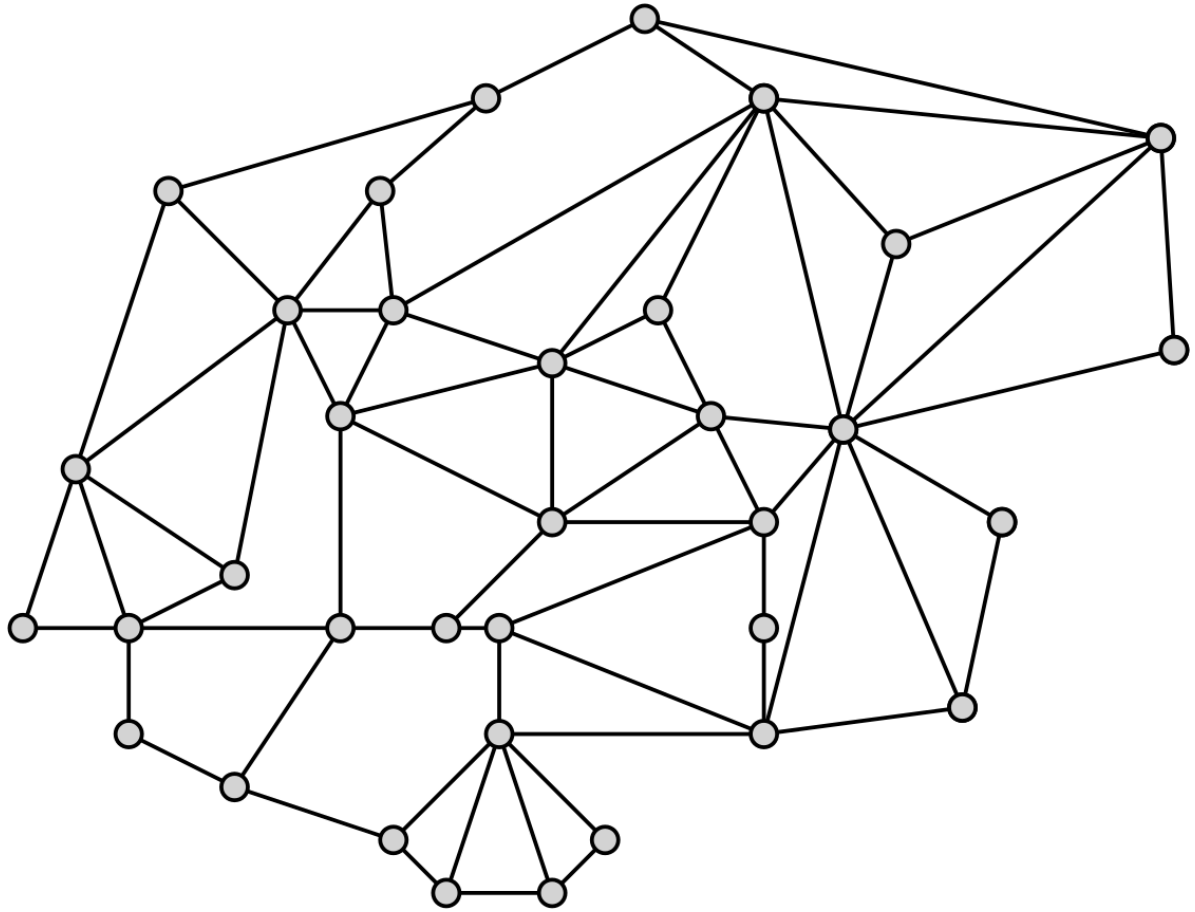
LOCAL Model *Linial* [FOCS'87]

- undirected graph $G = (V, E)$,
  n nodes, maximum degree Δ

- synchronous message-passing rounds
- unbounded message size

- unbounded computation

- **Round Complexity**:
  number of rounds to solve the problem



round complexity of a problem characterizes its locality

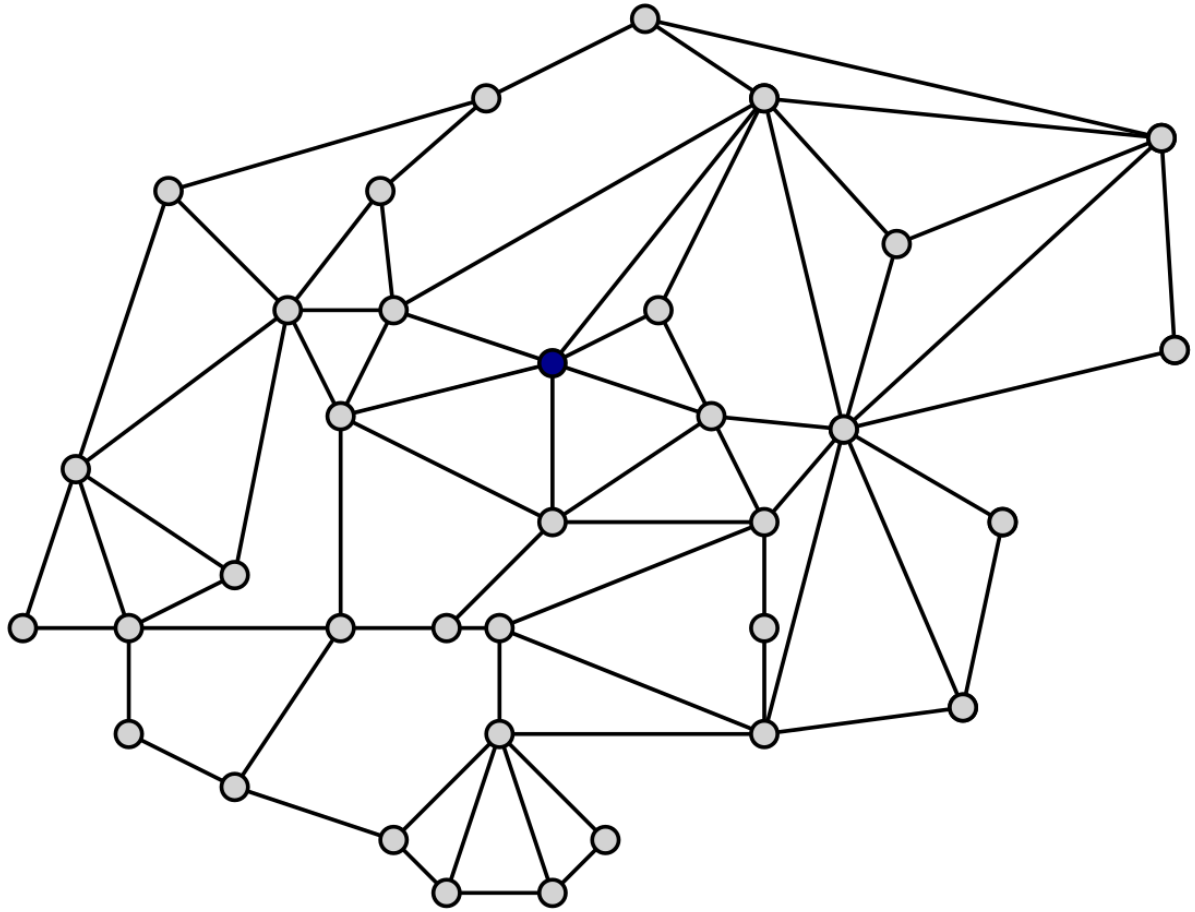
LOCAL Model *Linial* [FOCS'87]

- undirected graph $G = (V, E)$,
  n nodes, maximum degree Δ

- synchronous message-passing rounds
- unbounded message size

- unbounded computation

- **Round Complexity**:
  number of rounds to solve the problem



round complexity of a problem characterizes its locality

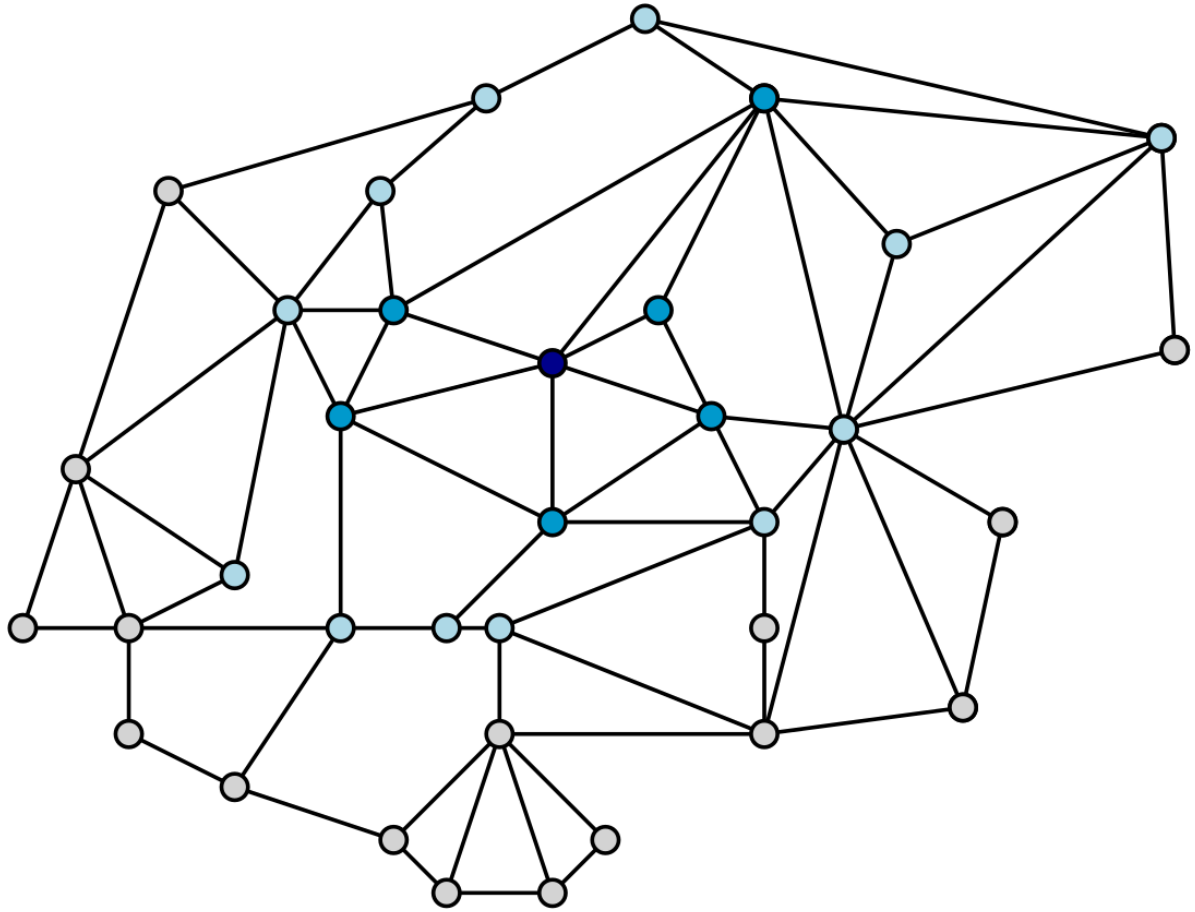
LOCAL Model *Linial* [FOCS'87]

- undirected graph $G = (V, E)$,
  n nodes, maximum degree Δ

- synchronous message-passing rounds
- unbounded message size

- unbounded computation

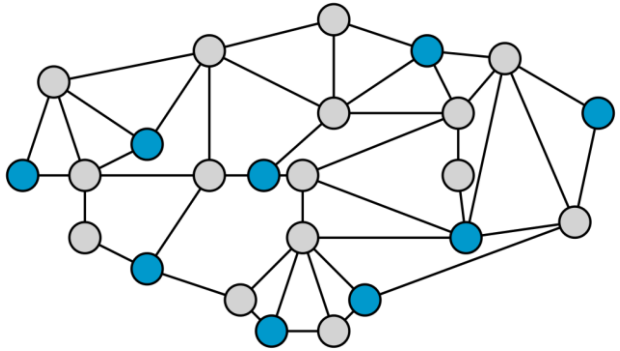
- **Round Complexity**:
  number of rounds to solve the problem



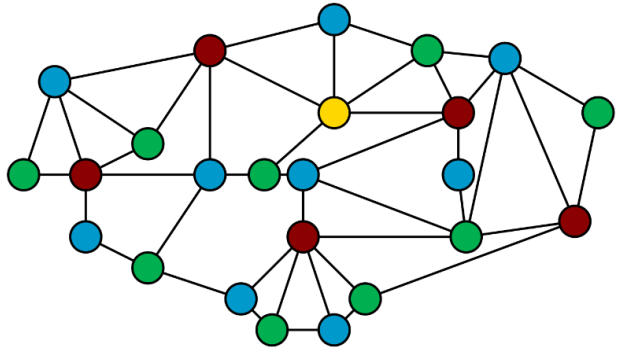
round complexity of a problem characterizes its locality

Classic LOCAL Graph Problems

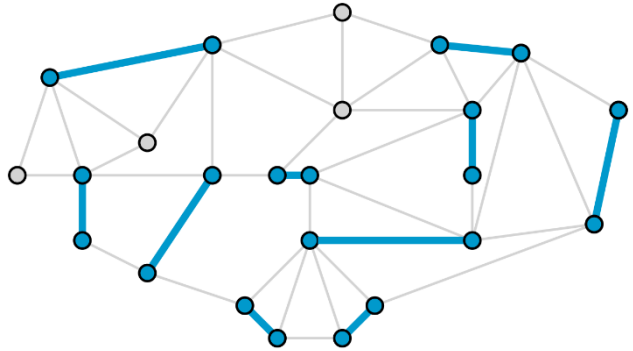
Maximal Independent Set



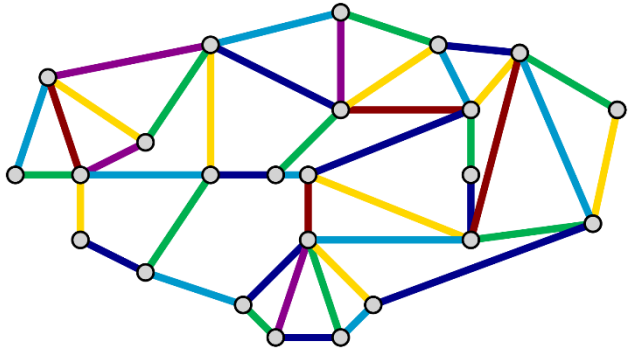
$(\Delta + 1)$-Vertex-Coloring



Maximal Matching



$(2\Delta - 1)$-Edge-Coloring



<u>Question from the 1980s</u> by Linial:

Are there efficient ($\text{poly}\log n$ rounds) **deterministic** algorithms for these problems?

Ghaffari, Kuhn, Maus [STOC'17]

Completeness of Deterministic Rounding

Ghaffari, Kuhn, Maus [STOC'17]

Completeness of Deterministic Rounding

Efficient **deterministic rounding** is **the only obstacle** for efficient deterministic LOCAL graph algorithms.

Rounding:
Turning fractional values into integral ones
while approximately preserving some linear constraints.

Our Contribution:

Deterministic Rounding for Matching

Our Contribution:

Deterministic Rounding for Matching

Matching admits an **efficient deterministic algorithm** because matching admits **efficient deterministic rounding**.

- first deterministic LOCAL rounding method
- not only improving over state of the art of matching algorithms, but (hopefully) step in direction of solving all classic LOCAL problems
- extension led to first efficient $(2\Delta - 1)$-edge-coloring algorithm *F., Ghaffari, Kuhn* [FOCS'17]

Our Results

- **Maximal Matching** $O(\log^2 \Delta \cdot \log n)$
- **Randomized** Maximal Matching $O(\log^3 \log\ n + \log \Delta)$

- $(\mathbf{2+\varepsilon})$ **- Approximate** Maximum Matching $O\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon} + \log^* n\right)$
- $(2+\varepsilon)$ - Approximate Maximum **Weighted Matching** $O\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon} + \log^* n\right)$

- $(2+\varepsilon)$ - Approximate Maximum (Weighted) **B-Matching** $O\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon} + \log^* n\right)$

- $\boldsymbol{\varepsilon}$ **- Maximal Matching** $O\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon}\right)$
- $(2+\varepsilon)$ - Approximate Minimum **Edge Dominating Set** $O\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon}\right)$

Our Results

$O(\log^4 n)$
Hańćkowiak, Karoński, Panconesi [SODA'98, PODC'99]

$O(\Delta + \log^* n)$
Panconesi, Rizzi [DIST'01]

- **Maximal Matching** $\mathrm{O}(\log^2 \Delta \cdot \log \mathrm{n})$
- **Randomized** Maximal Matching $\mathrm{O}(\log^3 \log\ \mathrm{n} + \log \Delta)$

- $(\mathbf{2} + \boldsymbol{\varepsilon})$ **- Approximate** Maximum Matching $\mathrm{O}\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon} + \log^* \mathrm{n}\right)$
- $(2 + \varepsilon)$ - Approximate Maximum **Weighted Matching** $\mathrm{O}\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon} + \log^* \mathrm{n}\right)$

- $(2 + \varepsilon)$ - Approximate Maximum (Weighted) **B-Matching** $\mathrm{O}\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon} + \log^* \mathrm{n}\right)$

- $\boldsymbol{\varepsilon}$ **- Maximal Matching** $\mathrm{O}\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon}\right)$
- $(2 + \varepsilon)$ - Approximate Minimum **Edge Dominating Set** $\mathrm{O}\left(\log^2 \Delta \cdot \log \frac{1}{\varepsilon}\right)$

Constant - Approximate Bipartite Matching $O(\log^2 \Delta)$ rounds

Constant - Approximate Bipartite Matching $O(\log^2 \Delta)$ rounds

I) 4 - Approximate Fractional Matching

$O(\log \Delta)$ rounds

Constant - Approximate Bipartite Matching $O(\log^2 \Delta)$ rounds

I) 4 - Approximate Fractional Matching

$O(\log \Delta)$ rounds

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

I) 4-Approximate Fractional Matching $O(\log \Delta)$ rounds

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds

Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds

Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds

Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

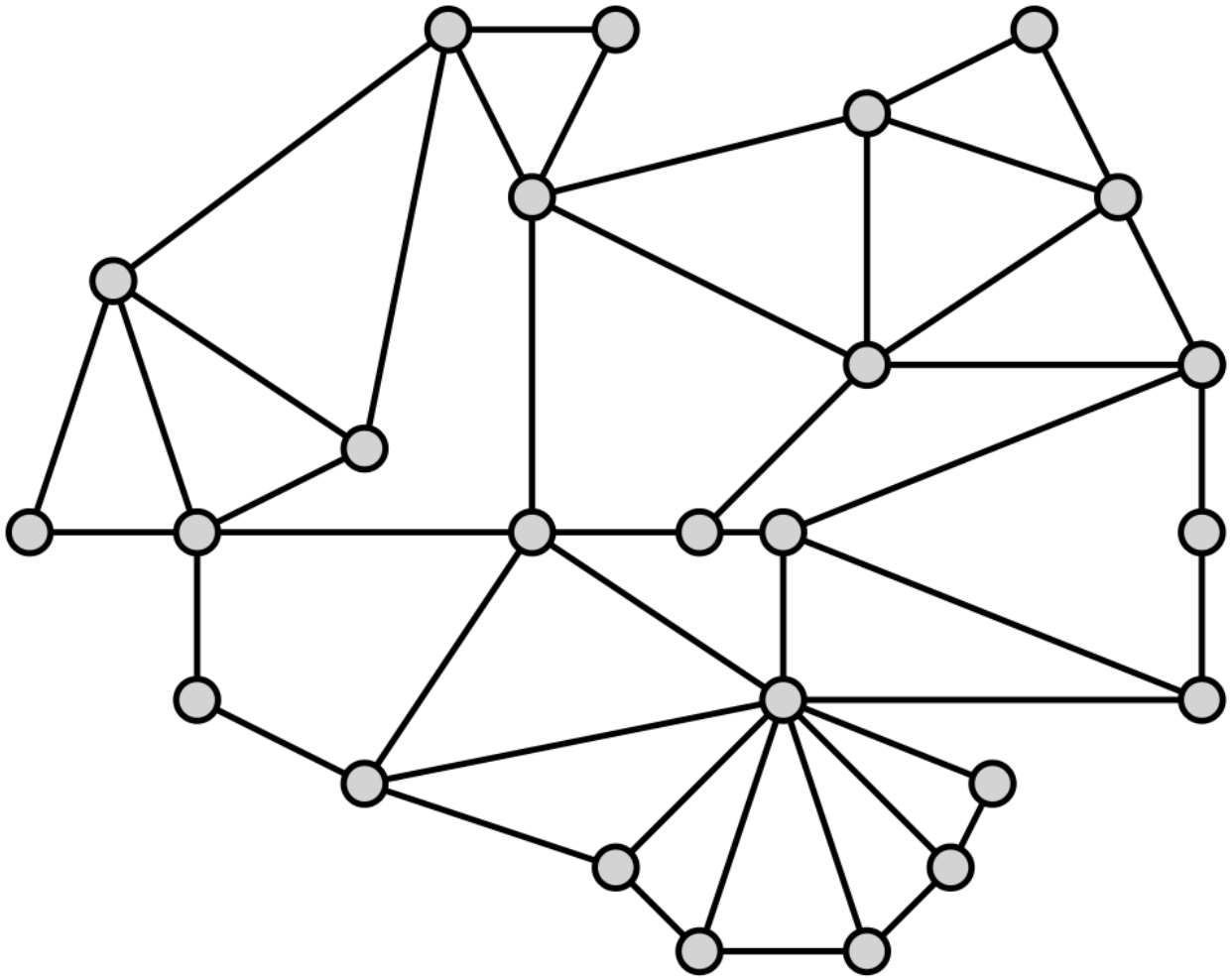
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

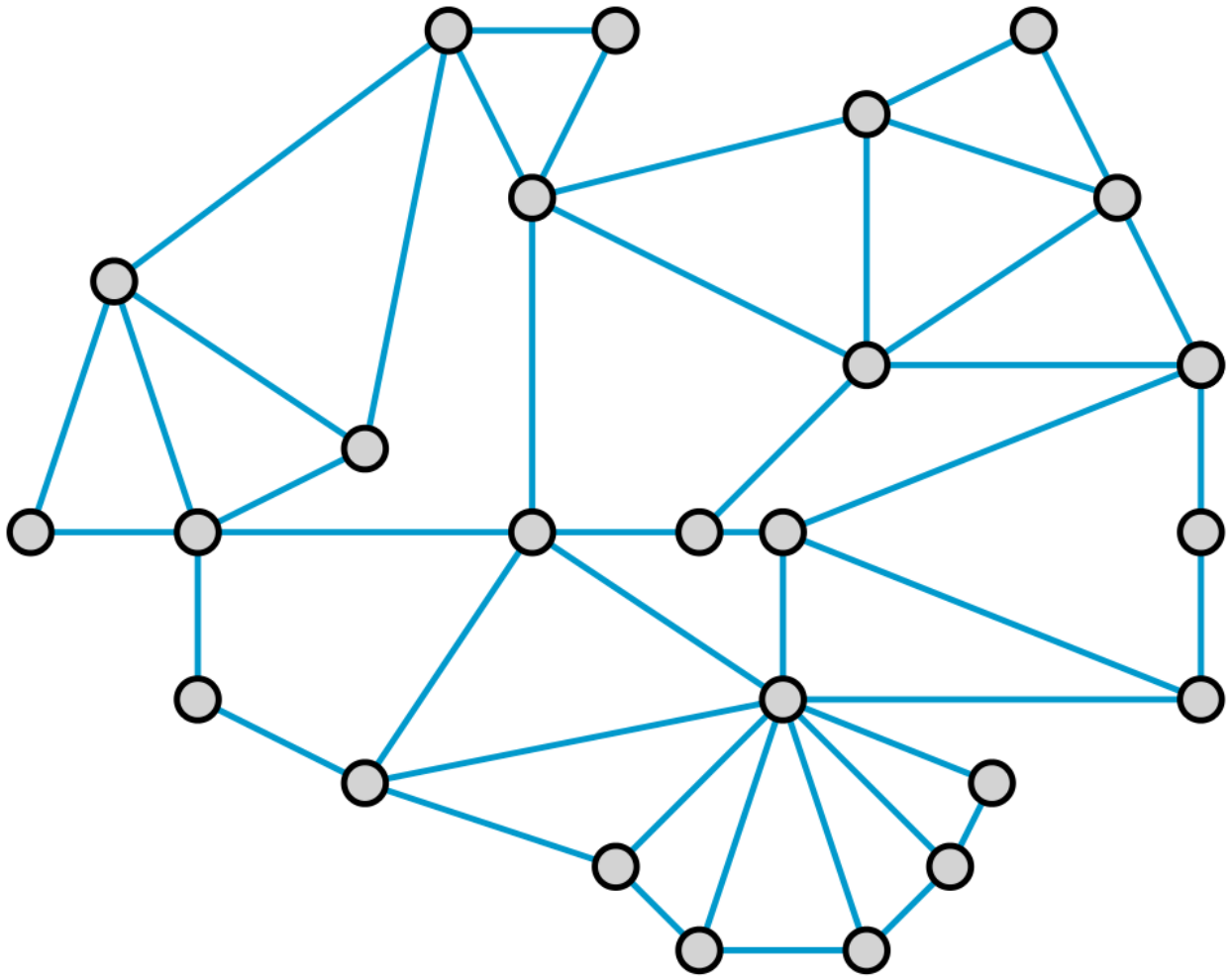
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



$\frac{1}{16}$

Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

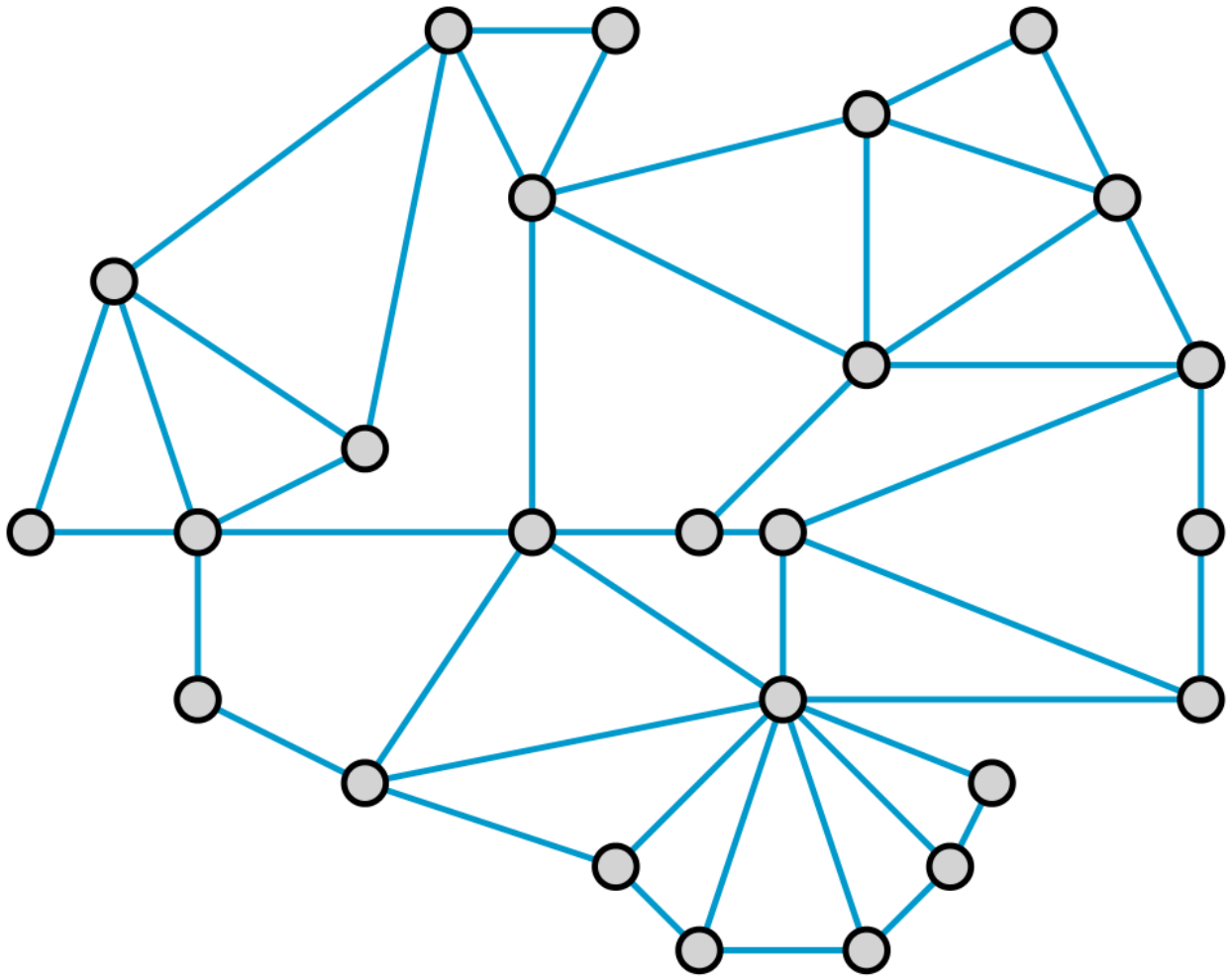
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



$\frac{1}{16}$

Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

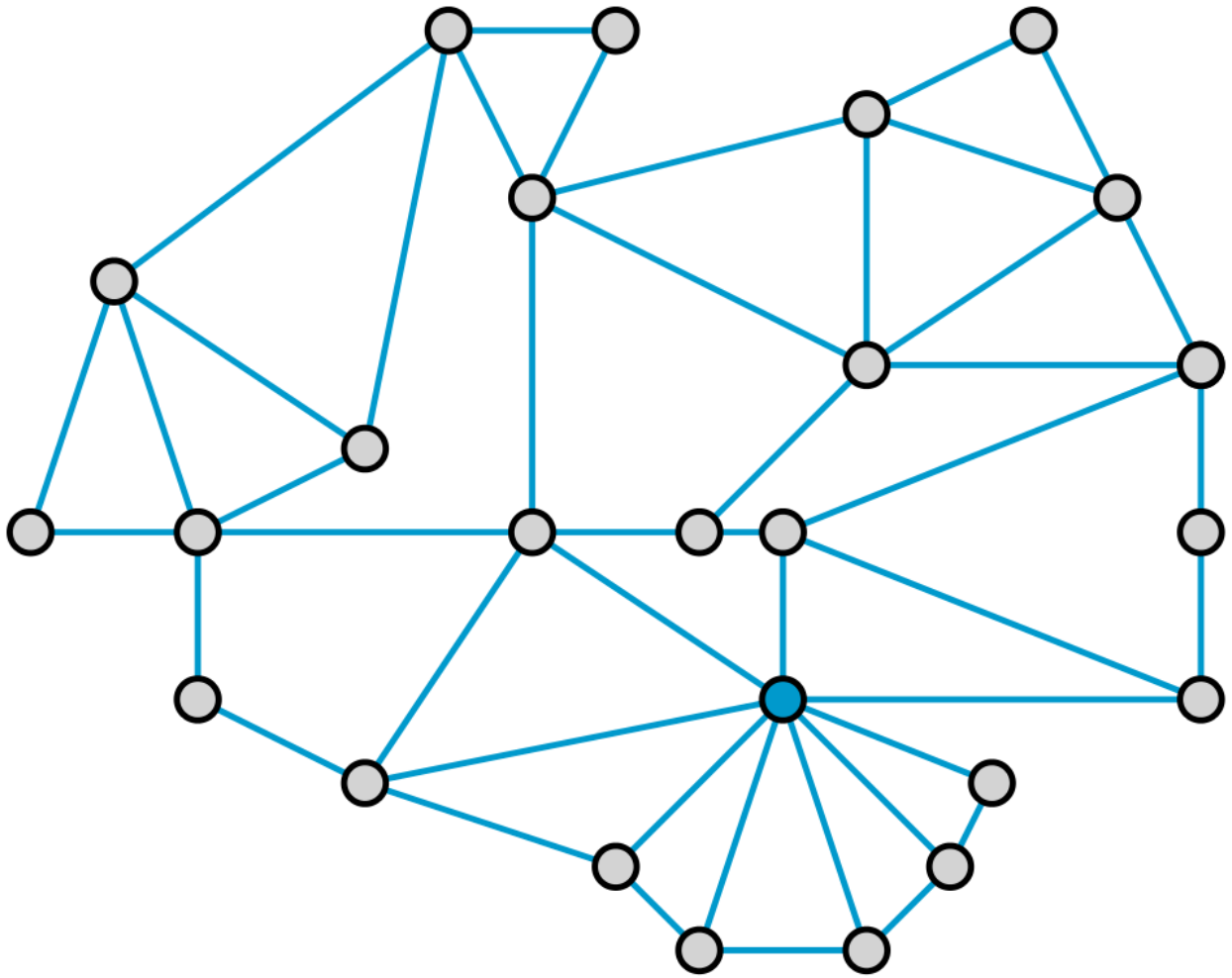
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



$\frac{1}{16}$

Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

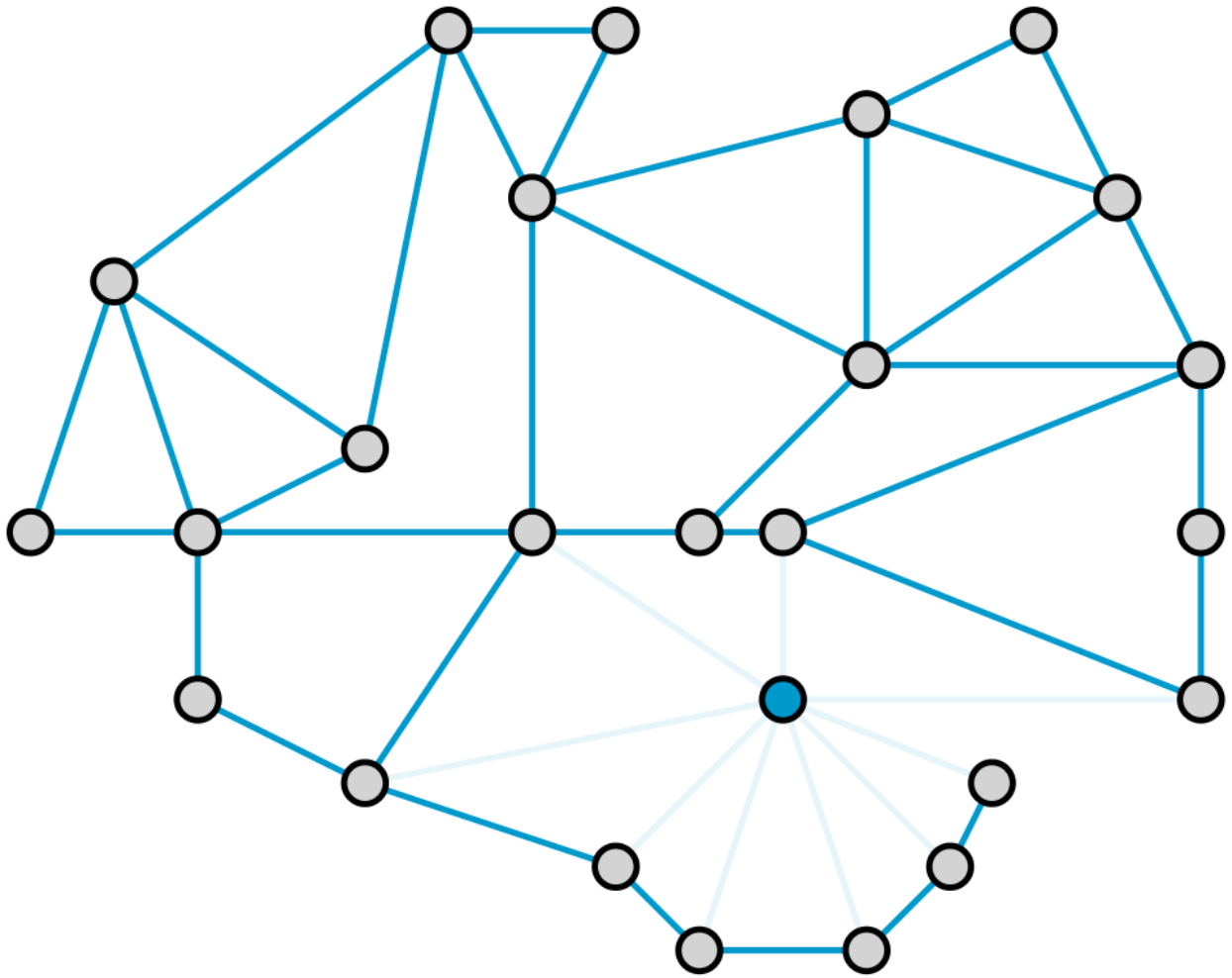
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



$\frac{1}{16}$

Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

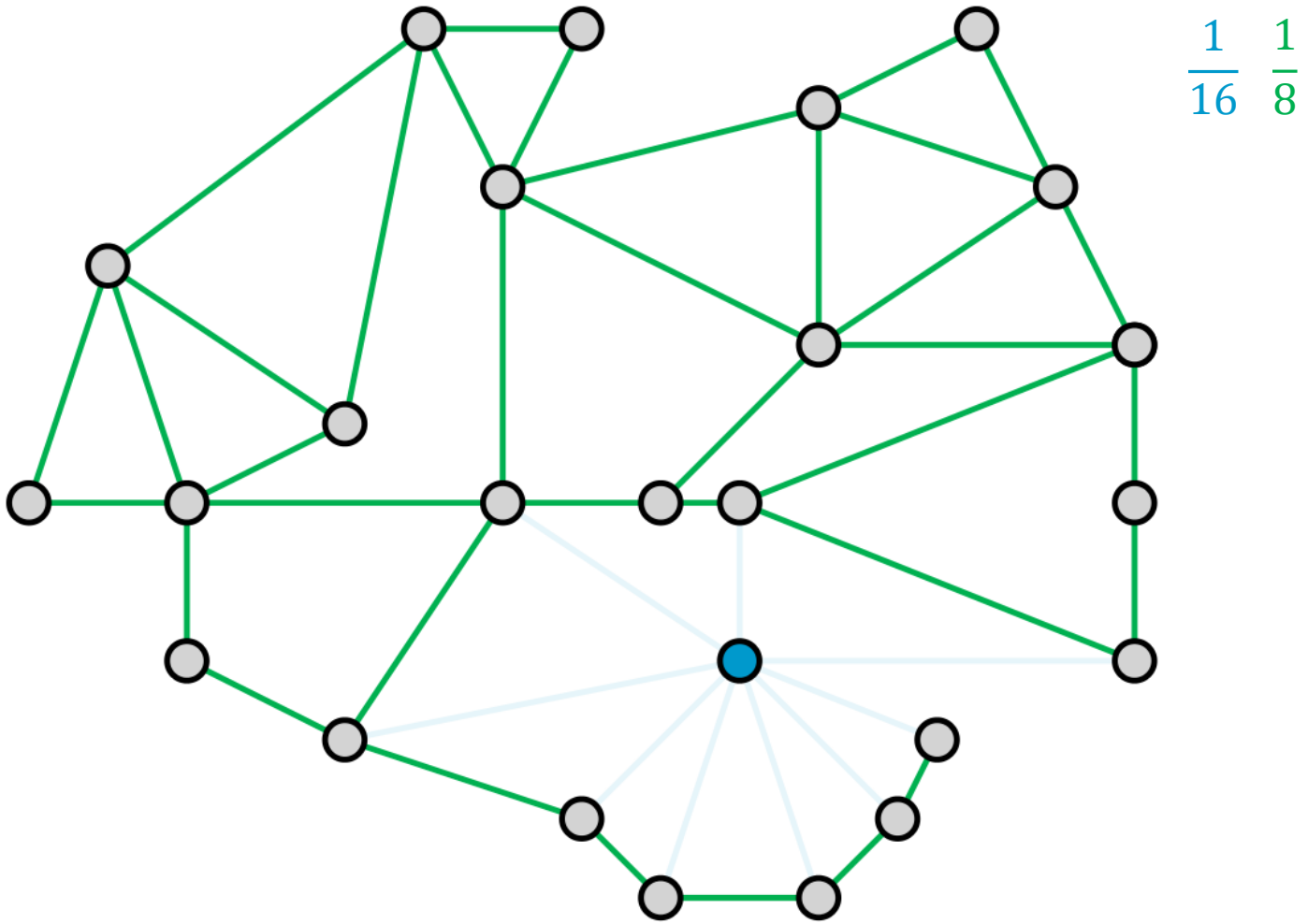
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



$\frac{1}{16}$ $\frac{1}{8}$

Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

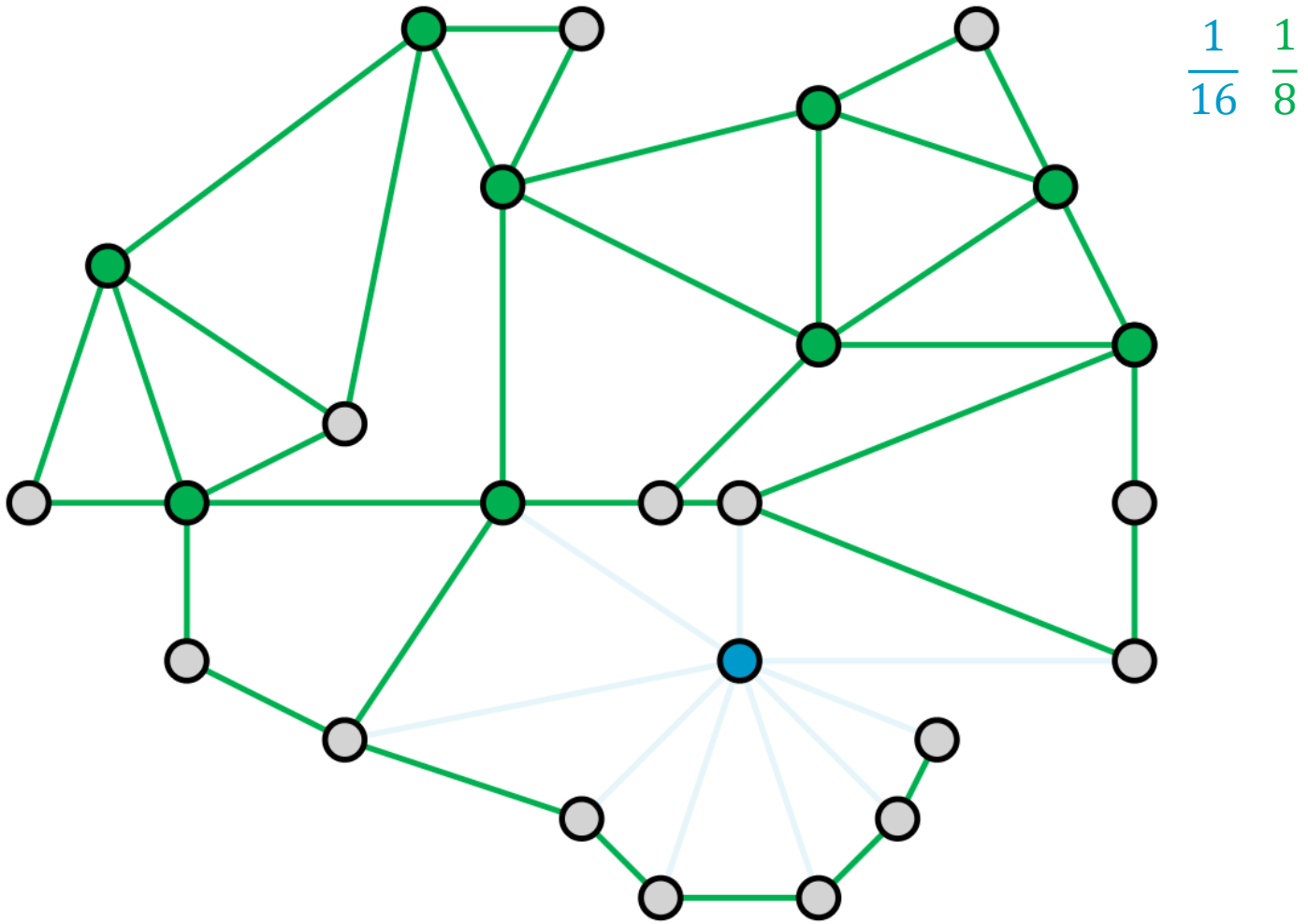
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



$\frac{1}{16}$ $\frac{1}{8}$

Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

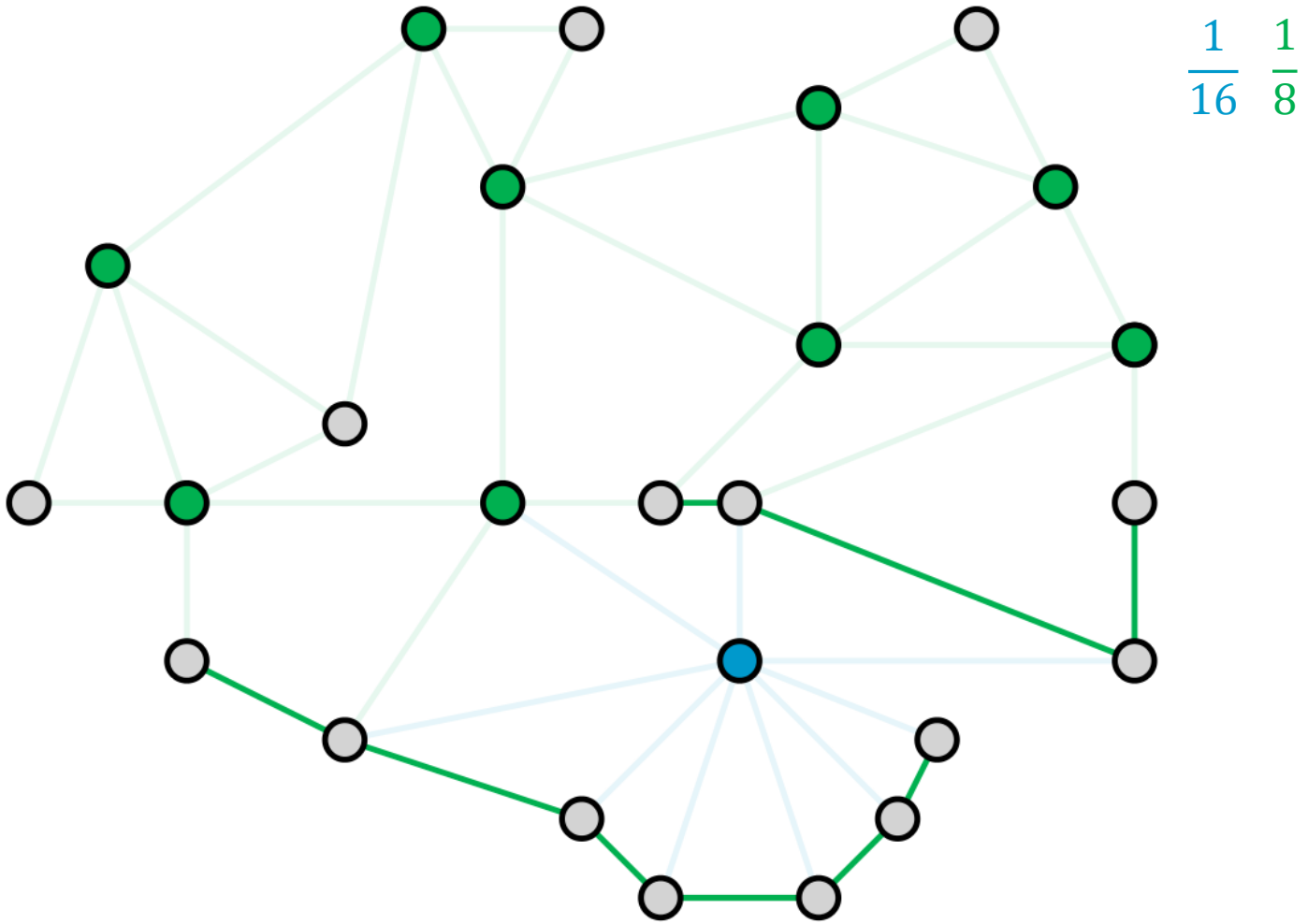
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$ value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

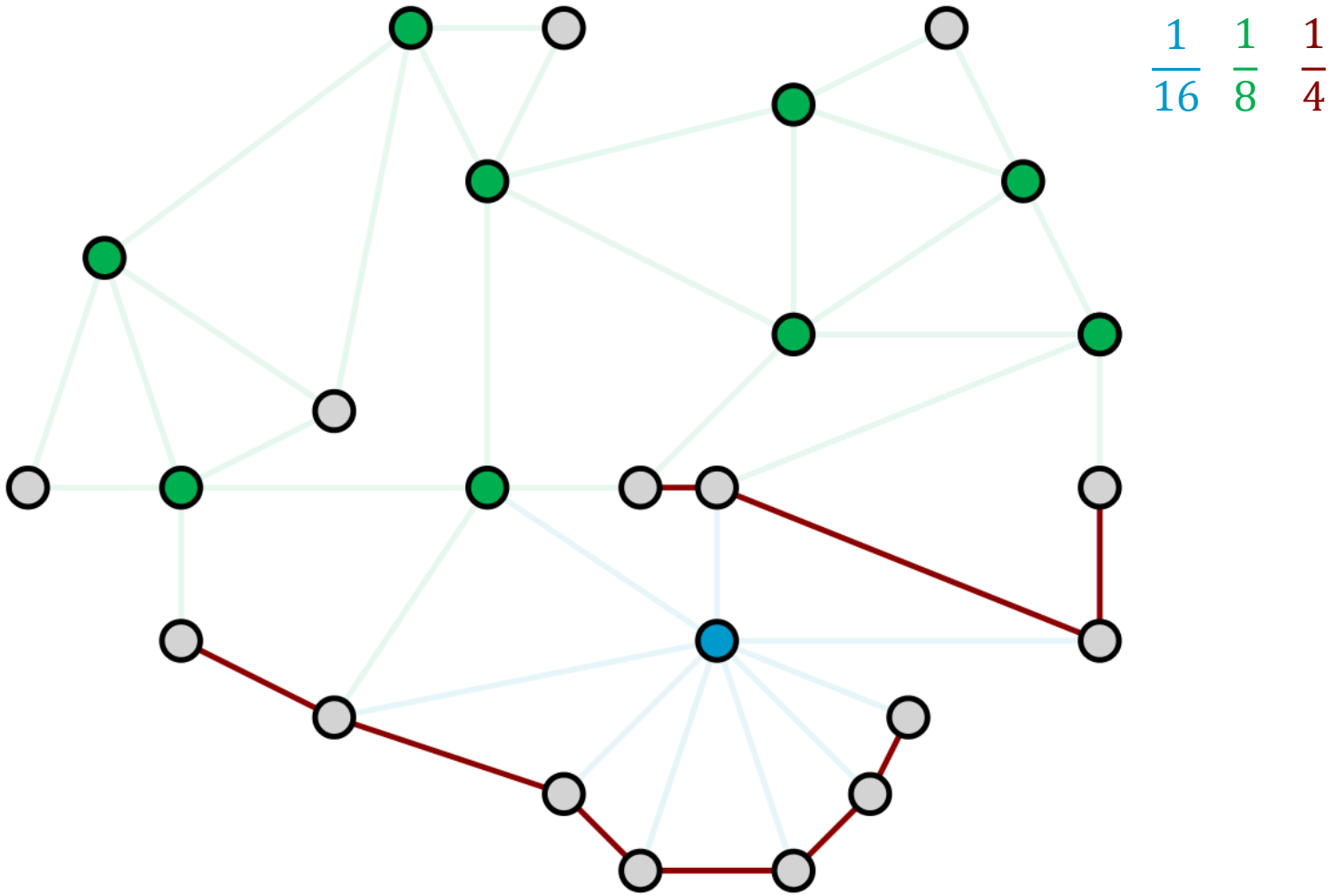
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

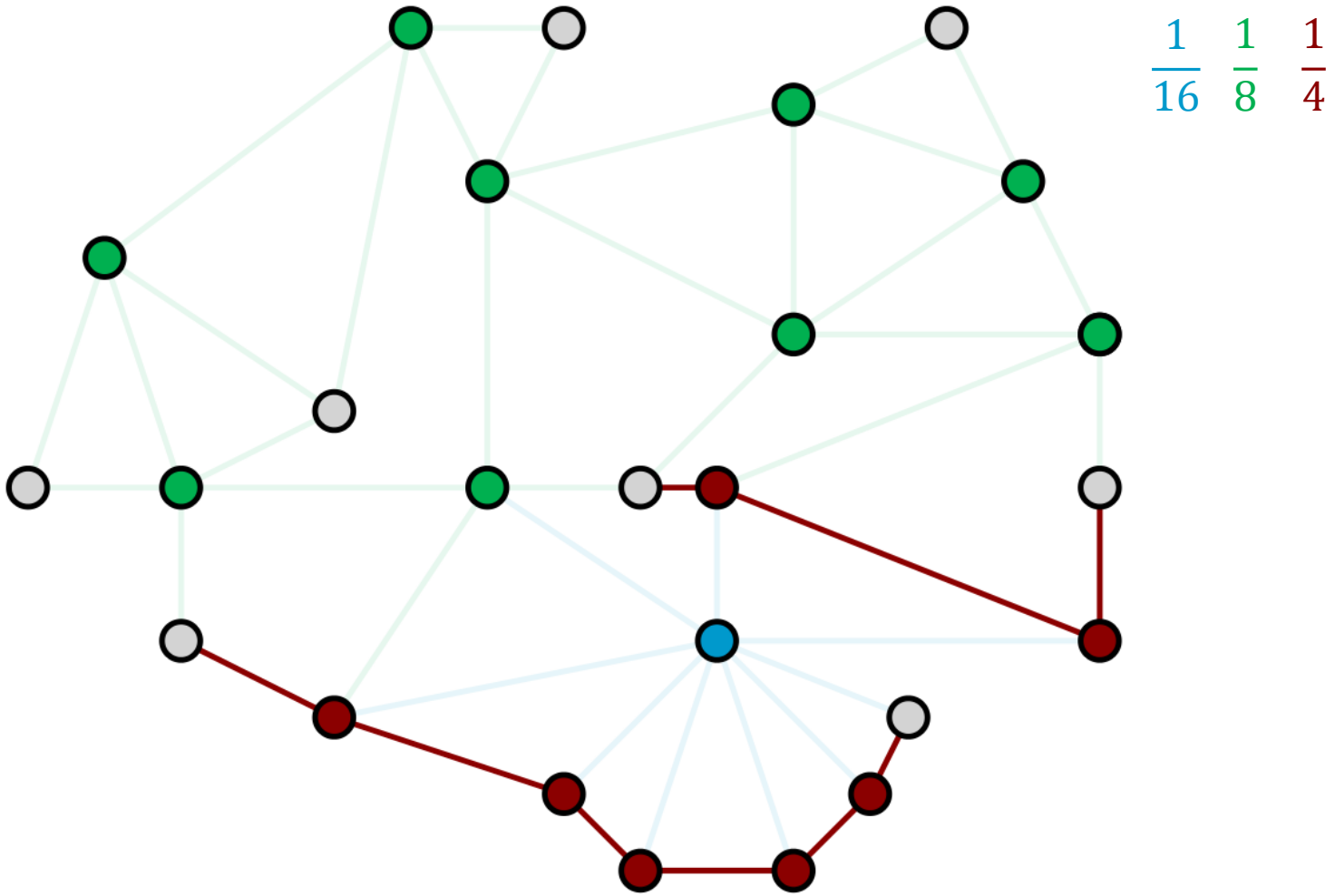
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

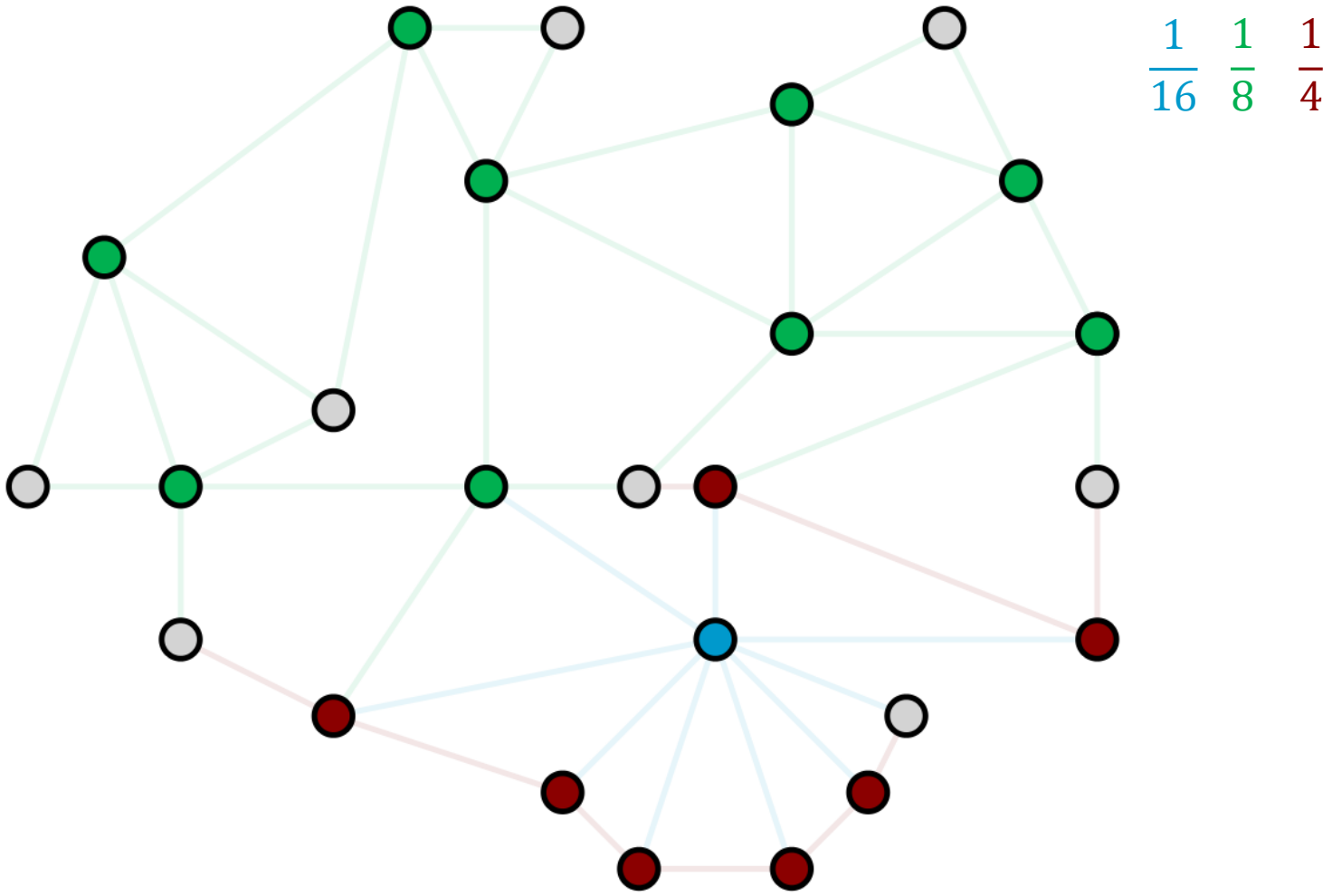
mark half-tight nodes

block its edges

double value of unblocked edges

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds



Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL Greedy Algorithm

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

mark half-tight nodes

block its edges

double value of unblocked edges

Constant - Approximate Bipartite Matching $O(\log^2 \Delta)$ rounds

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds

II) Rounding Fractional Bipartite Matching

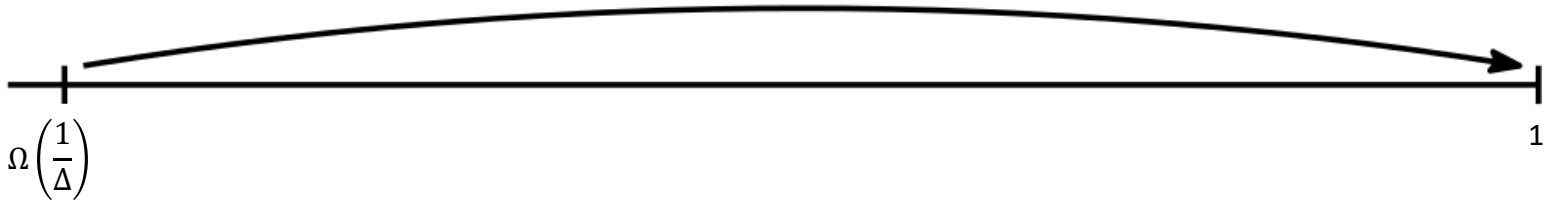
$O(\log^2 \Delta)$ rounds, $O(1)$ loss

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

II) Rounding Fractional Bipartite Matching

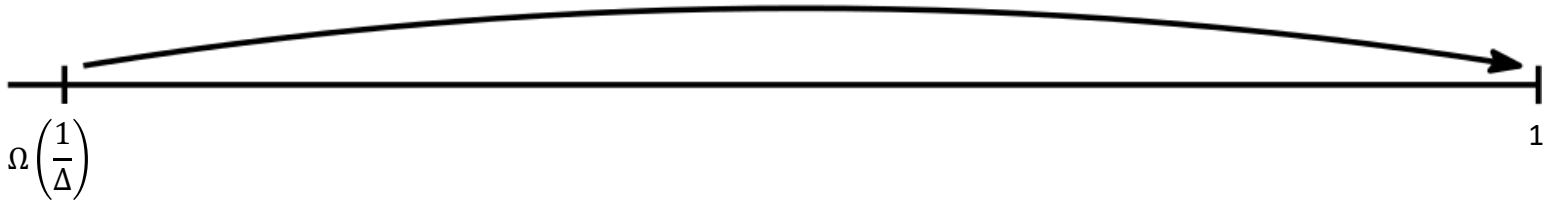
$O(\log^2 \Delta)$ rounds, $O(1)$ loss



II) Rounding Fractional Bipartite Matching

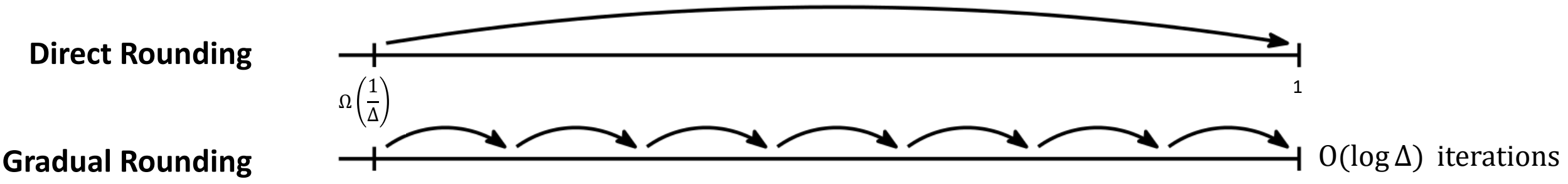
$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Direct Rounding



II) Rounding Fractional Bipartite Matching

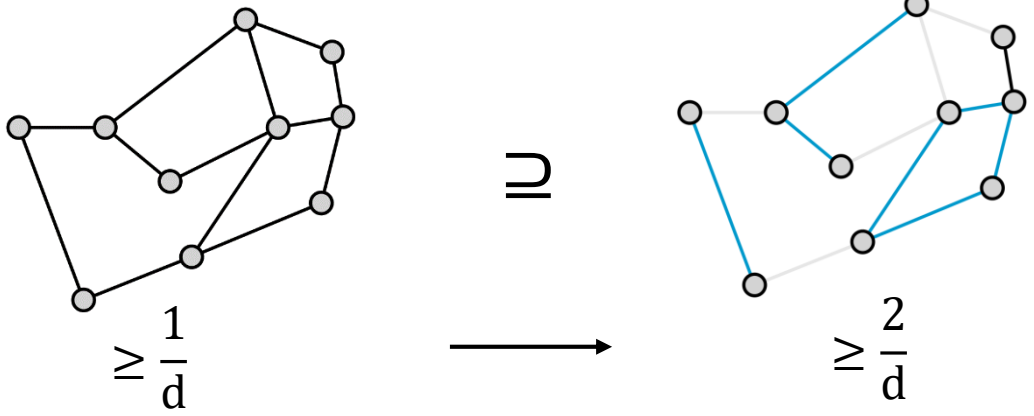
$O(\log^2 \Delta)$ rounds, $O(1)$ loss

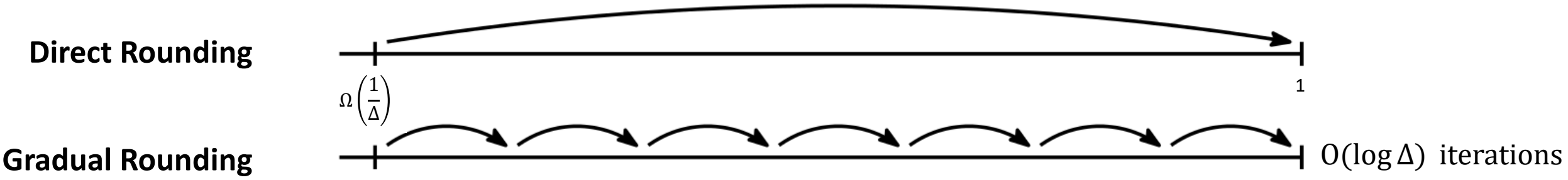


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Factor-2-Rounding

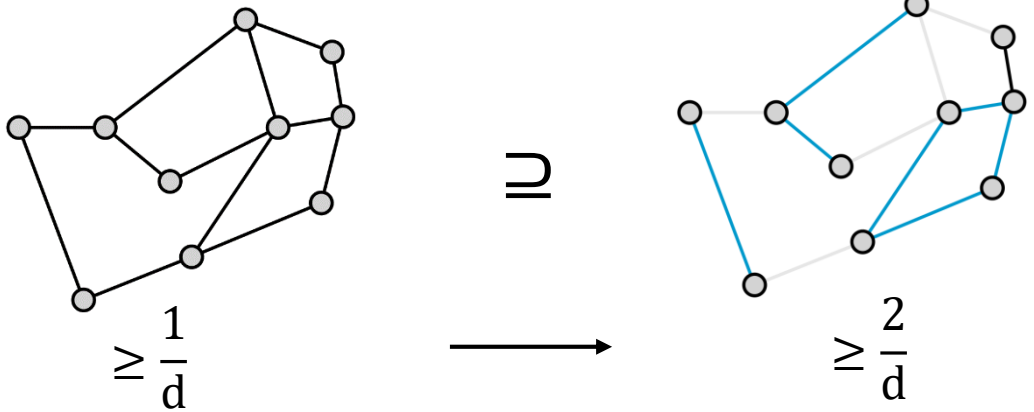




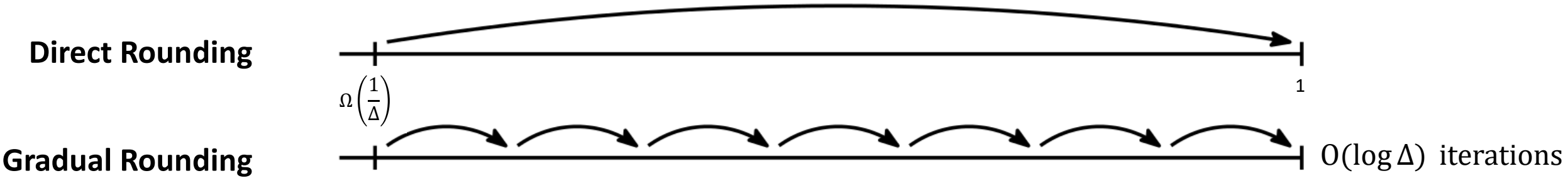
II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Factor-2-Rounding



using Locally Balanced Splitting,
inspired by
Hańćkowiak, Karoński, Panconesi [SODA'98,PODC'99]



II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

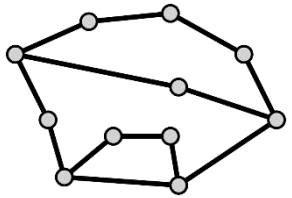
Iterated Factor-2-Rounding using Locally Balanced Splitting

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced

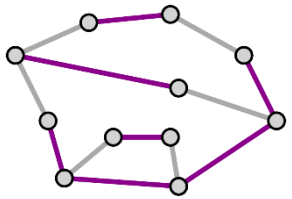


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced

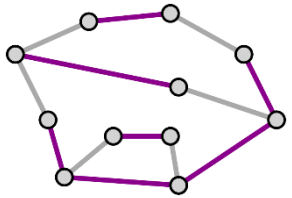


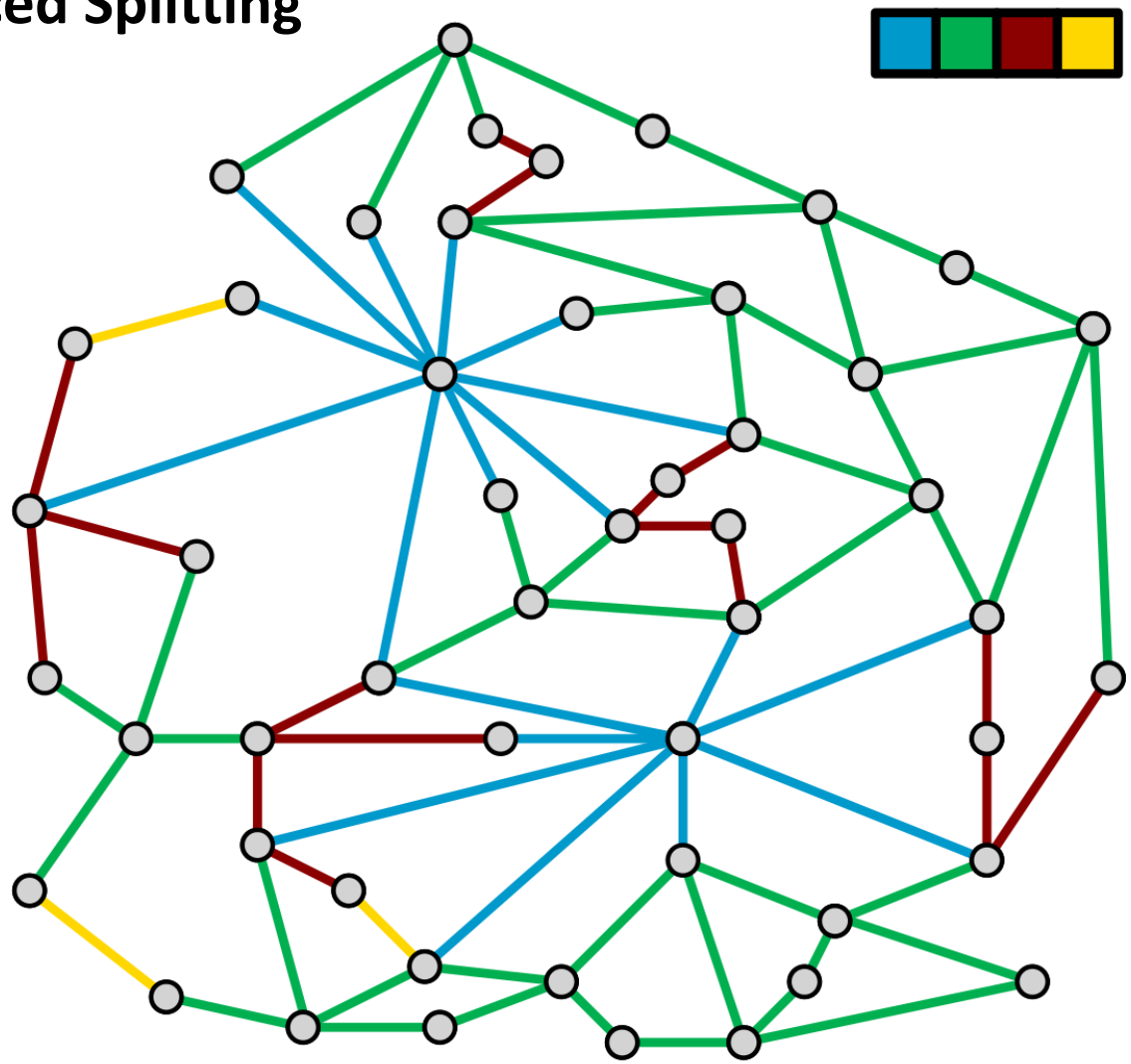
II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



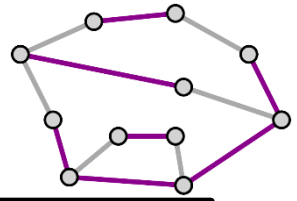


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

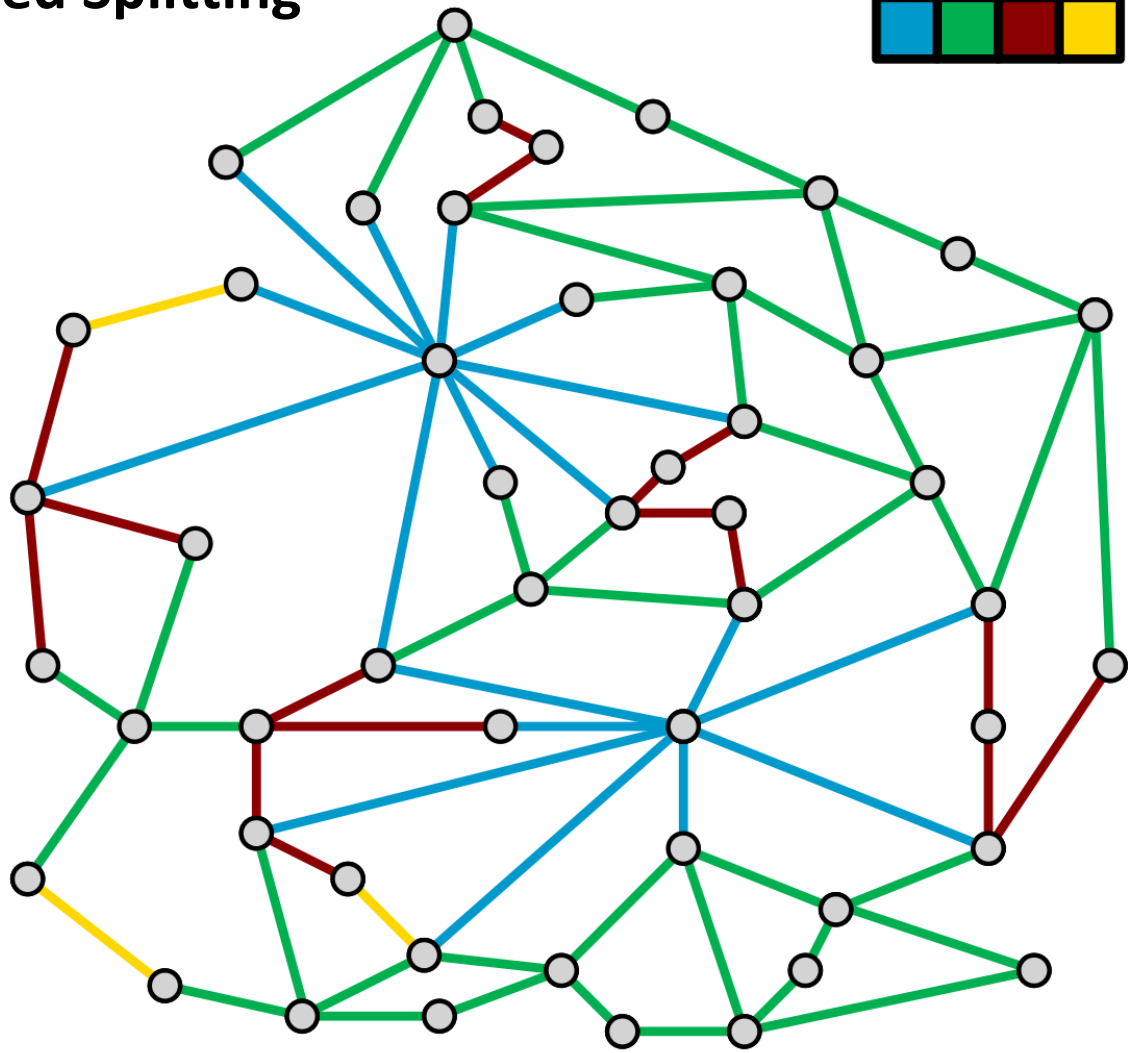
for $i = \lceil \log \Delta \rceil, \dots, 1$

$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

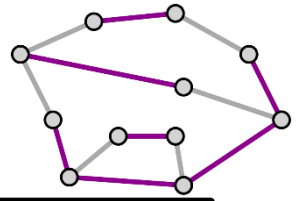


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

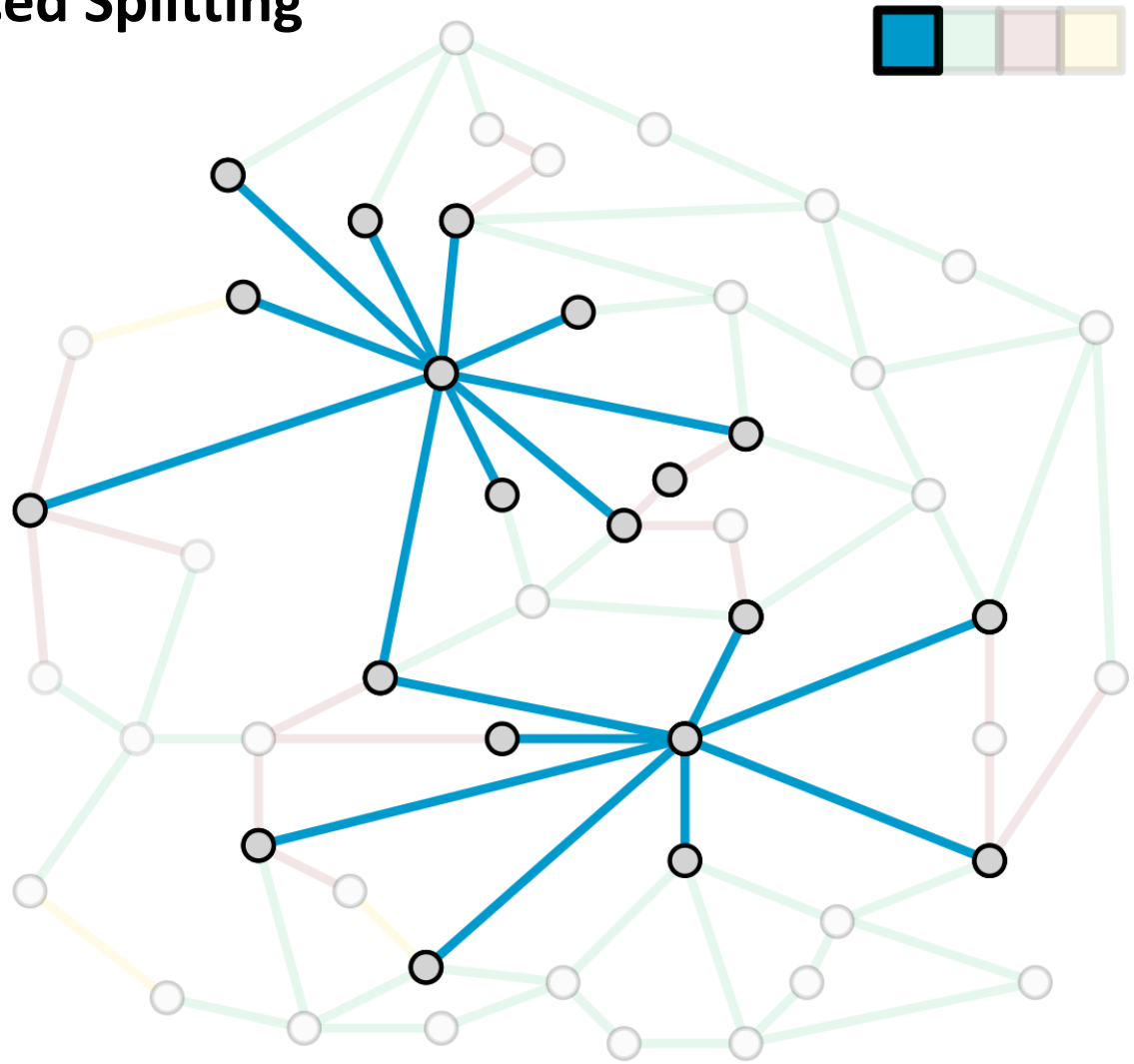
for $i = \lceil \log \Delta \rceil, \ldots, 1$

$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

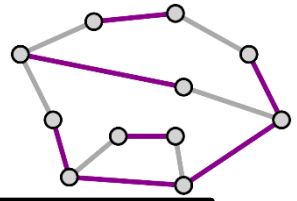


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

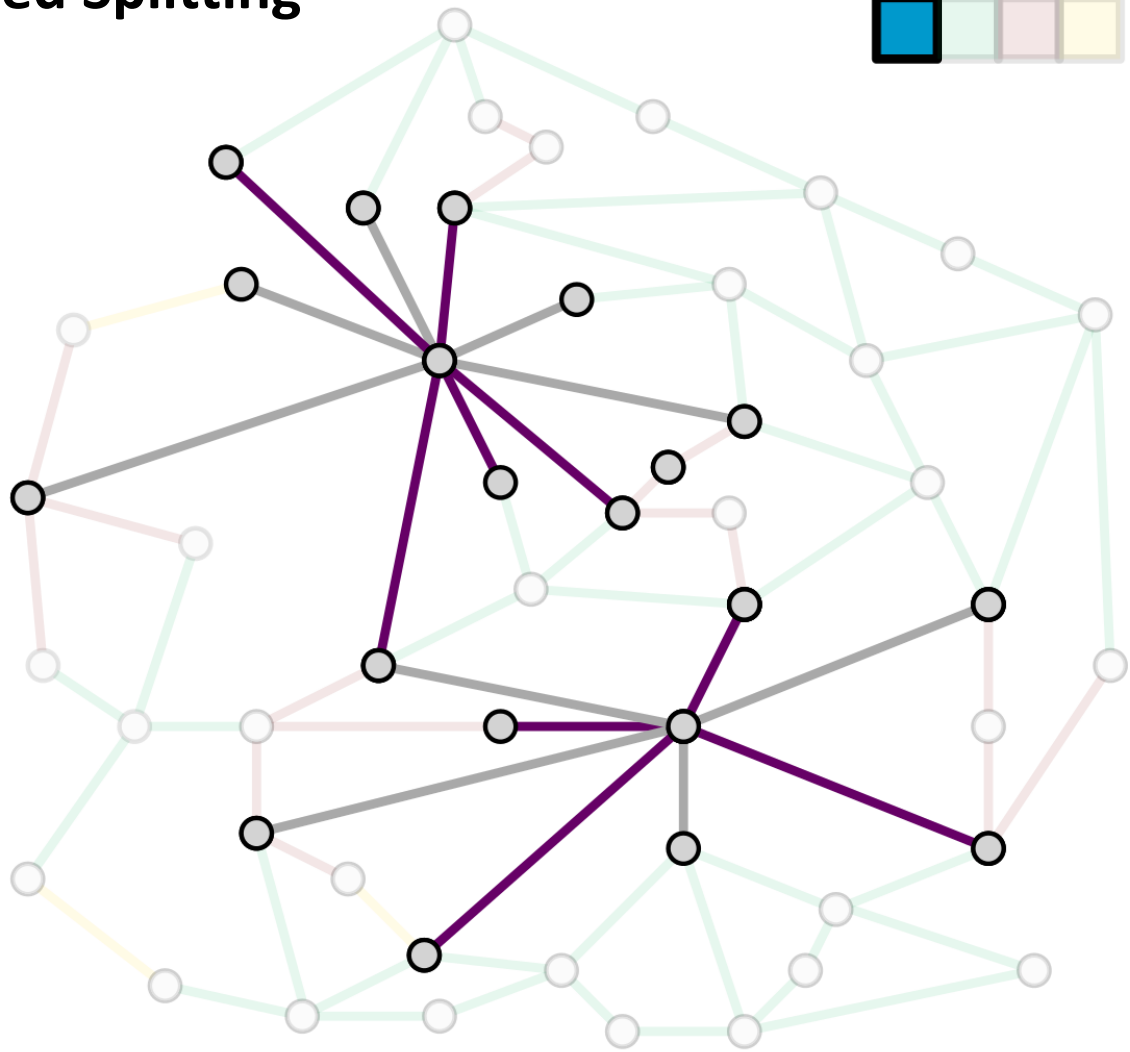
for $i = \lceil \log \Delta \rceil, \ldots, 1$

$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

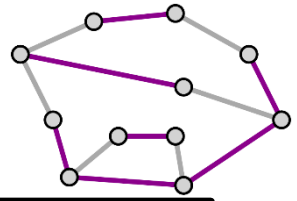


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

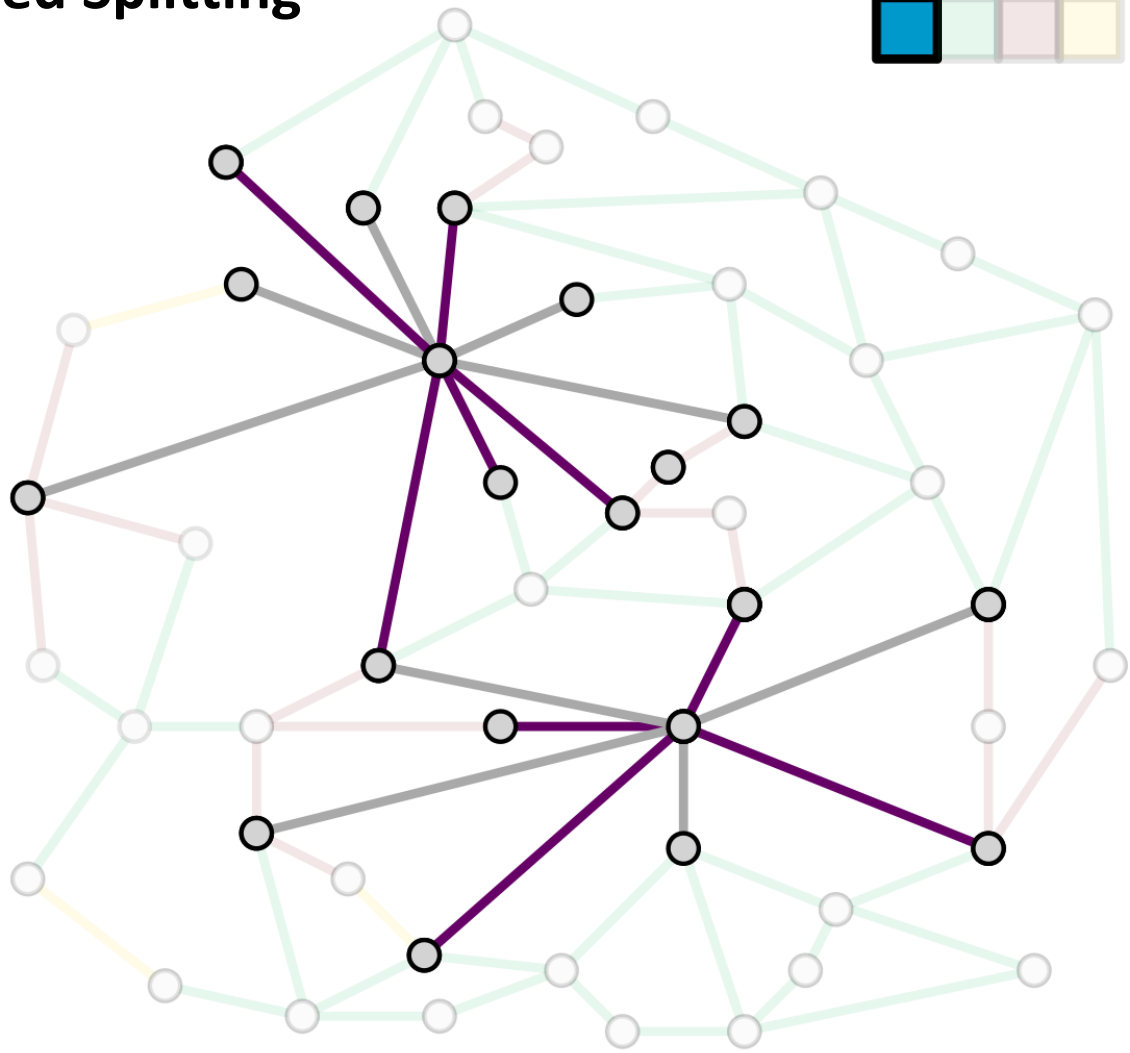
for $i = \lceil \log \Delta \rceil, \ldots, 1$

$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0



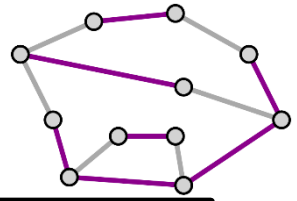
In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

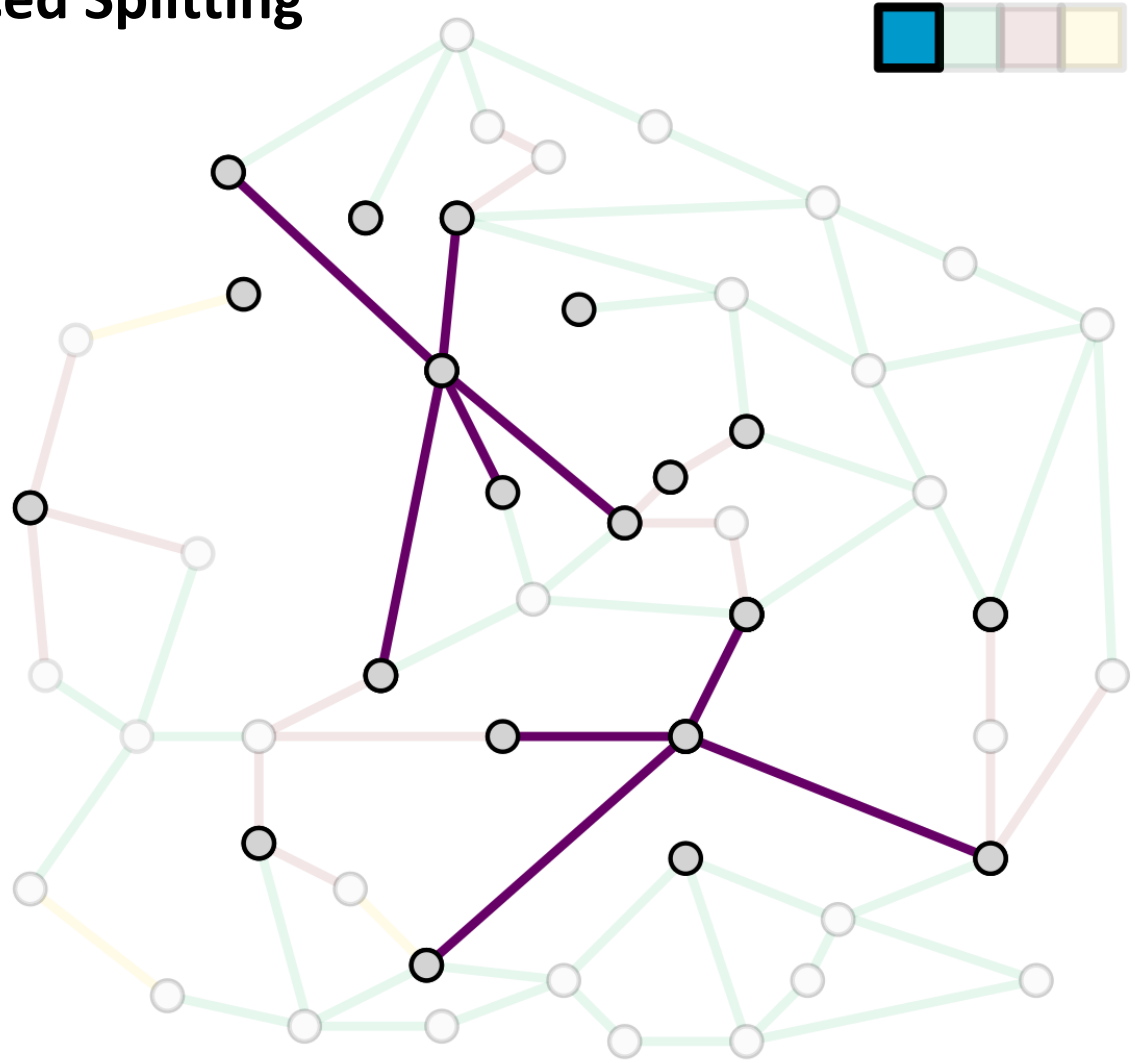
for $i = \lceil \log \Delta \rceil, \ldots, 1$

$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0



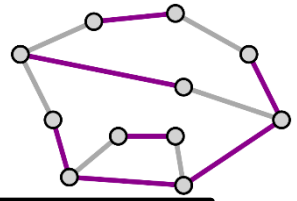
In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

for $i = \lceil \log \Delta \rceil, \ldots, 1$

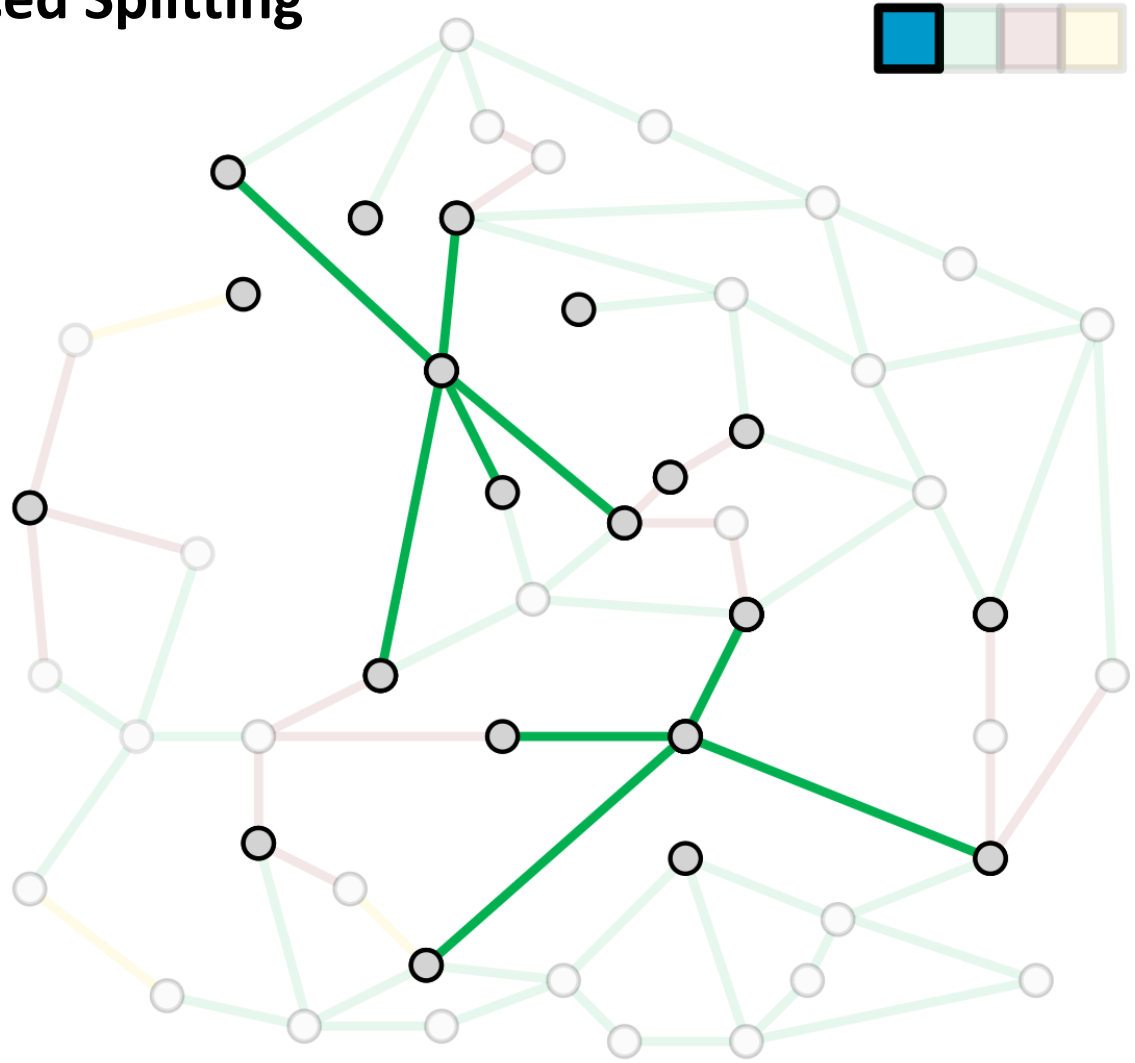
$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

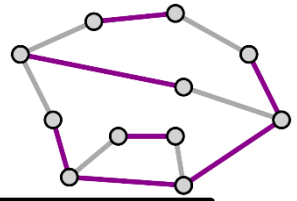


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

for $i = \lceil \log \Delta \rceil, \ldots, 1$

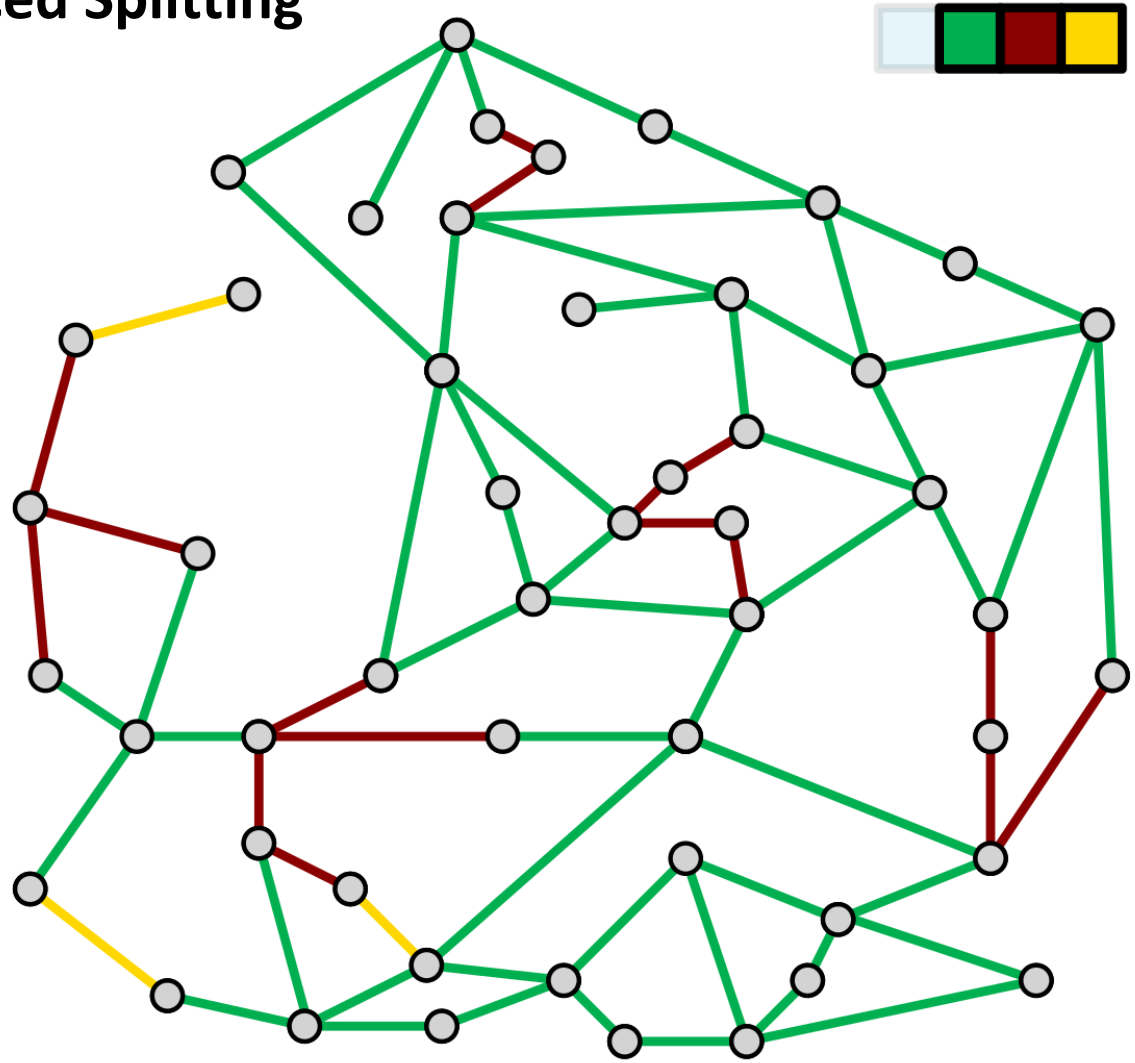
$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

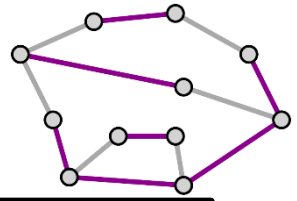


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

for $i = \lceil \log \Delta \rceil, \ldots, 1$

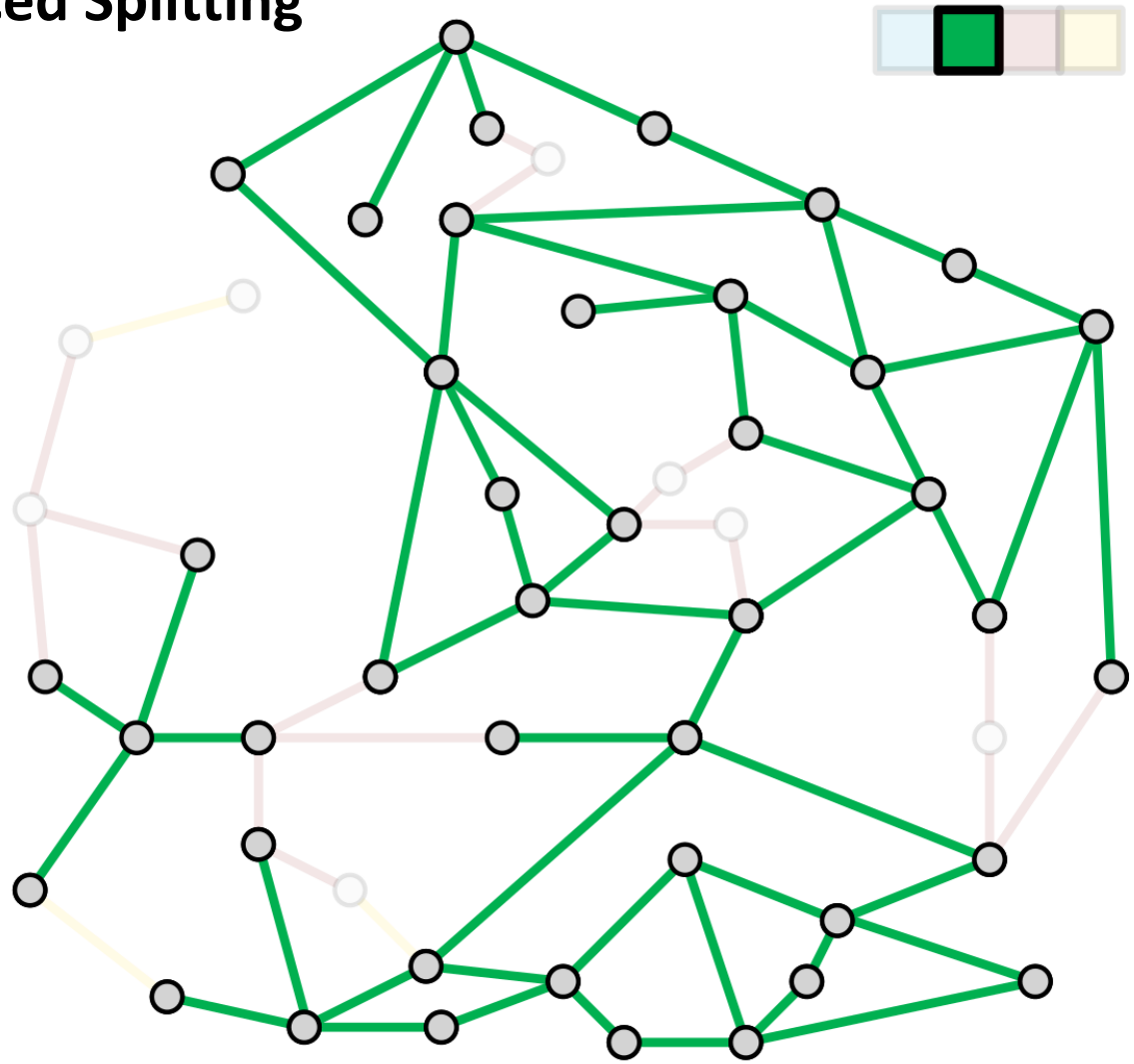
$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

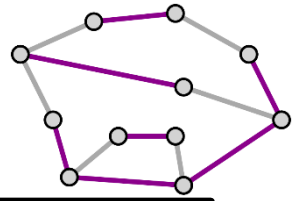


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

for $i = \lceil \log \Delta \rceil, \ldots, 1$

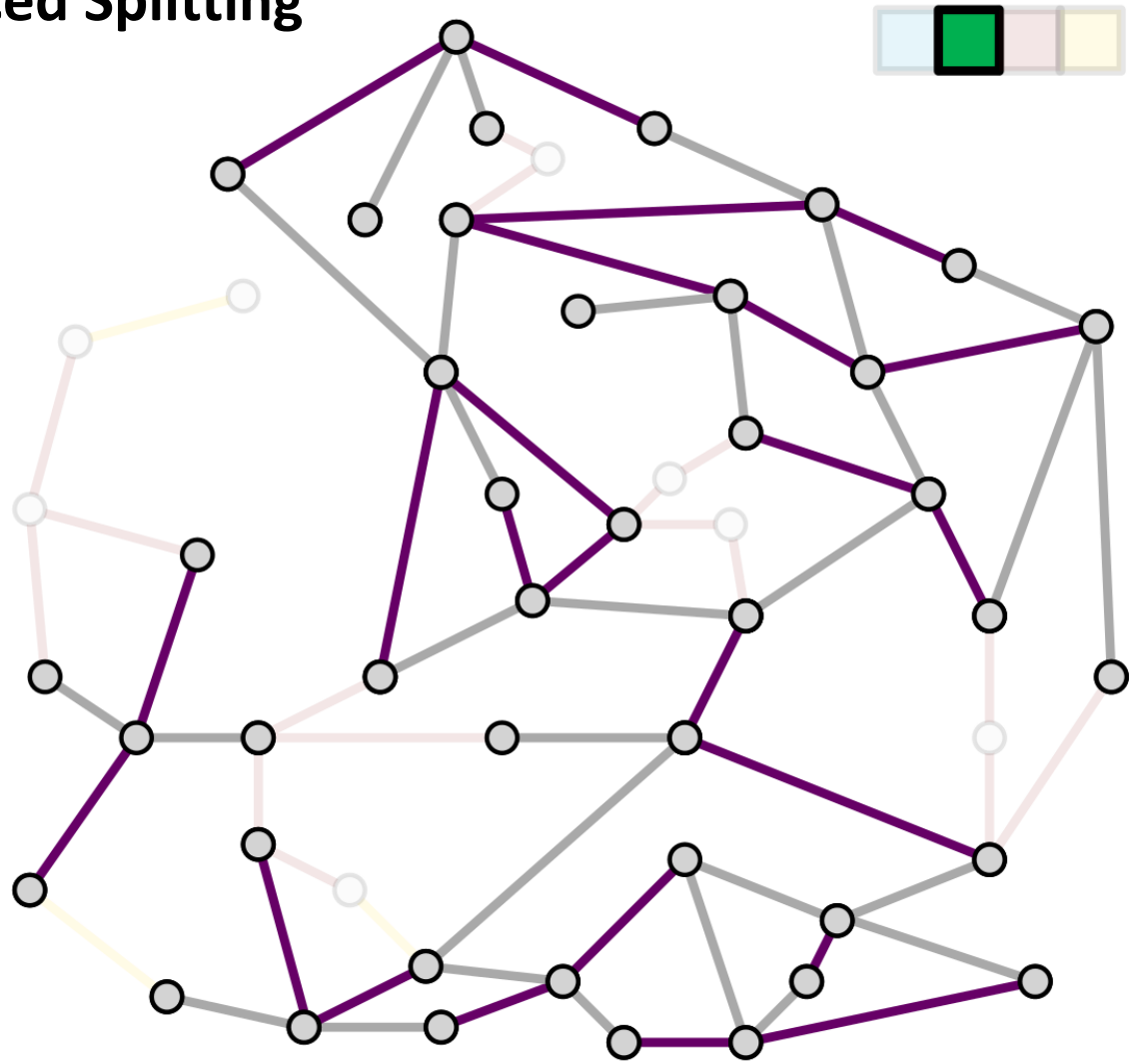
$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

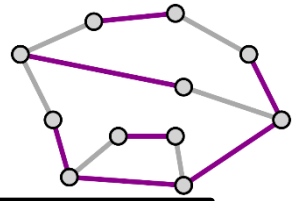


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

for $i = \lceil \log \Delta \rceil, \ldots, 1$

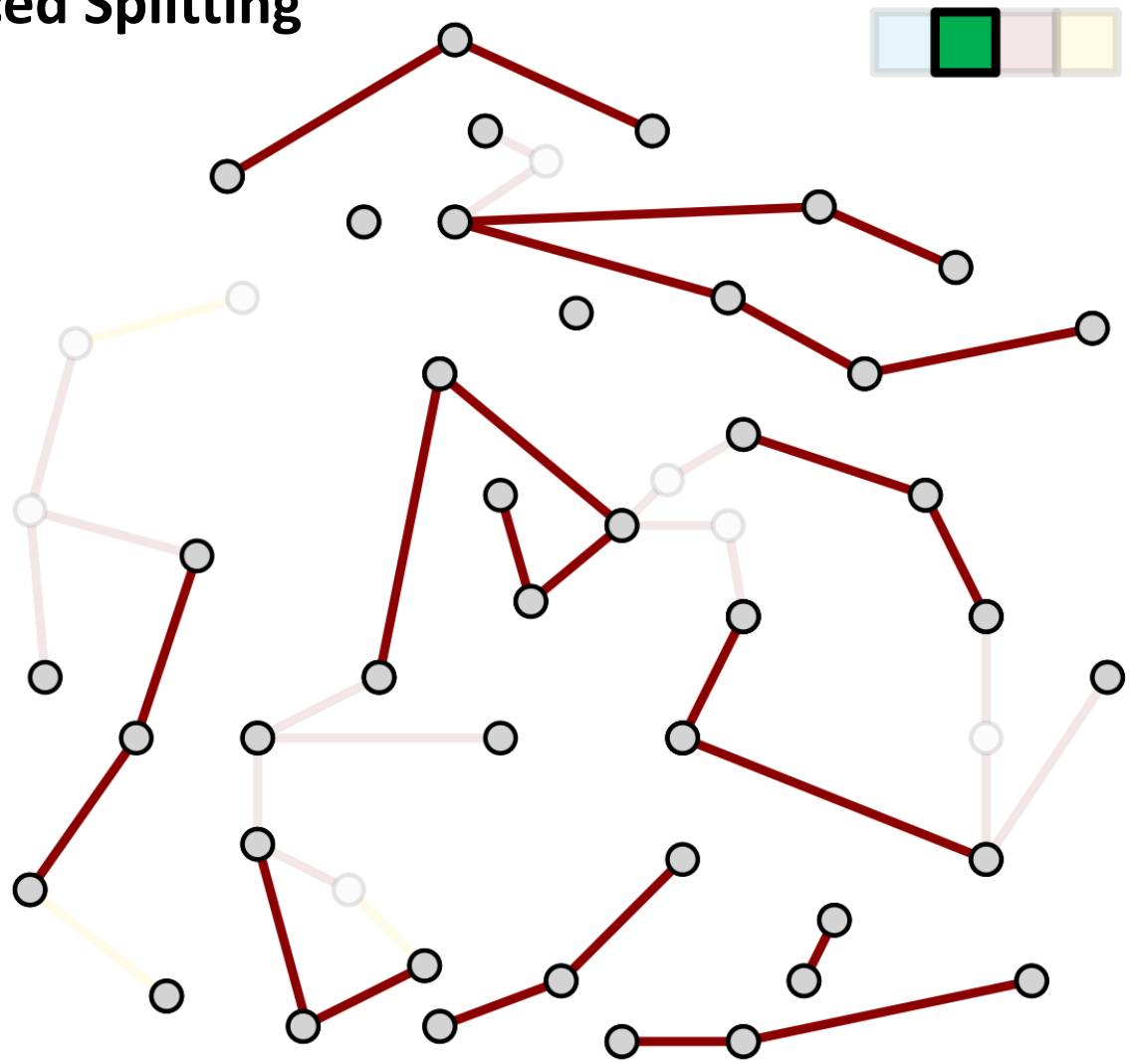
$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

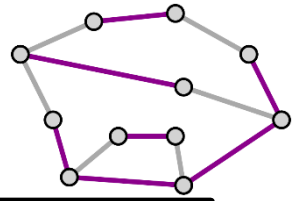


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

for $i = \lceil \log \Delta \rceil, \ldots, 1$

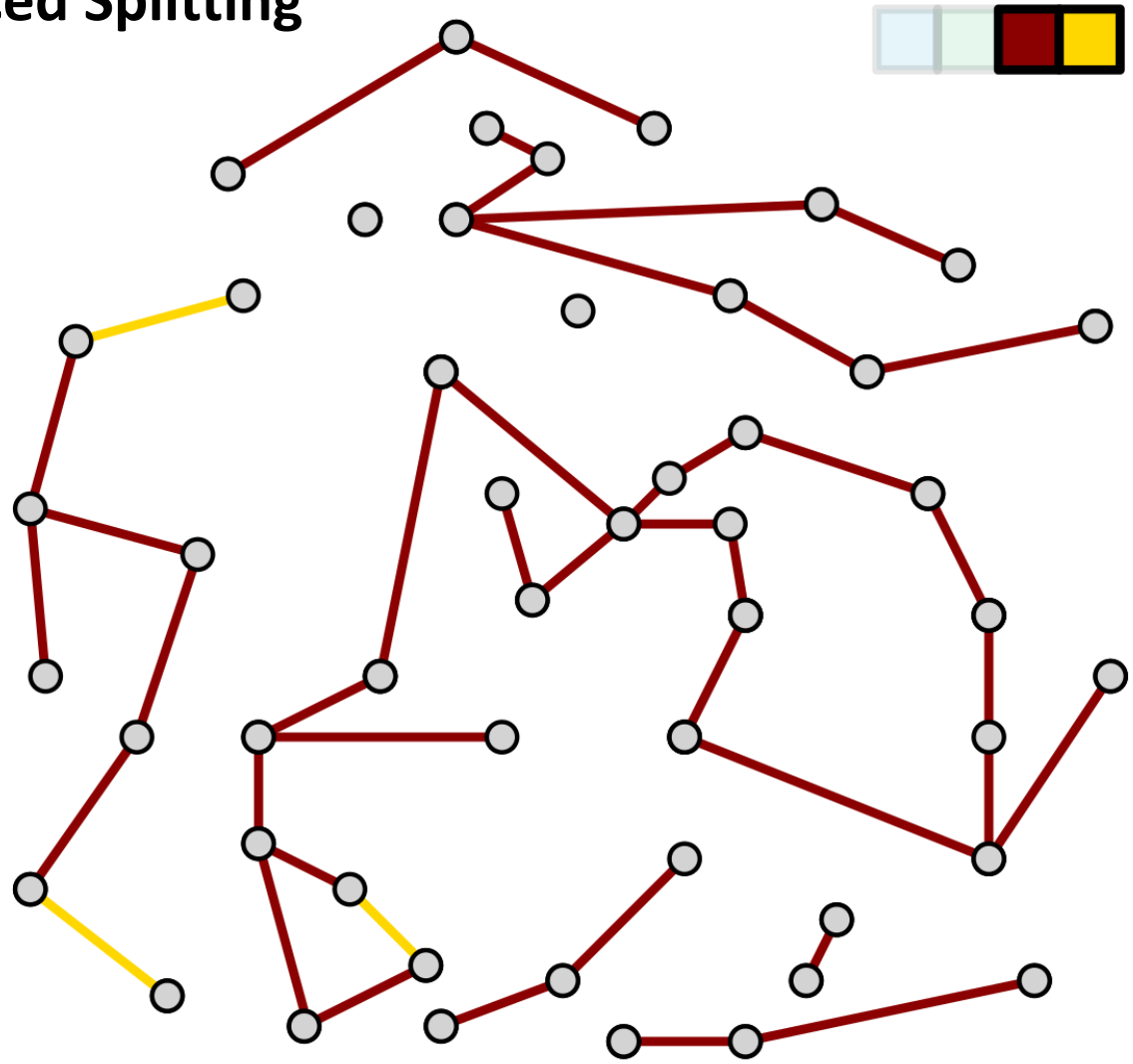
$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

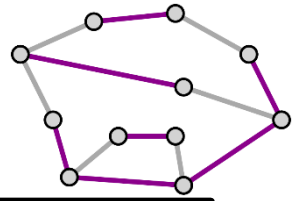


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

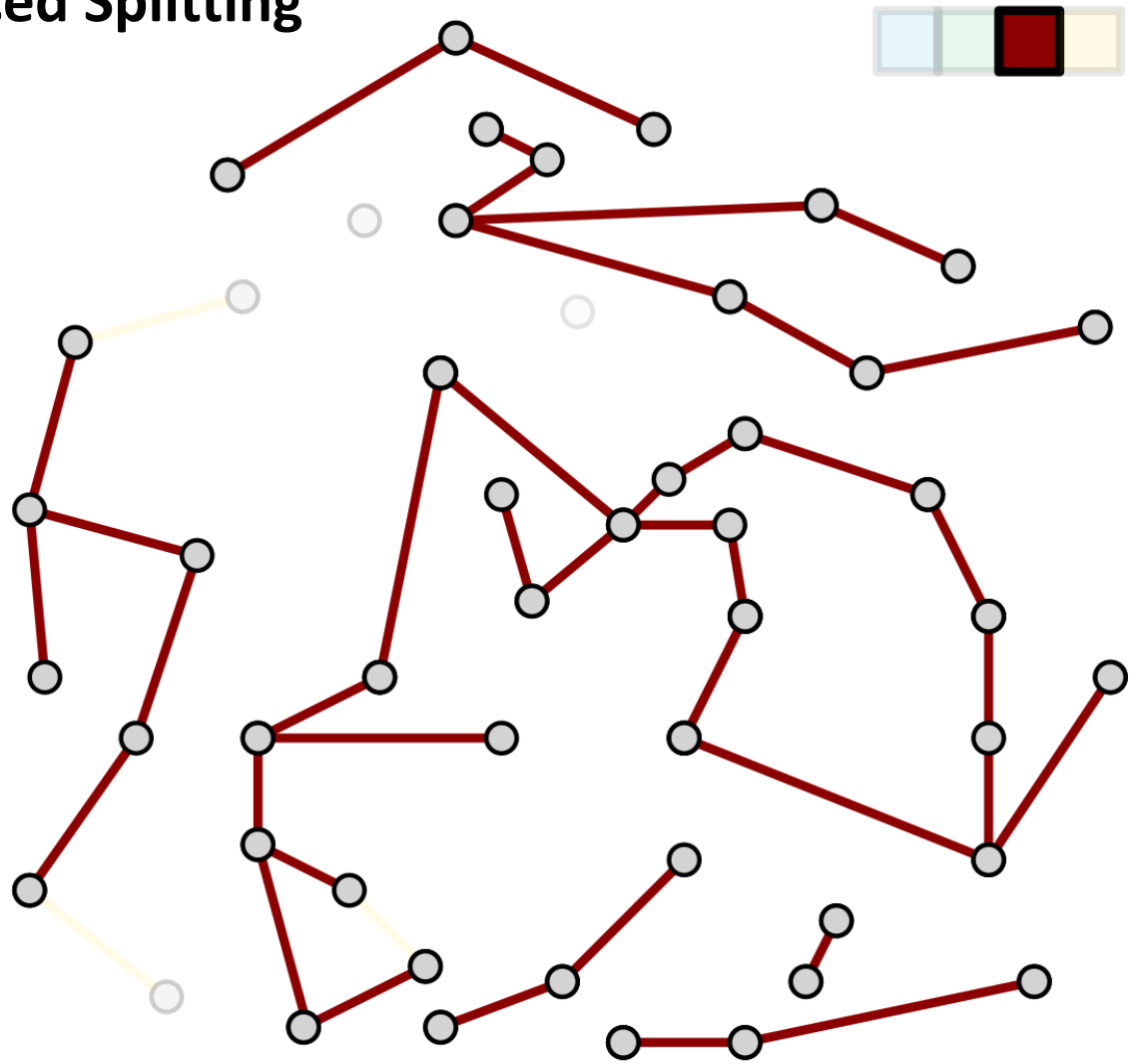
for $i = \lceil \log \Delta \rceil, \ldots, 1$

$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0



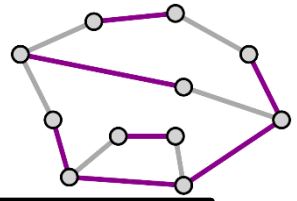
In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

for $i = [\log \Delta], \dots, 1$

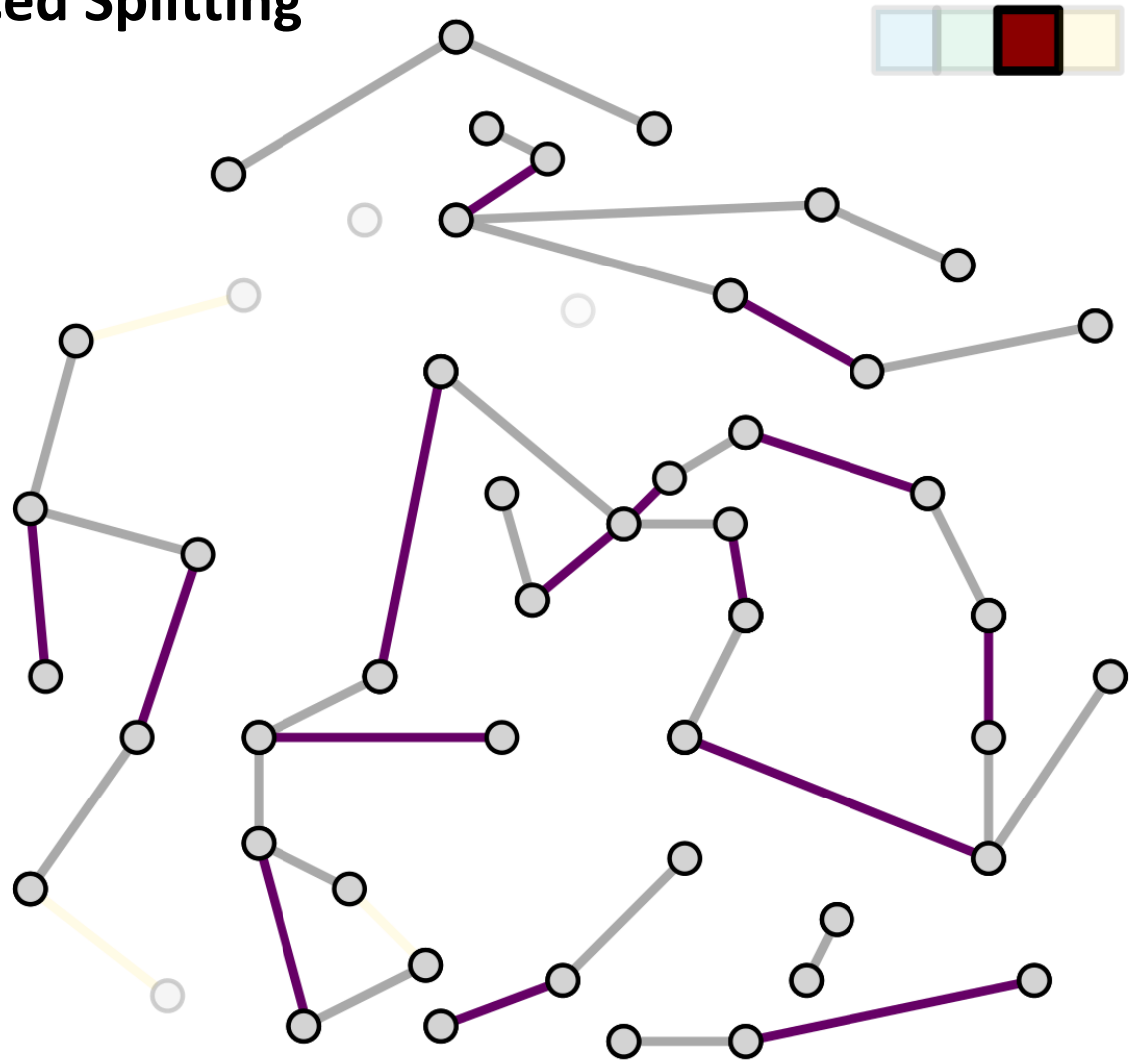
$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

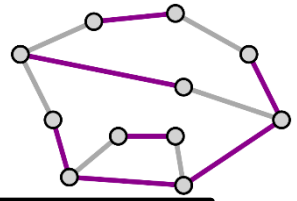


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

for $i = \lceil \log \Delta \rceil, \dots, 1$

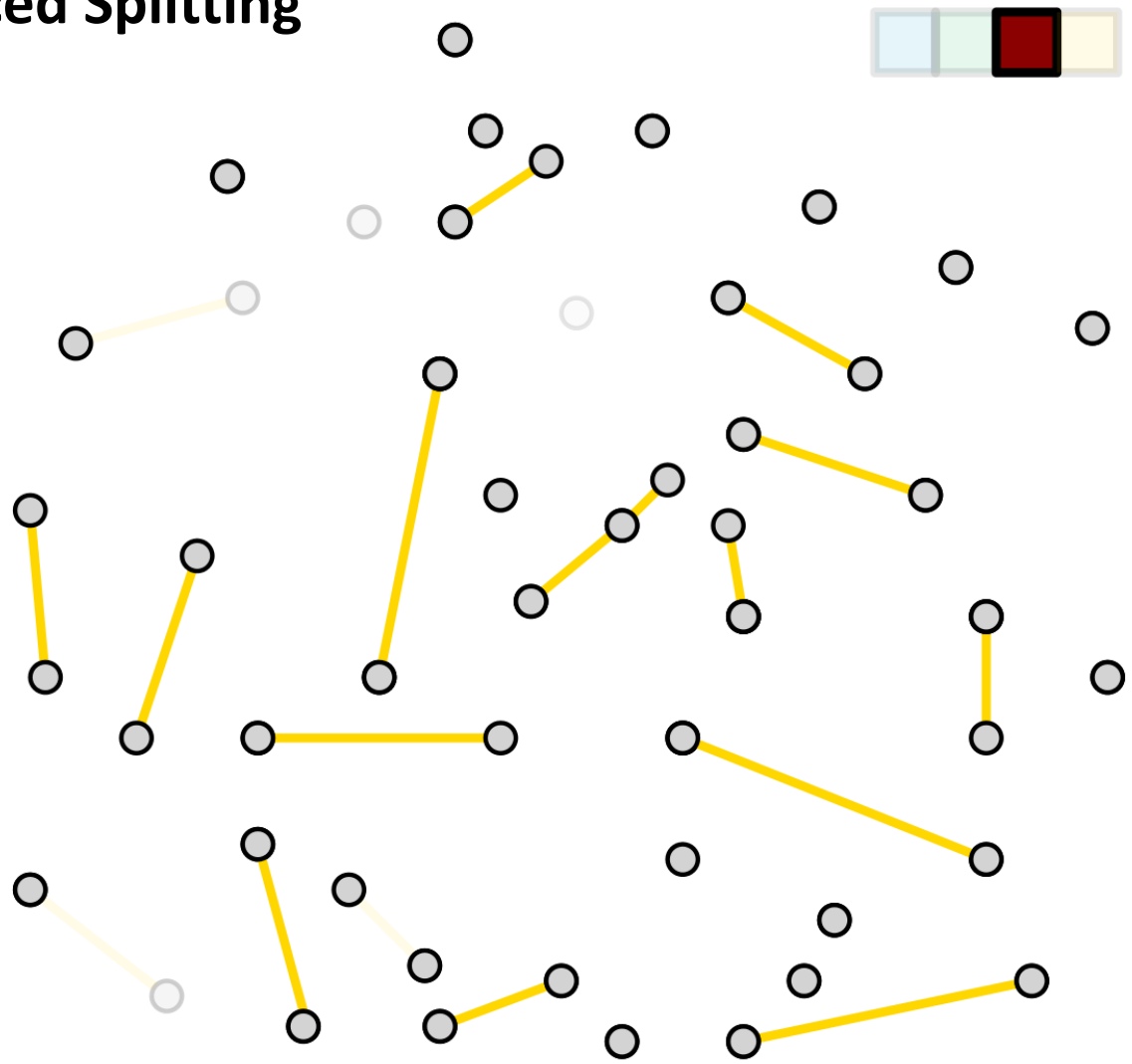
$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

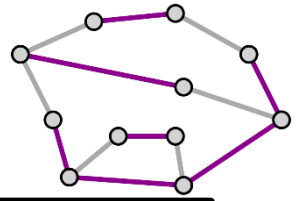


II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

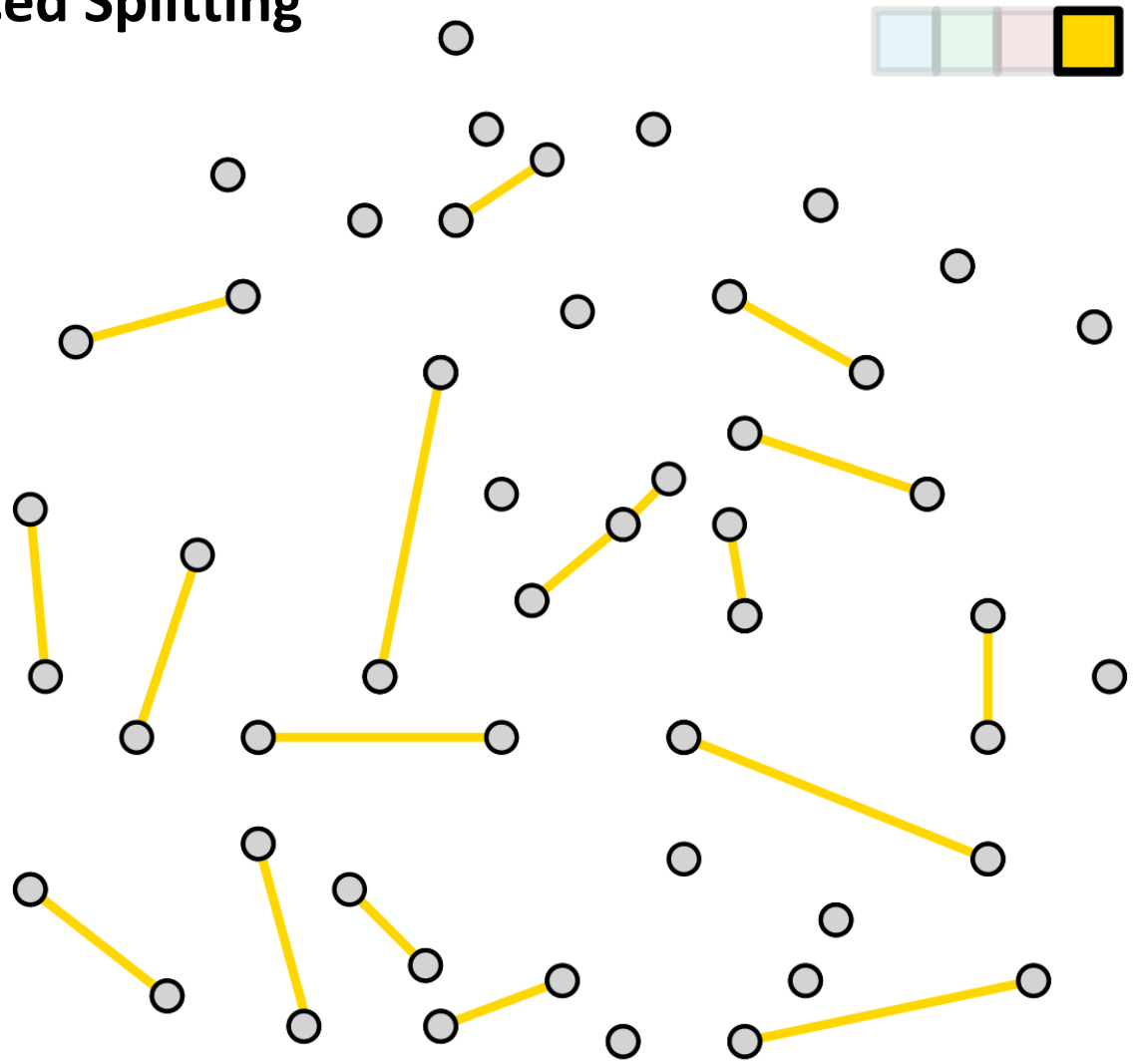
for $i = \lceil \log \Delta \rceil, \ldots, 1$

$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0



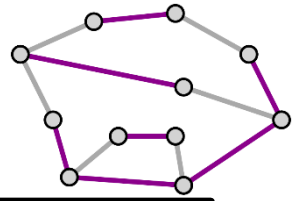
In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

Locally Balanced Splitting:
2-edge-coloring so that
every node roughly balanced



Iterated Factor-2-Rounding

for $i = \lceil \log \Delta \rceil, \ldots, 1$

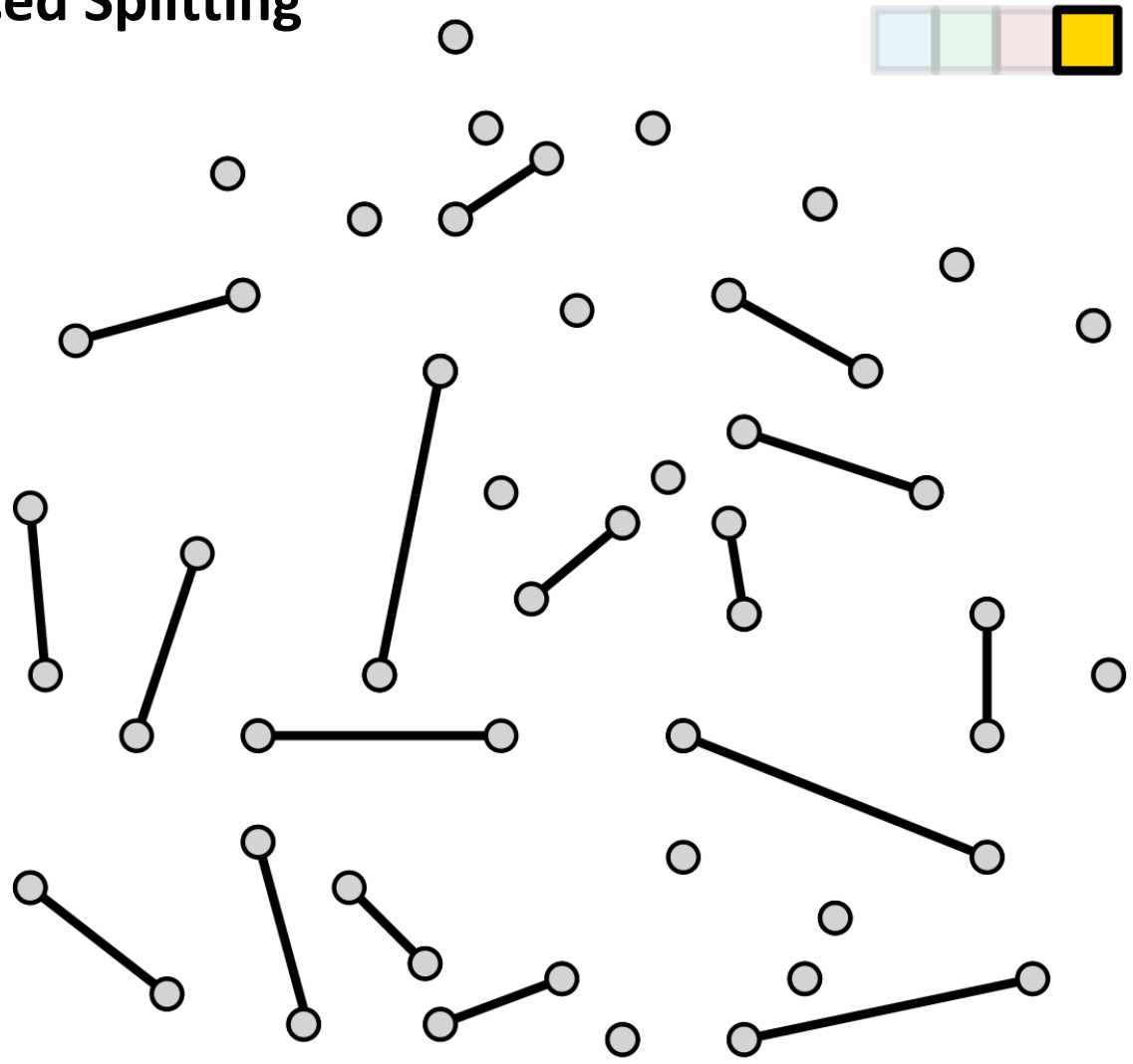
$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into ■□

increase ■ to 2^{-i+1}

decrease □ to 0

In case of **perfect** locally balanced **splitting**:
no constraint violated & no loss in total value
(i.e., **perfect rouding**)



II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

II) Rounding Fractional Bipartite Matching

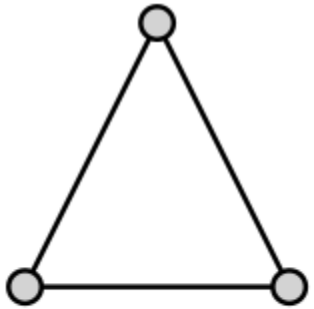
$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Perfect Splitting not possible in case of…

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Perfect Splitting not possible in case of…

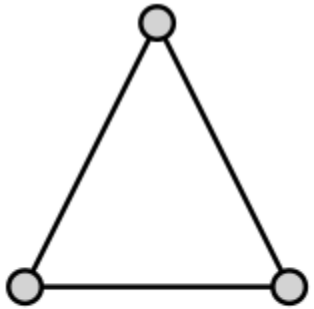


… odd cycles

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Perfect Splitting not possible in case of…



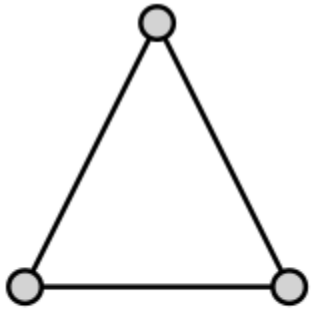
… odd cycles

bipartite graph!

II) Rounding Fractional Bipartite Matching

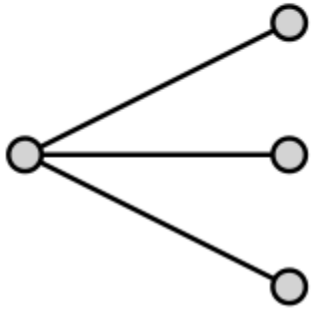
$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Perfect Splitting not possible in case of…



… odd cycles

bipartite graph!

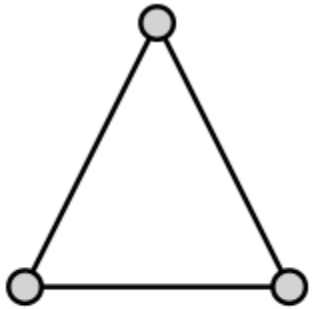


… odd-degree vertices

II) Rounding Fractional Bipartite Matching

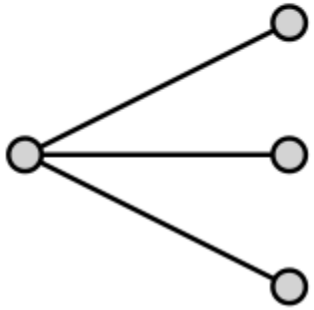
$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Perfect Splitting not possible in case of…



… odd cycles

bipartite graph!



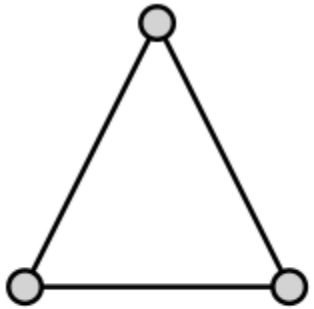
… odd-degree vertices

small technicality.

II) Rounding Fractional Bipartite Matching

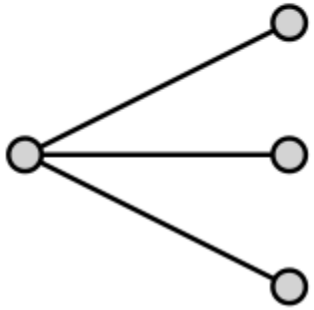
$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Perfect Splitting not possible in case of…



… odd cycles

bipartite graph!



… odd-degree vertices

small technicality.

Suppose that bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
  pick arbitrary cycle
  alternate 

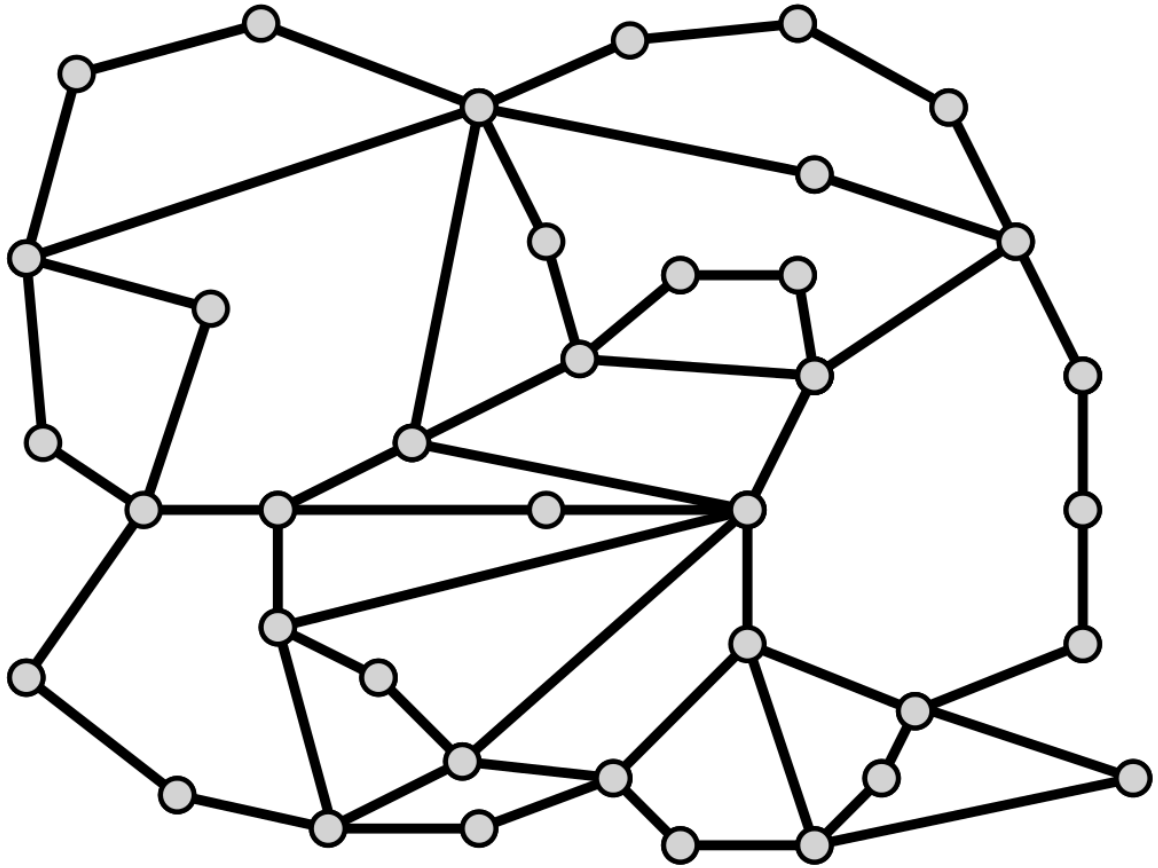
* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
  pick arbitrary cycle
  alternate ■□



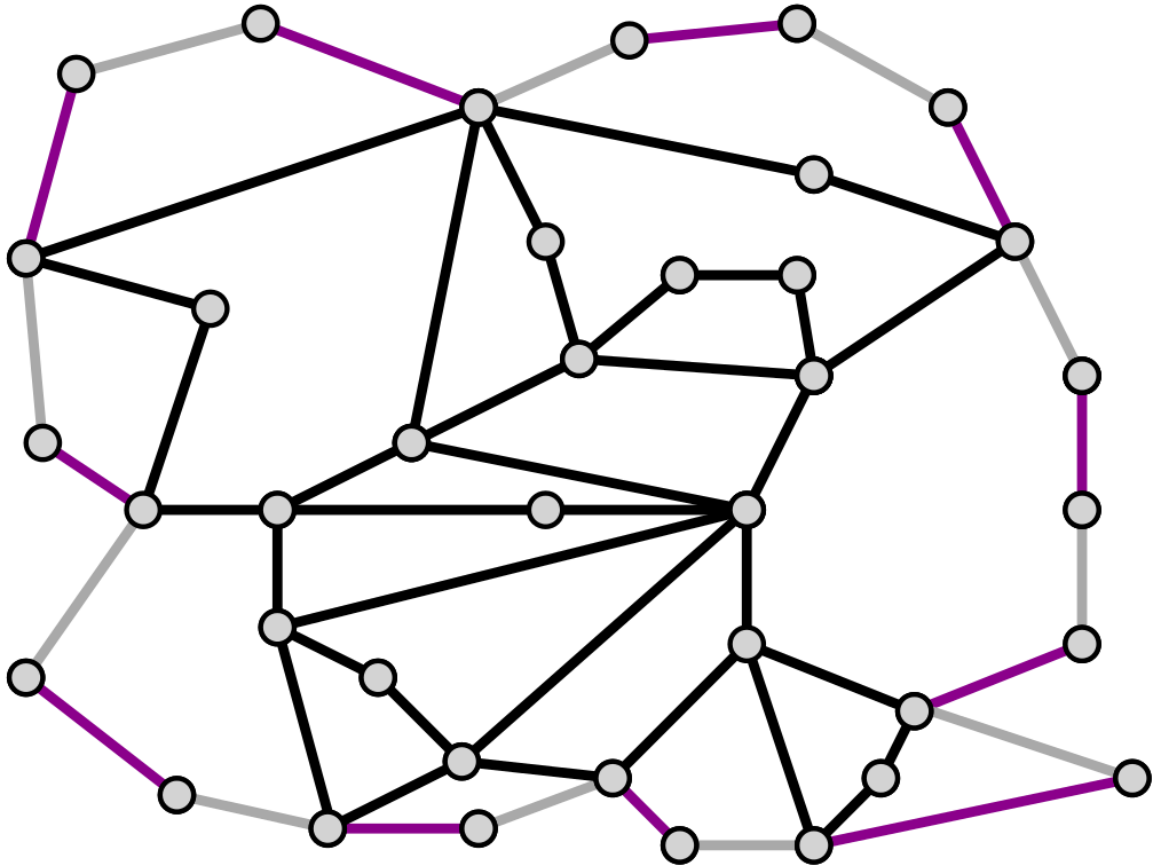
* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□



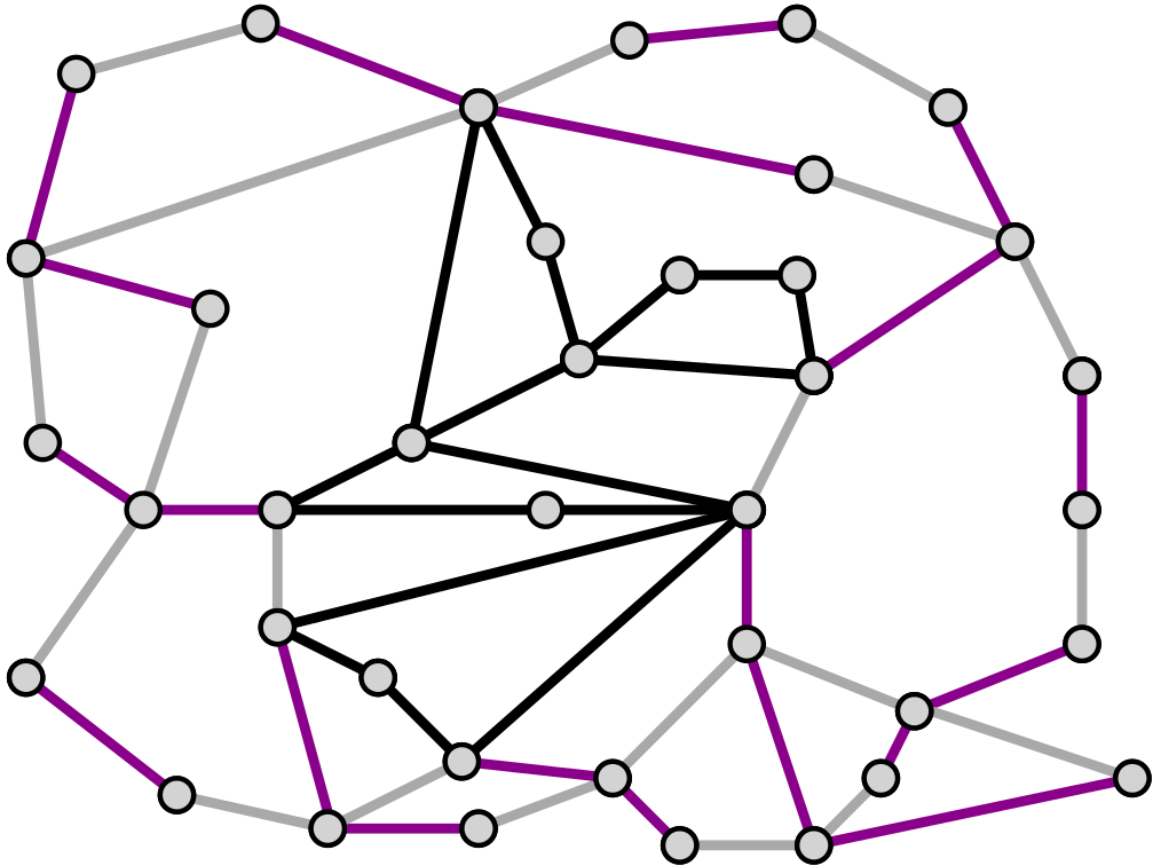
* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
  pick arbitrary cycle
  alternate ■□



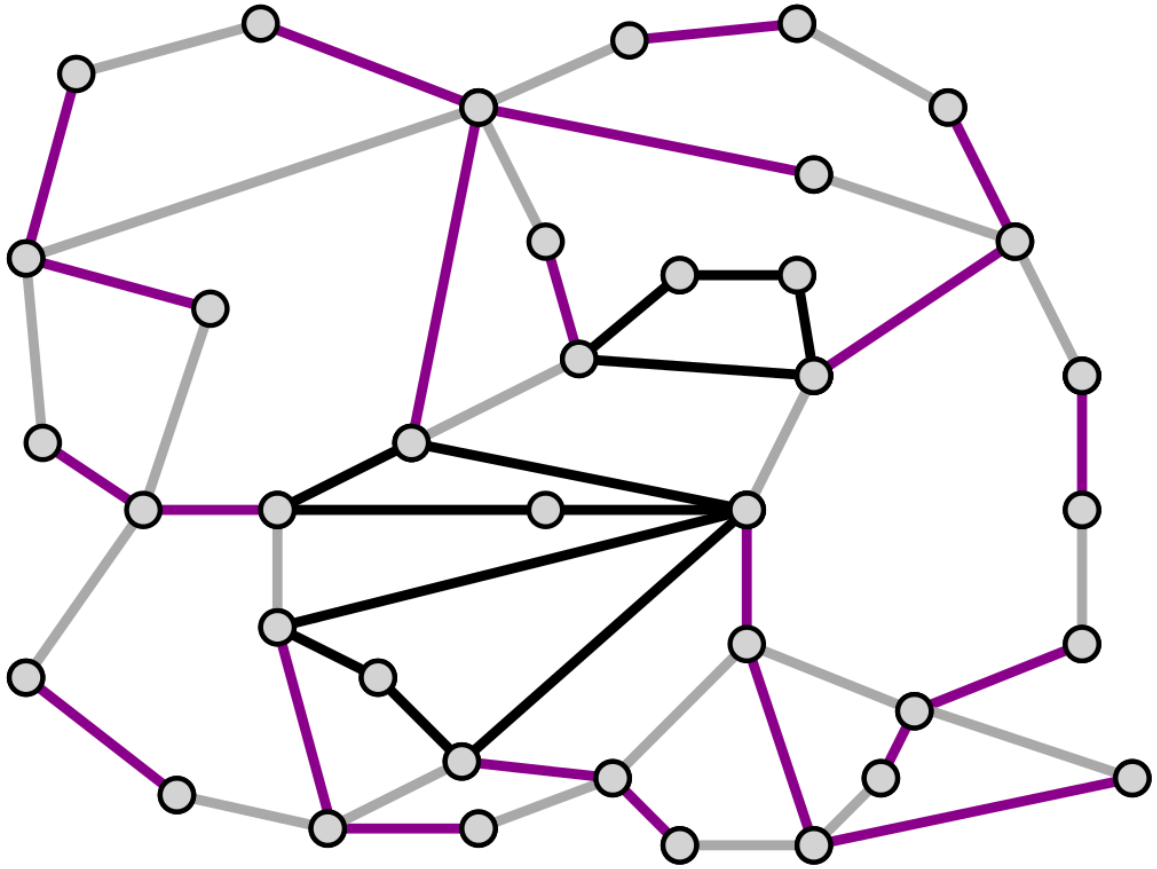
* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□



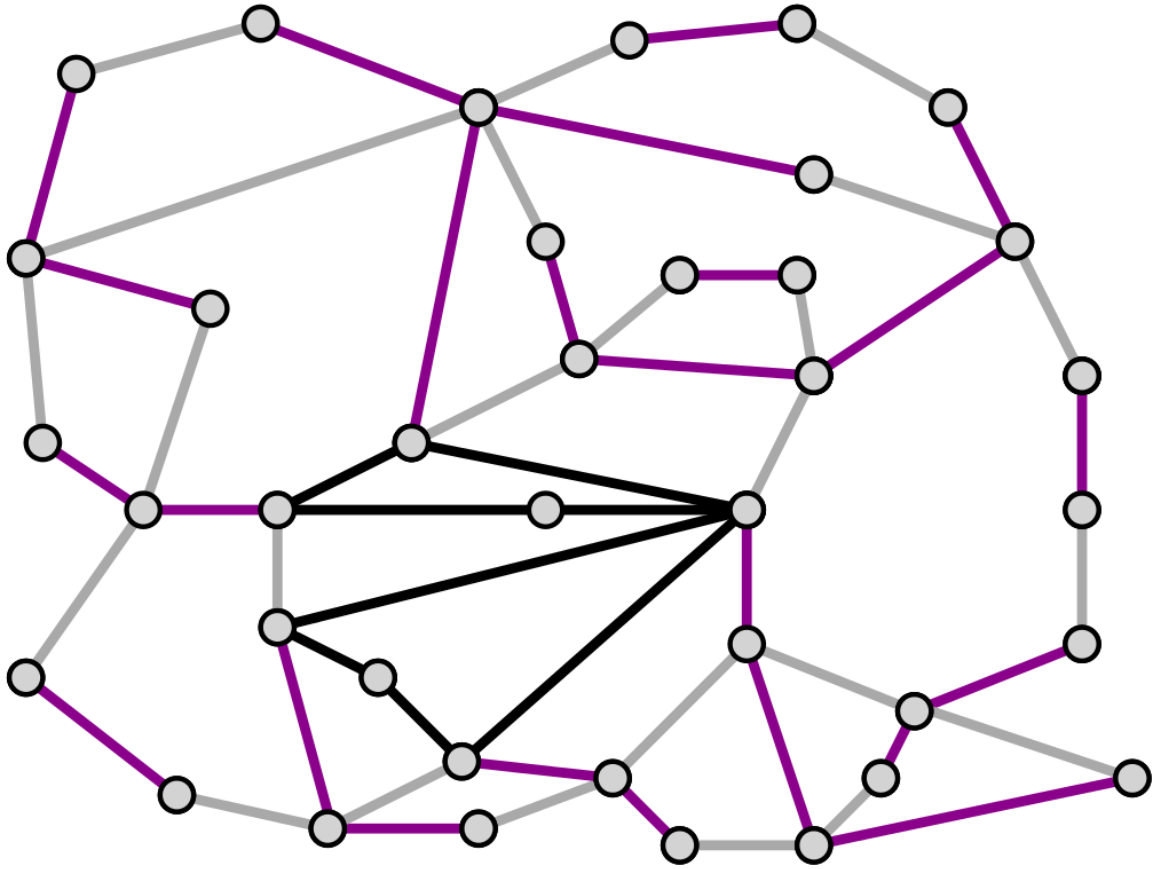
* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
  pick arbitrary cycle
  alternate ■□



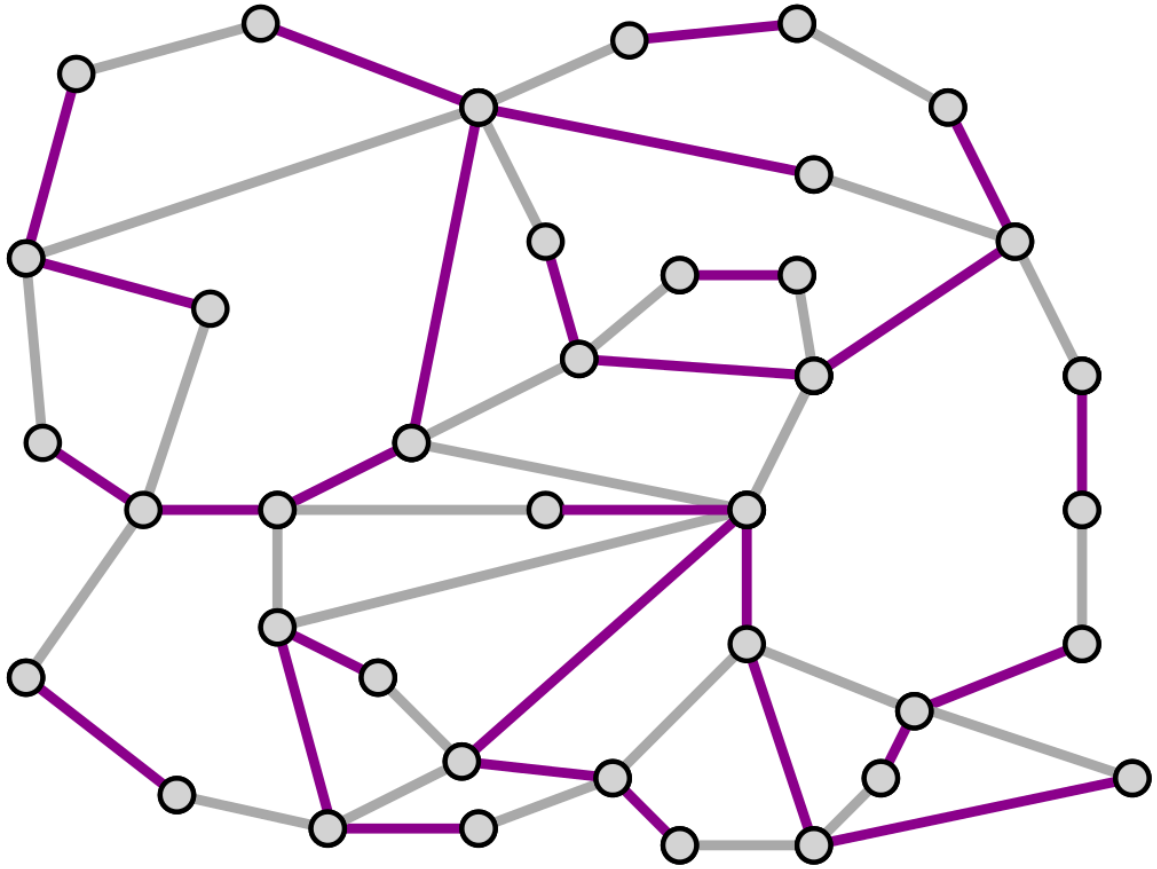
* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
  pick arbitrary cycle
  alternate ■□



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate 

LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate 

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate  in between

* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

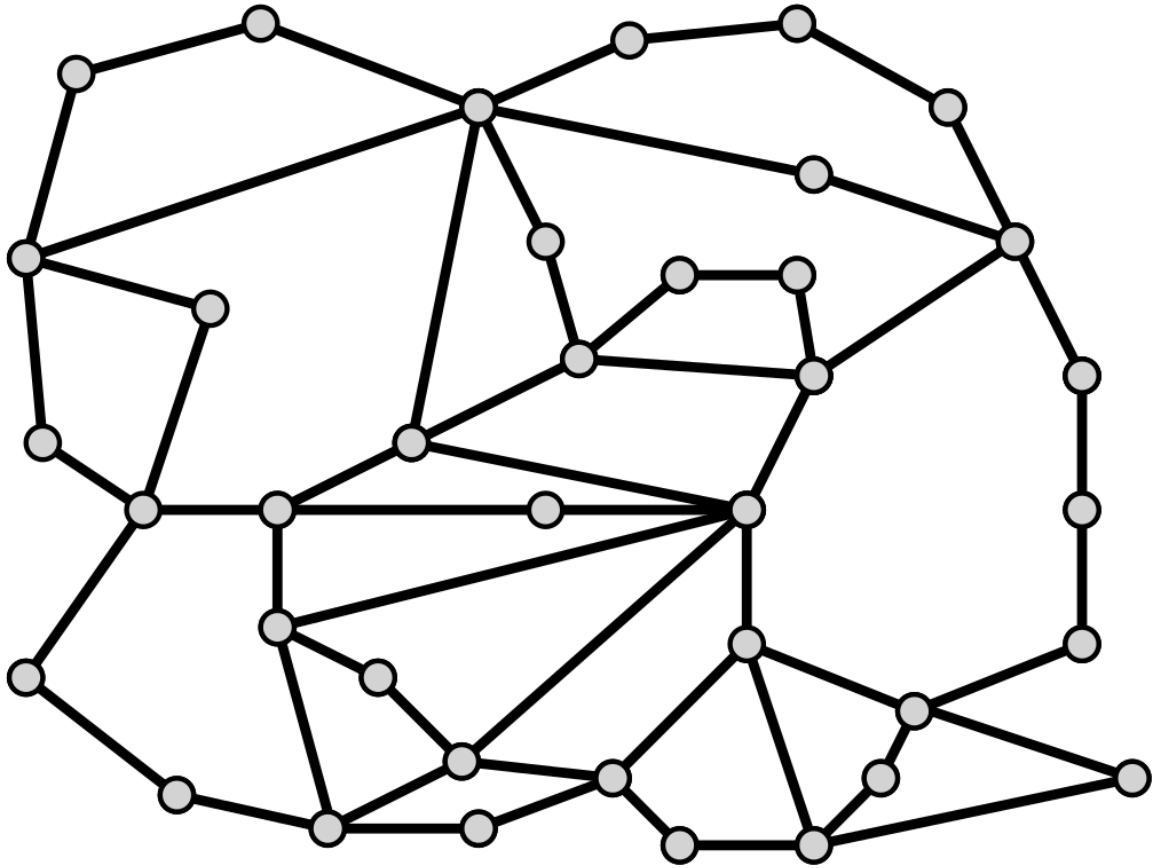
LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

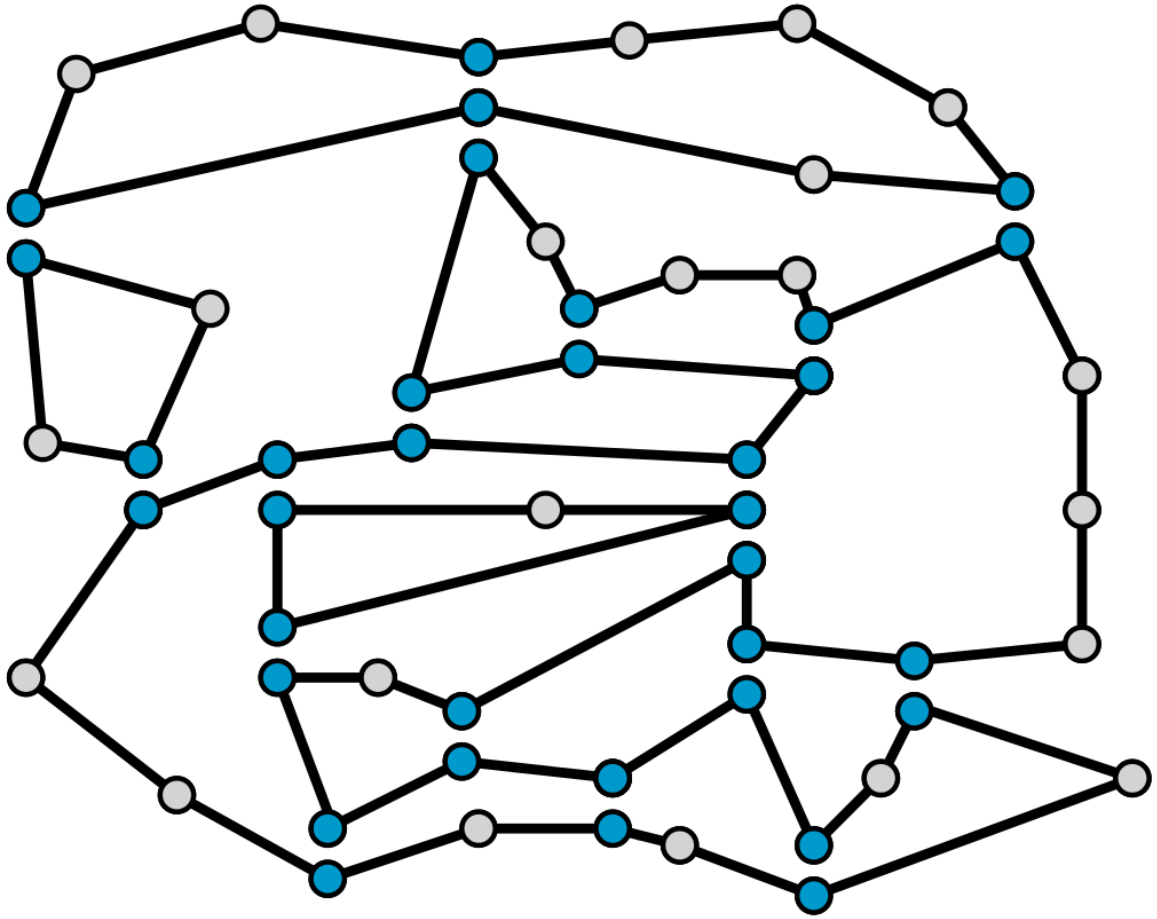
LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

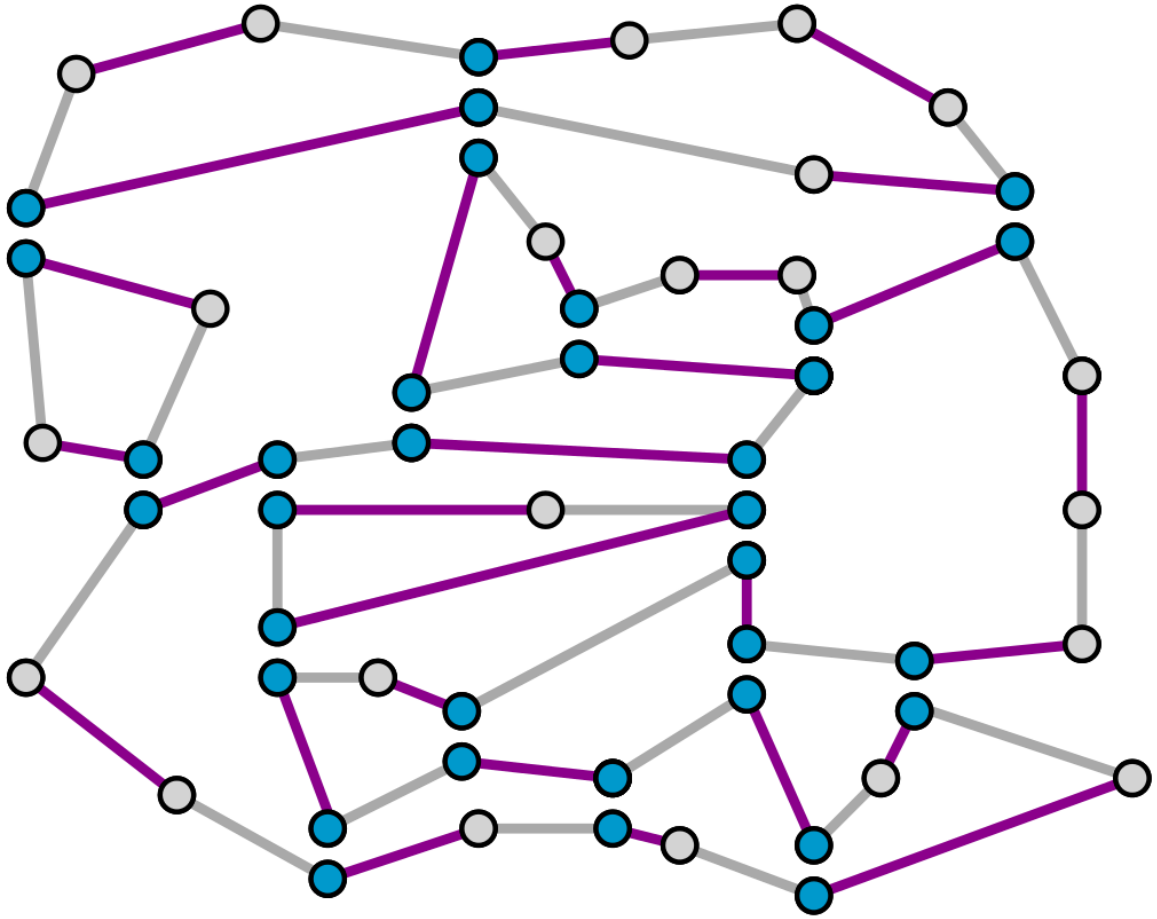
LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

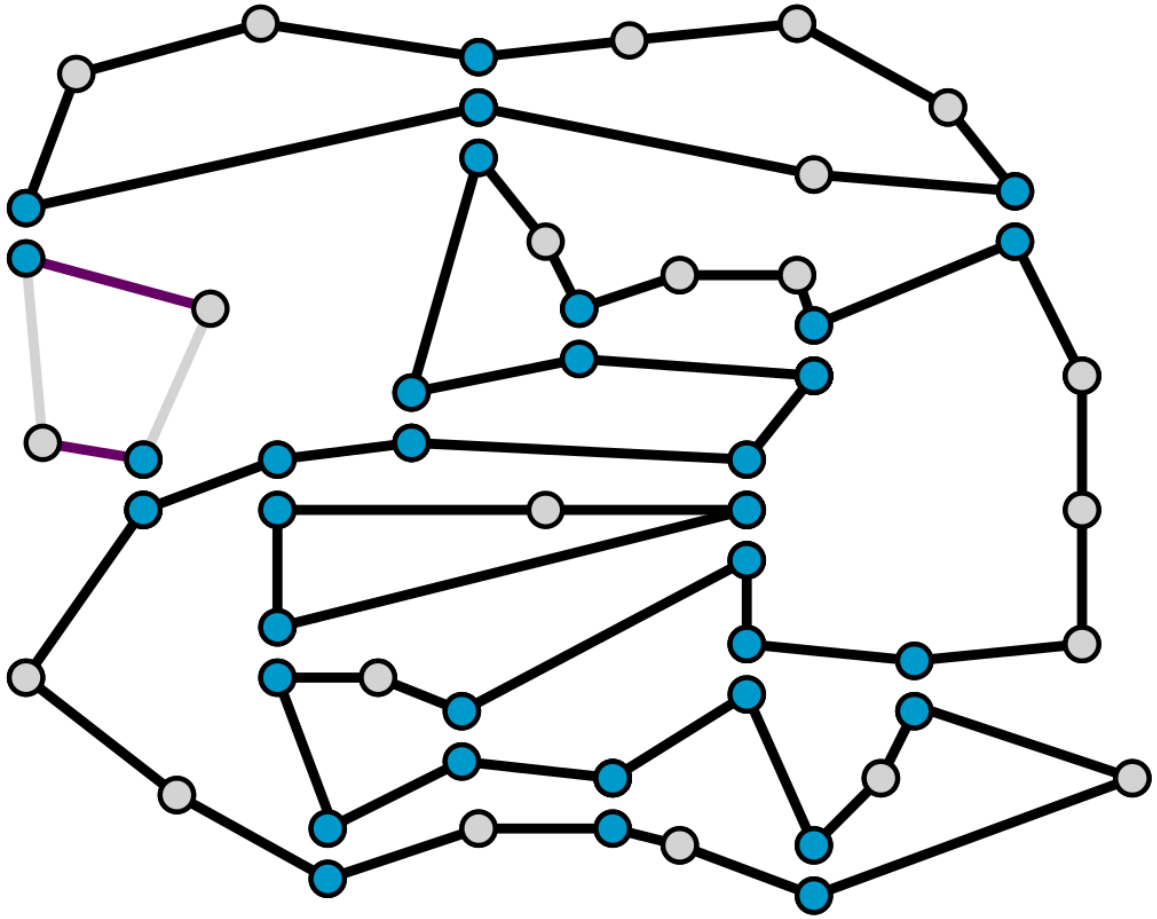
LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

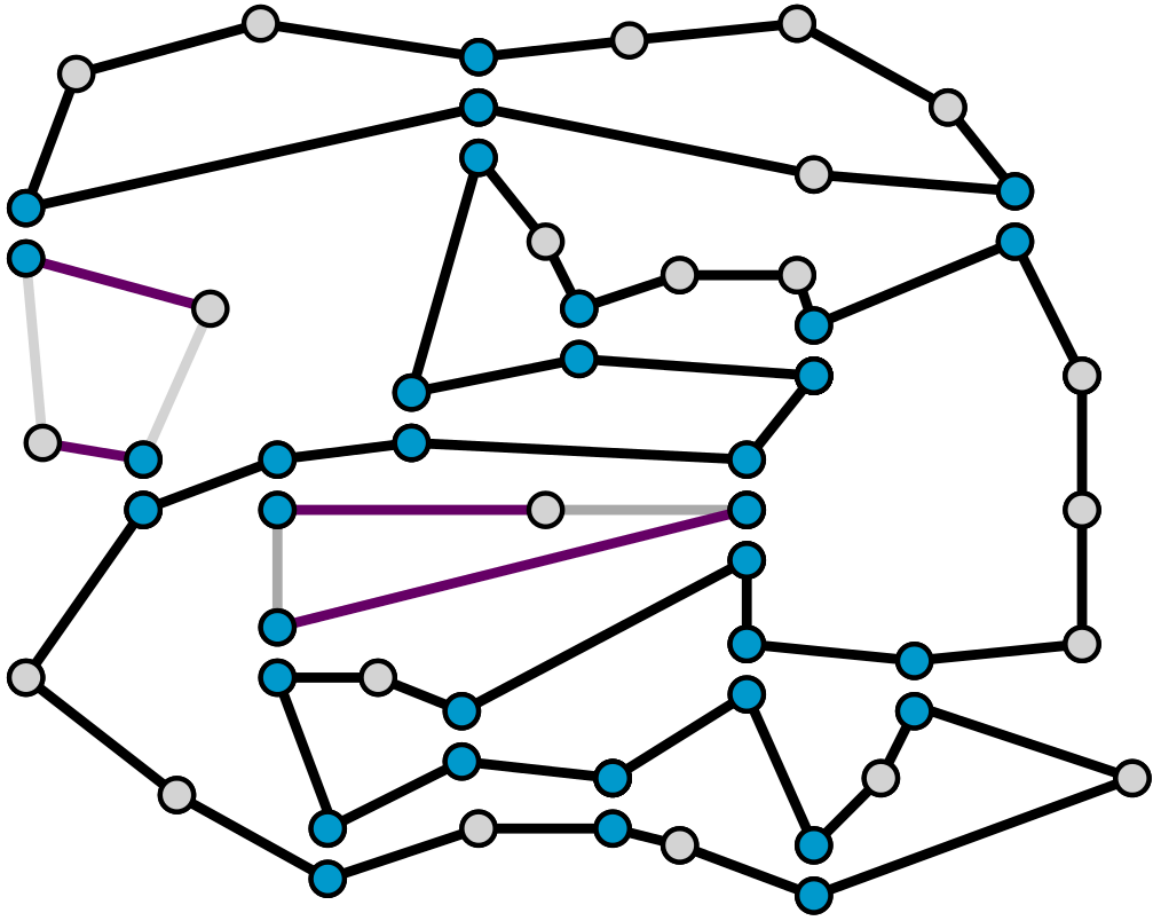
LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

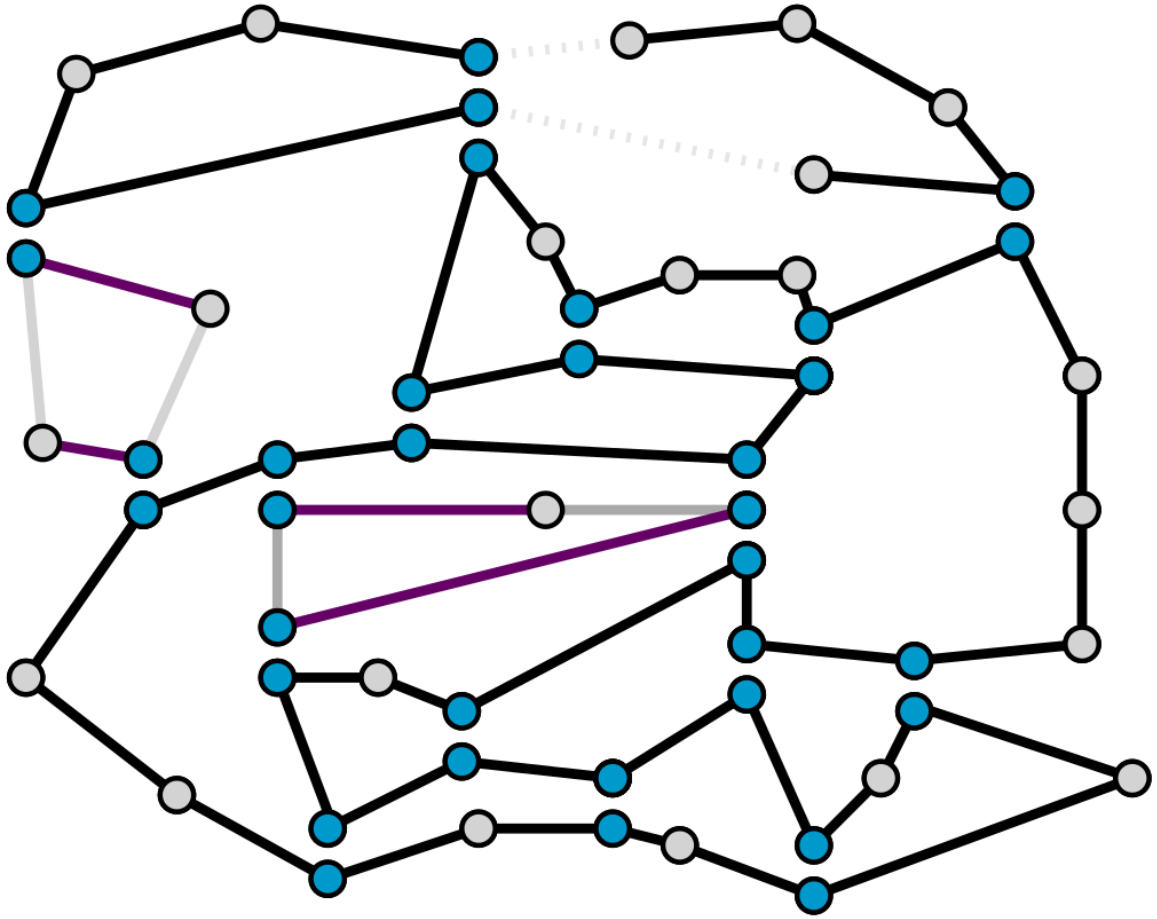
LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

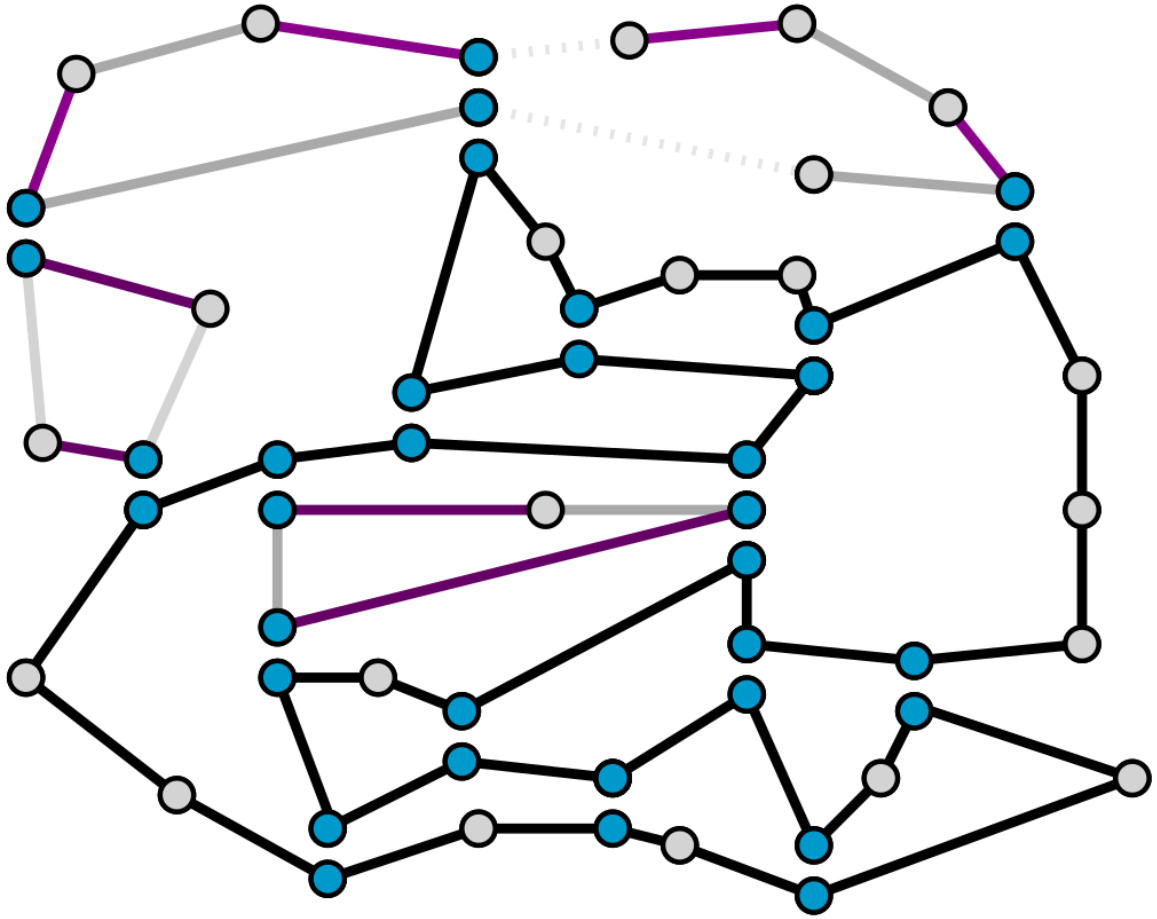
LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

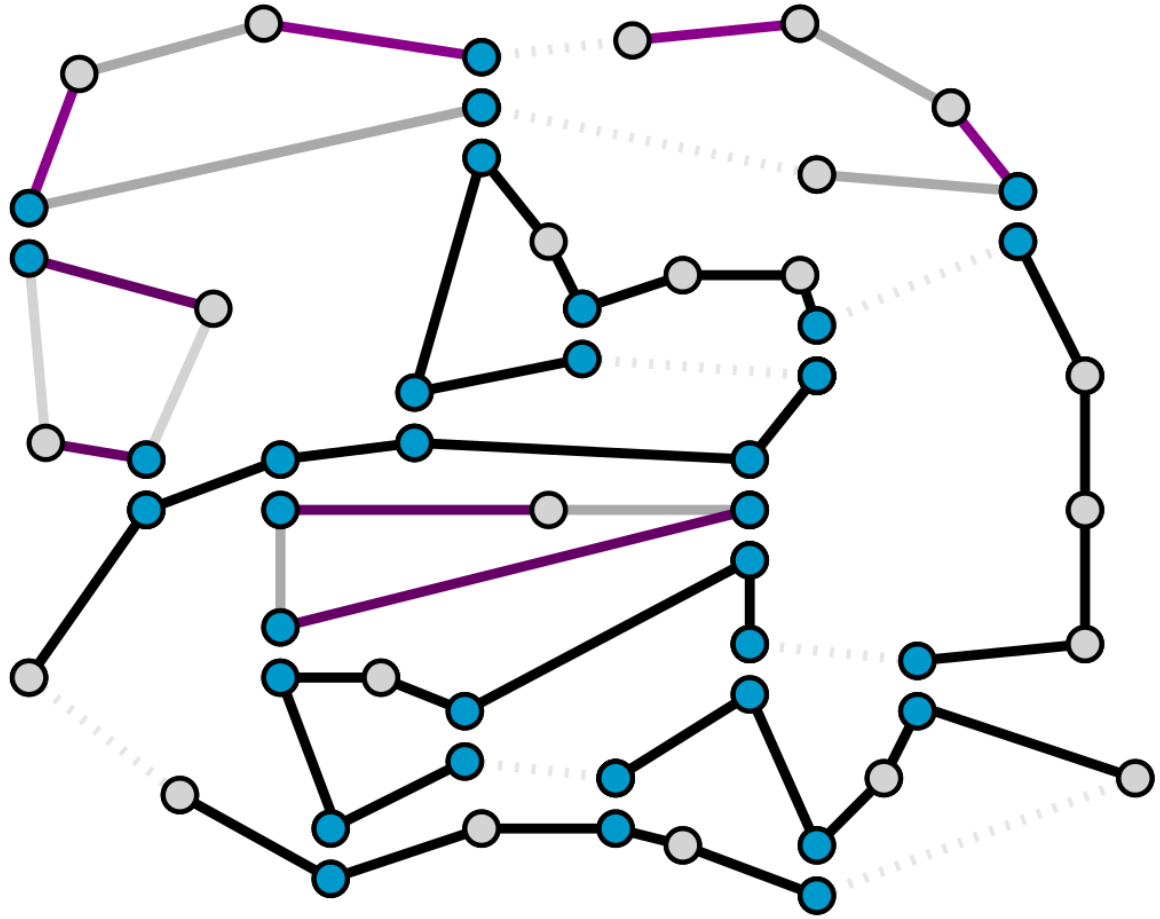
LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

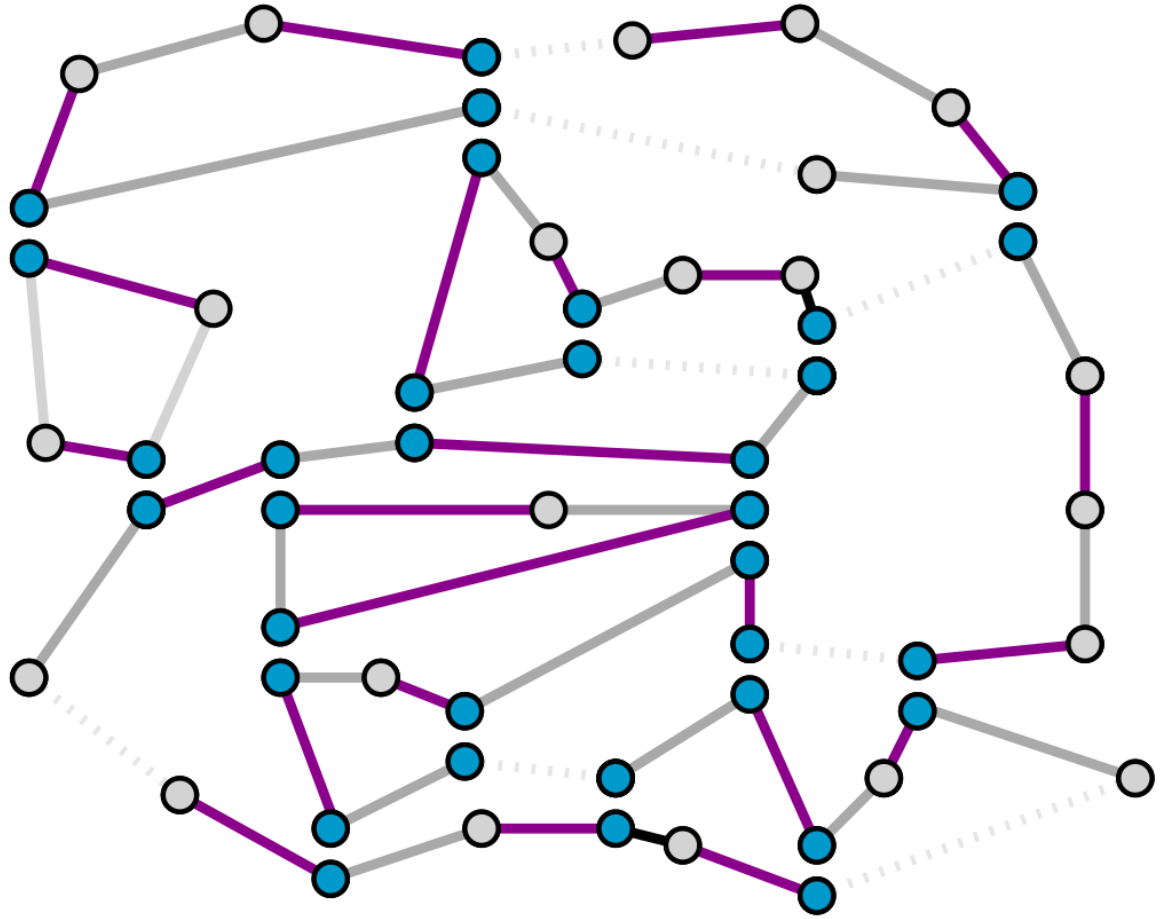
LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

LOCAL Almost-Perfect Splitting*

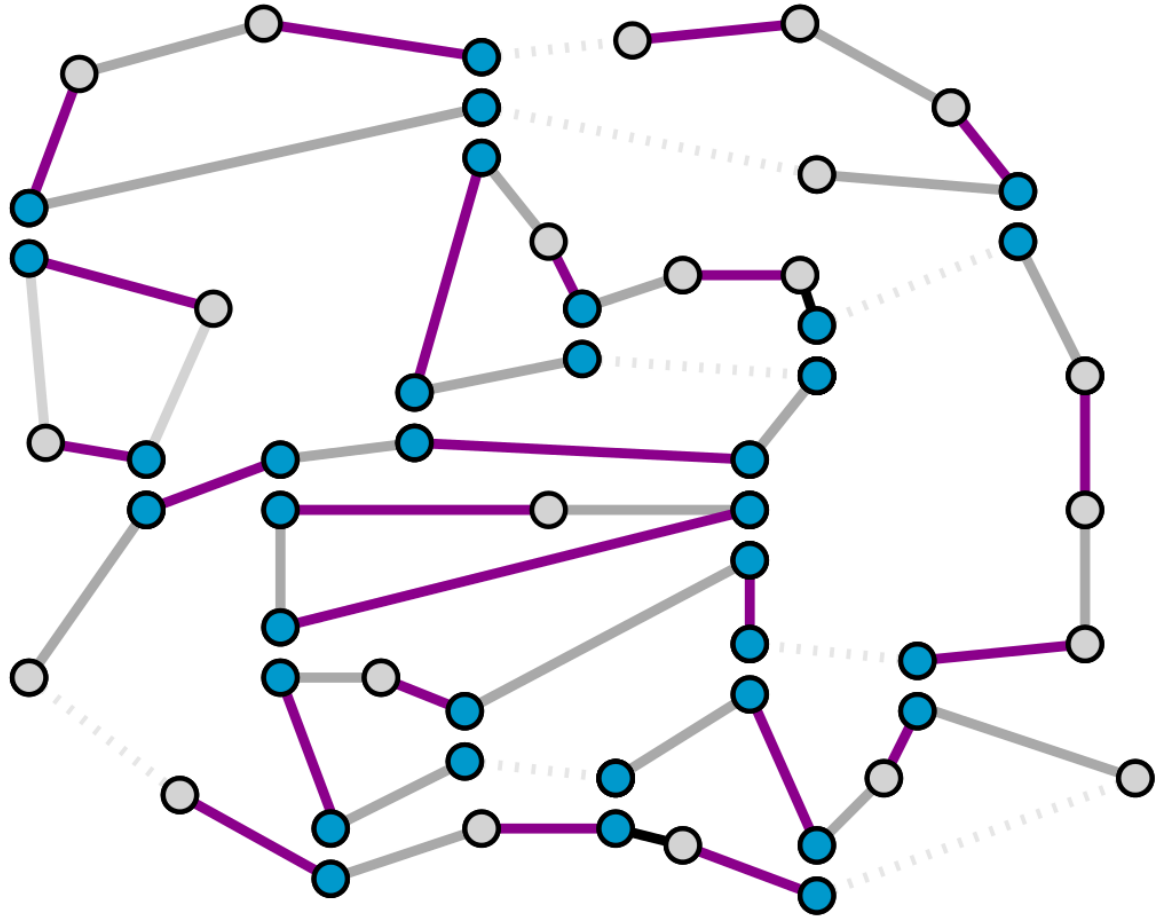
Decompose into edge-disjoint cycles

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$
- alternate ■□

B) **Long cycles**
- chop at length $\Theta(\log \Delta)$
- set boundary to 0
- alternate ■□ in between

$\Theta\left(\frac{1}{\log \Delta}\right)$ loss



* bipartite and even degree!

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

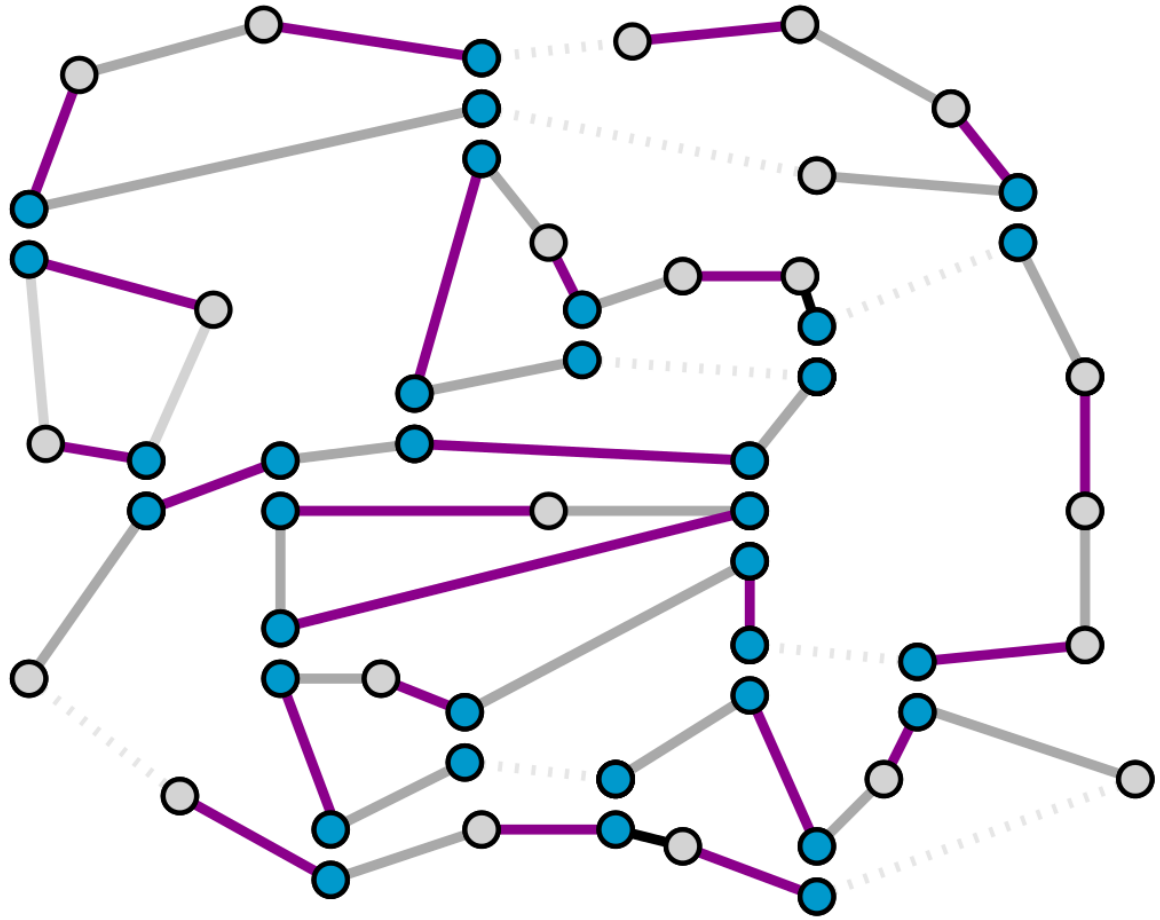
Sequential Perfect Splitting*

Repeat until all edges colored
  pick arbitrary cycle
  alternate ■□

LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles
In parallel, for all cycles
  A) **Short cycles** of length $O(\log \Delta)$
    alternate ■□
  B) **Long cycles**
    chop* at length $\Theta(\log \Delta)$
    set boundary to 0
    alternate ■□ in between

$\Theta\left(\frac{1}{\log \Delta}\right)$ loss



* by *Hańćkowiak, Karoński, Panconesi* [SODA'98,PODC'99] in $O(\log \Delta)$

*** bipartite and even degree!**

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

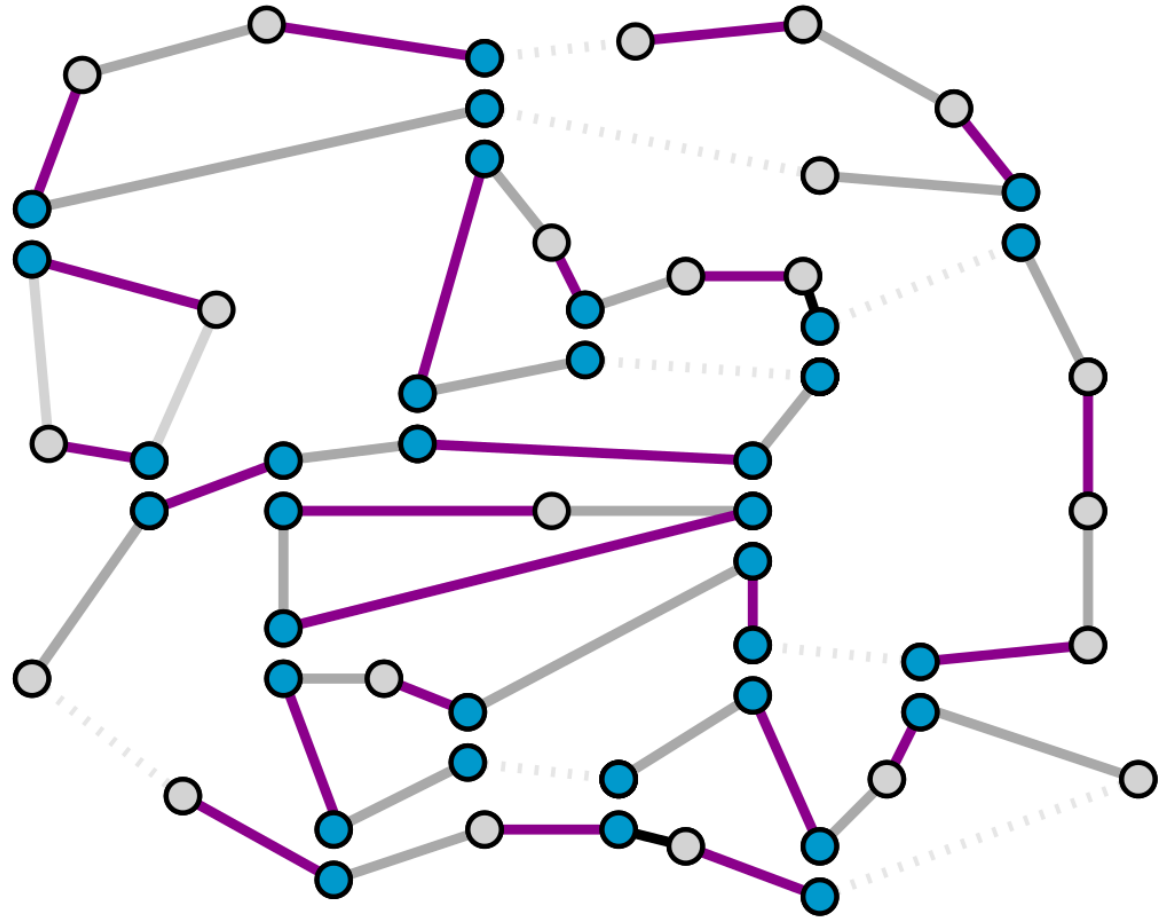
Sequential Perfect Splitting*

Repeat until all edges colored
- pick arbitrary cycle
- alternate ■□

LOCAL Almost-Perfect Splitting*

Decompose into edge-disjoint cycles
In parallel, for all cycles
- A) **Short cycles** of length $O(\log \Delta)$
  - alternate ■□
- B) **Long cycles**
  - chop* at length $\Theta(\log \Delta)$
  - set boundary to 0
  - alternate ■□ in between

$\Theta\left(\frac{1}{\log \Delta}\right)$ loss



* by *Hańćkowiak, Karoński, Panconesi* [SODA'98,PODC'99] in $O(\log \Delta)$

Over all $O(\log \Delta)$ rounding iterations, total loss still constant!

*** bipartite and even degree!**

Summary

Deterministic LOCAL Complexity of Matching

- $O(\log^2 \Delta \cdot \log n)$ for Maximal Matching
- $O(\log^2 \Delta + \log^* n)$ for constant-approximate Matching
  - exponentially improving over $O(\Delta + \log^* n)$
    Panconesi, Rizzi [DIST'01]
  - close to $\Omega\left(\frac{\log \Delta}{\log \log \Delta} + \log^* n\right)$
    Kuhn, Moscibroda, Wattenhofer [PODC'04]
    Linial [FOCS'87]

Deterministic Rounding

- First efficient deterministic rounding method
- Rounding as **the only obstacle** for efficient deterministic LOCAL graph algorithms.
  Ghaffari, Kuhn, Maus [STOC'17]
- First **efficient deterministic $(2\Delta - 1)$-Edge-Coloring** using rounding for hypergraph matching
  F., Ghaffari, Kuhn [FOCS'17]

Open Problems

- Prove or disprove $\Omega(\log n)$ for Maximal Matching.
- **Conjecture** *Göös, Hirvonen, Suomela* [PODC'14]
  No $o(\Delta) + O(\log^* n)$ algorithm for Maximal Matching.

- Rounding in $O(\log \Delta)$ instead of $O(\log^2 \Delta)$?
- Devise a more general deterministic rounding method.

- **Linial's Question from the 1980s:**
  Is there an efficient deterministic algorithm for MIS?