Deterministic Distributed Edge-Coloring
via
Hypergraph Maximal Matching

<u>Manuela Fischer</u>
ETH Zurich

Mohsen Ghaffari
ETH Zurich

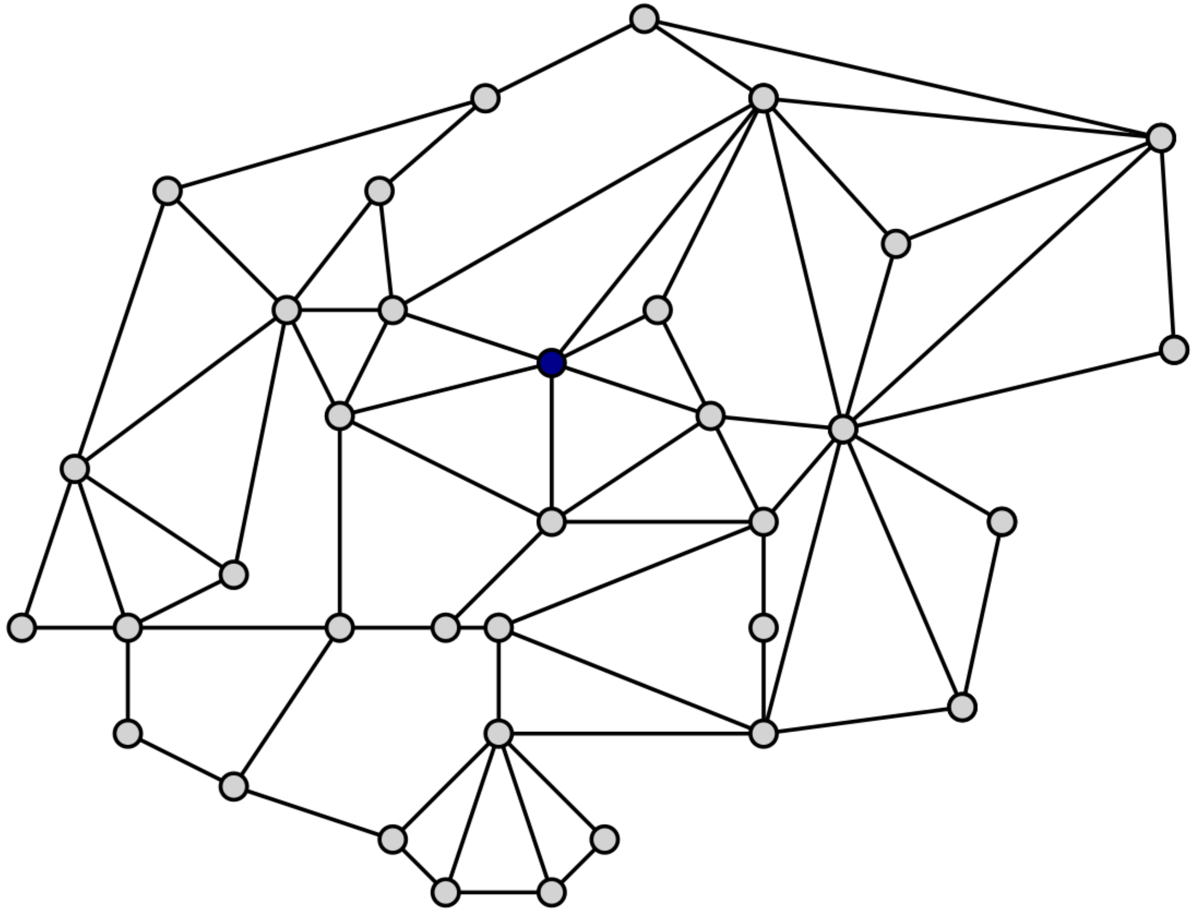
Fabian Kuhn
Uni Freiburg

LOCAL Model (Linial [FOCS'87])

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

- synchronous message-passing rounds
- unbounded message size and computation

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



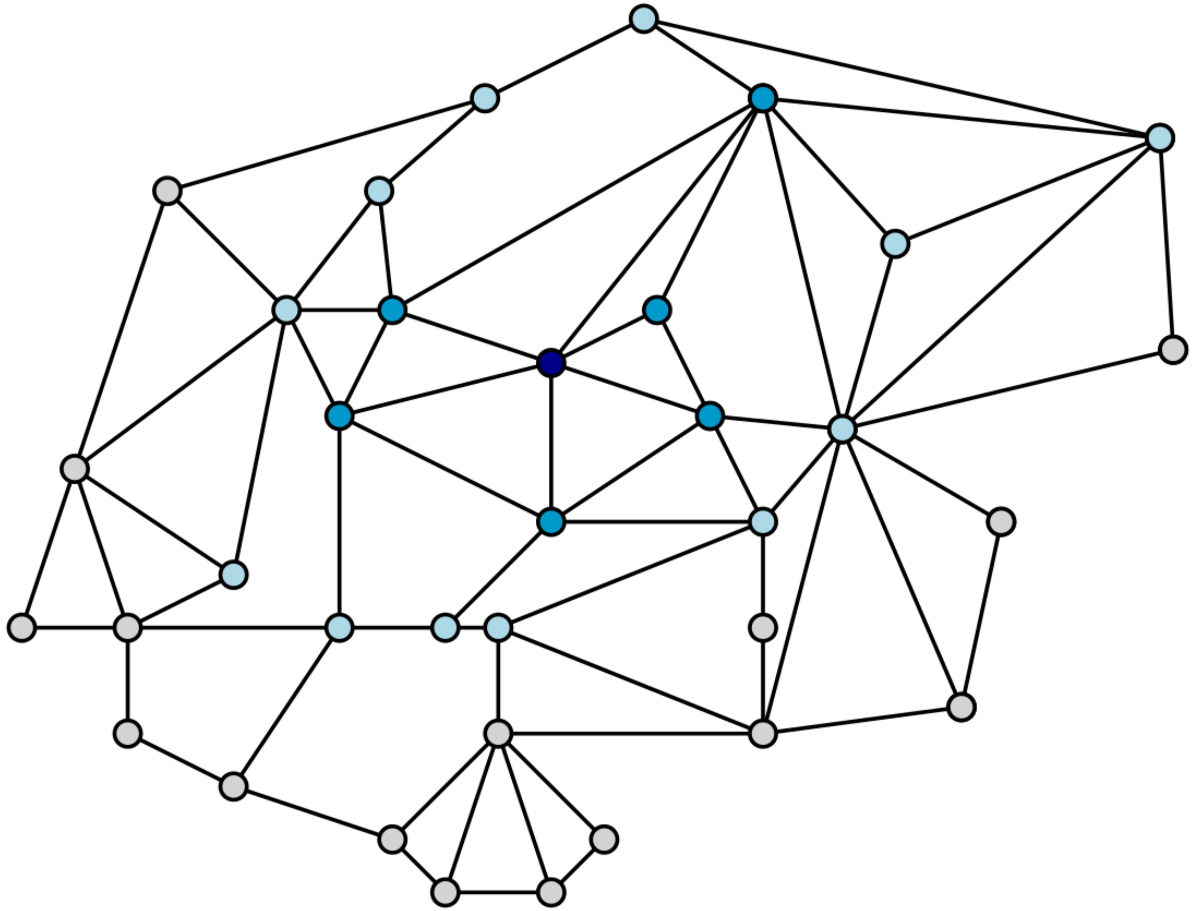
round complexity of a problem characterizes its locality

LOCAL Model (Linial [FOCS'87])

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

- synchronous message-passing rounds
- unbounded message size and computation

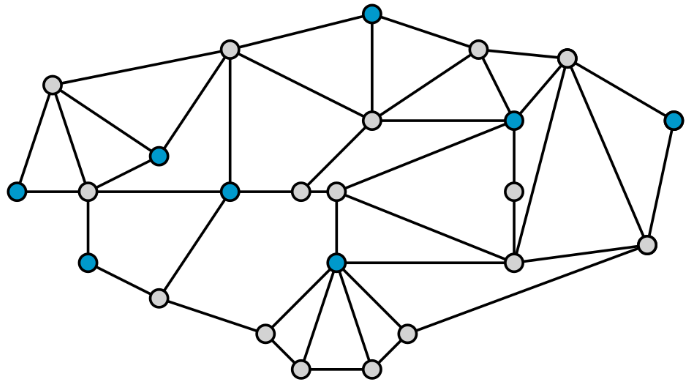
- **Round Complexity**:
  number of rounds to solve the problem



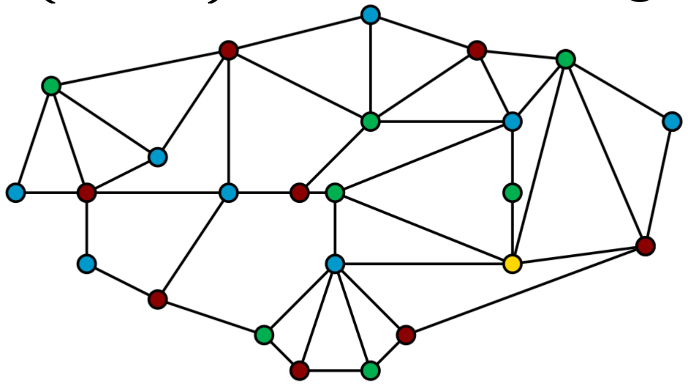
round complexity of a problem characterizes its locality

Classic LOCAL Graph Problems

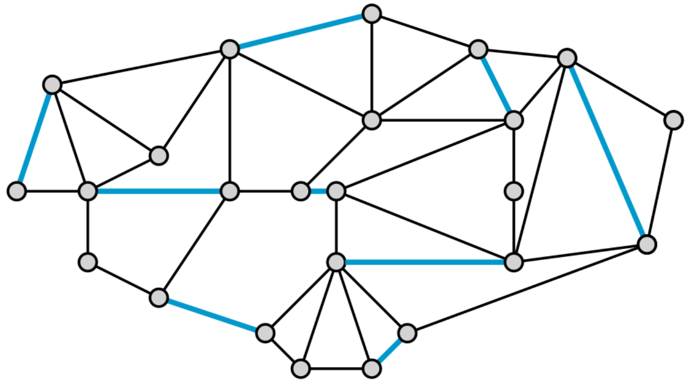
Maximal Independent Set



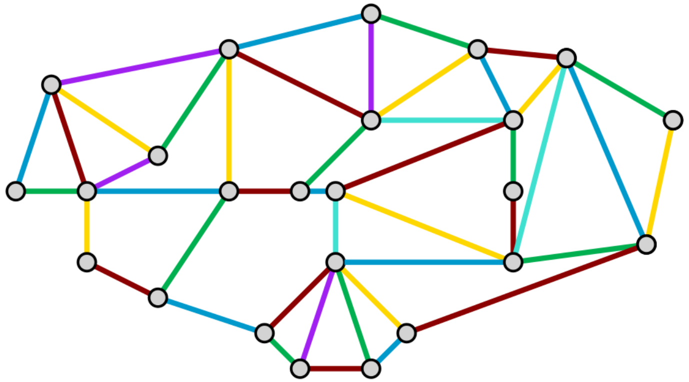
$(\Delta + 1)$-Vertex-Coloring



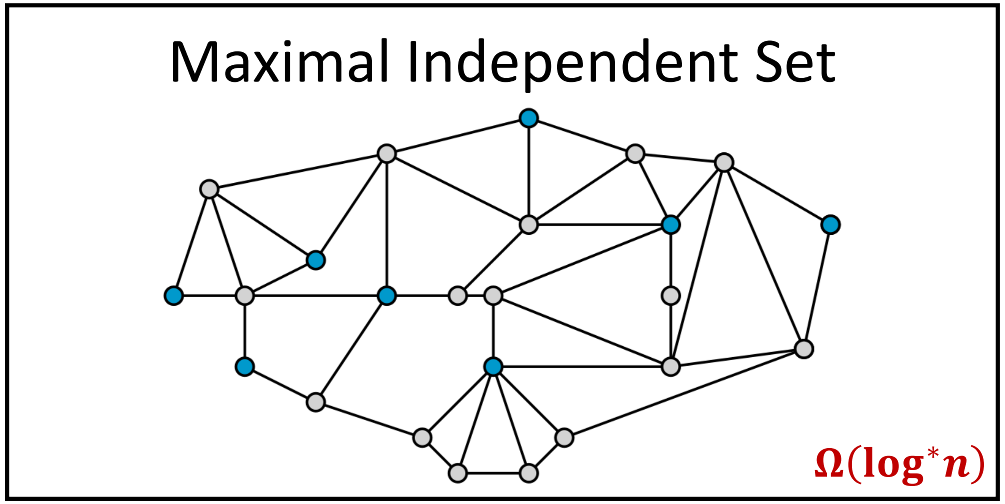
Maximal Matching

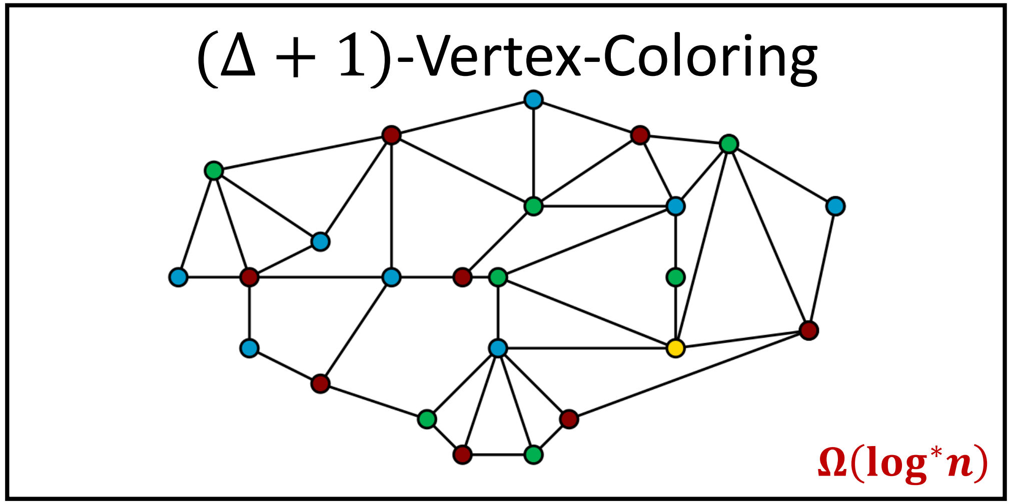


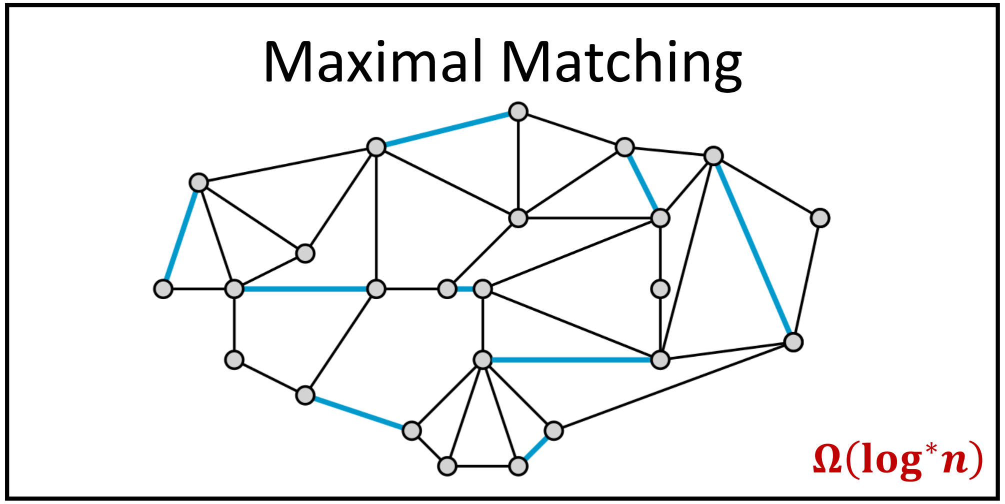
$(2\Delta - 1)$-Edge-Coloring

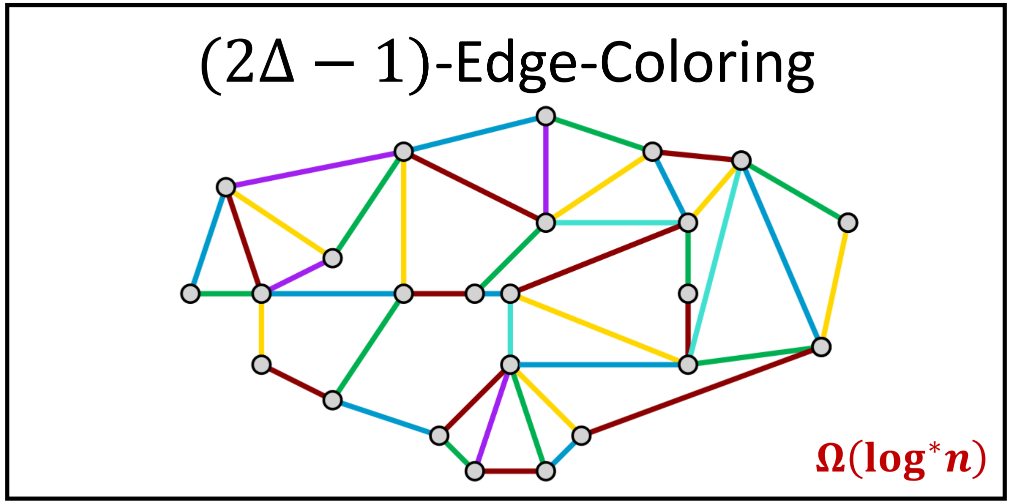


Classic LOCAL Graph Problems





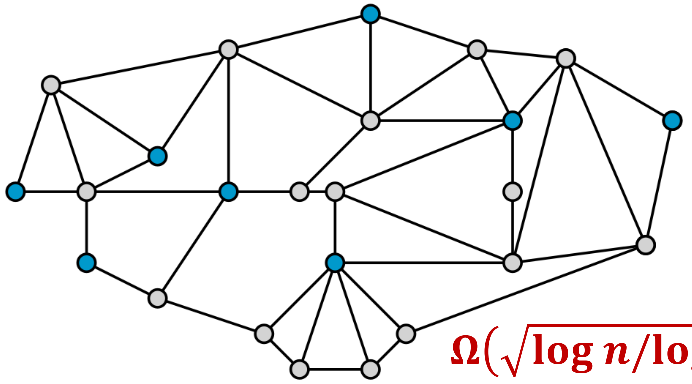




$\Omega(\log^* n)$ Linial [SICOMP'92]

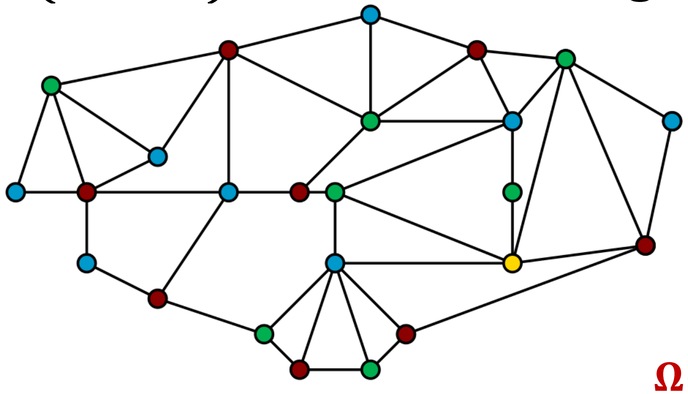
Classic LOCAL Graph Problems

Maximal Independent Set



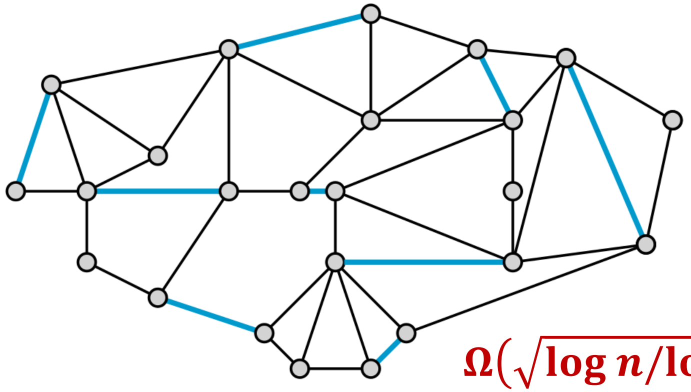
$\Omega\left(\sqrt{\log n / \log\log n}\right)$

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



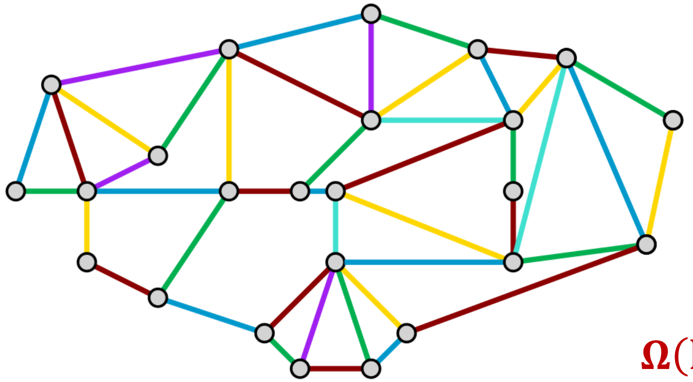
$\Omega(\log^* n)$

Maximal Matching



$\Omega\left(\sqrt{\log n / \log\log n}\right)$

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



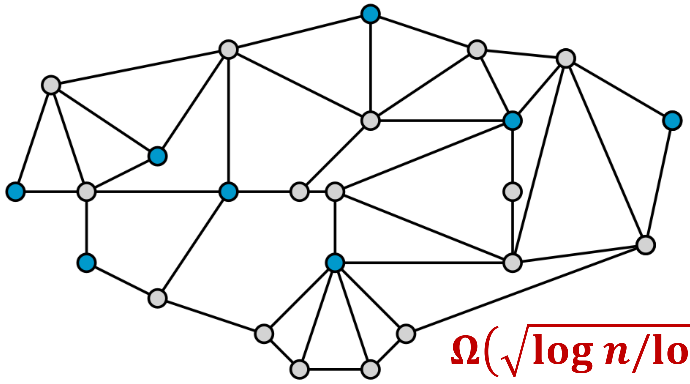
$\Omega(\log^* n)$

$\Omega(\log^* n)$ — **Linial [SICOMP'92]**

$\Omega\left(\sqrt{\log n / \log\log n}\right)$ — **Kuhn, Moscibroda, Wattenhofer [PODC'06]**

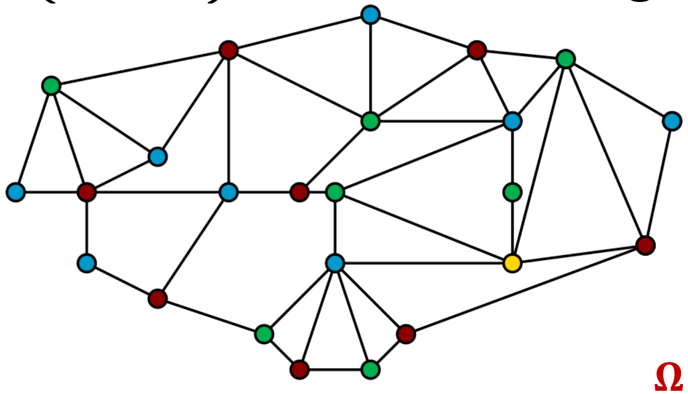
Classic LOCAL Graph Problems

Maximal Independent Set



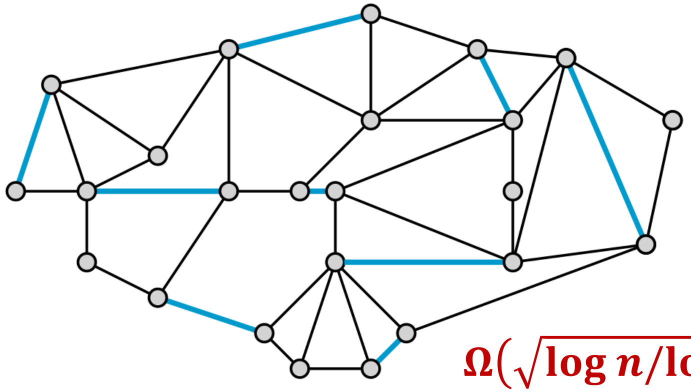
$\Omega\left(\sqrt{\log n/\log\log n}\right)$

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



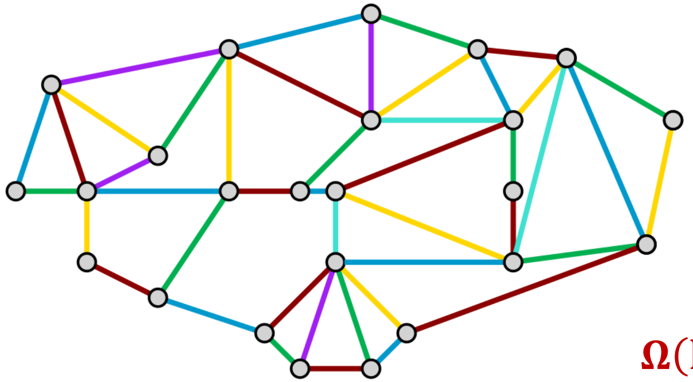
$\Omega(\log^* n)$

Maximal Matching



$\Omega\left(\sqrt{\log n/\log\log n}\right)$

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



$\Omega(\log^* n)$

$\Omega(\log^* n)$	Linial [SICOMP'92]
$\Omega\left(\sqrt{\log n/\log\log n}\right)$	Kuhn, Moscibroda, Wattenhofer [PODC'06]
randomized $O(\log n)$	Luby [STOC'85], Alon, Babai, Itai [JALG'86]

Classic LOCAL Graph Problems

Maximal Independent Set

$2^{O(\sqrt{\log n})}$ Panconesi, Srinivasan [STOC'92]

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

$2^{O(\sqrt{\log n})}$ Panconesi, Srinivasan [STOC'92]

Maximal Matching

$2^{O(\sqrt{\log n})}$ Panconesi, Srinivasan [STOC'92]

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

$2^{O(\sqrt{\log n})}$ Panconesi, Srinivasan [STOC'92]

Classic LOCAL Graph Problems

Maximal Independent Set

$2^{O(\sqrt{\log n})}$ Panconesi, Srinivasan [STOC’92]

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

$2^{O(\sqrt{\log n})}$ Panconesi, Srinivasan [STOC’92]

Maximal Matching

$2^{O(\sqrt{\log n})}$ Panconesi, Srinivasan [STOC’92]

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

$2^{O(\sqrt{\log n})}$ Panconesi, Srinivasan [STOC’92]

Linial’s Question from the 1980s:

Is there an efficient ($\mathrm{poly}\log n$ rounds) deterministic algorithm for Maximal Independent Set?

Classic LOCAL Graph Problems

Maximal Independent Set

$2^{O(\sqrt{\log n})}$	Panconesi, Srinivasan [STOC'92]

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

$2^{O(\sqrt{\log n})}$	Panconesi, Srinivasan [STOC'92]

Maximal Matching

$2^{O(\sqrt{\log n})}$	Panconesi, Srinivasan [STOC'92]
$O(\log^7 n)$	Hańćkowiak, Karoński, Panconesi [SODA'98]
$O(\log^4 n)$	Hańćkowiak, Karoński, Panconesi [PODC'99]
$O(\log^3 n)$	F. [DISC'17]

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

$2^{O(\sqrt{\log n})}$	Panconesi, Srinivasan [STOC'92]

Linial's Question from the 1980s:

Is there an efficient (poly $\log n$ rounds) deterministic algorithm for Maximal Independent Set?

Classic LOCAL Graph Problems

"While maximal matchings can be computed in polylogarithmic time […], it is a decade old open problem whether the same running time is achievable for the remaining three structures."

Panconesi, Rizzi '01

Linial's Question from the 1980s:

Is there an efficient ($\mathrm{poly} \log n$ rounds) deterministic algorithm for Maximal Independent Set?

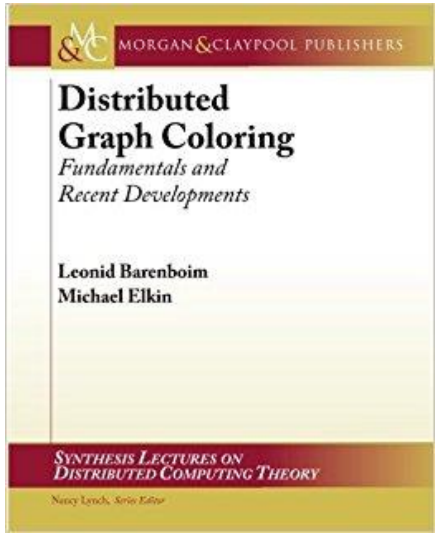
Classic LOCAL Graph Problems

"While maximal matchings can be computed in polylogarithmic time [...], it is a decade old open problem whether the same running time is achievable for the remaining three structures."

Panconesi, Rizzi '01

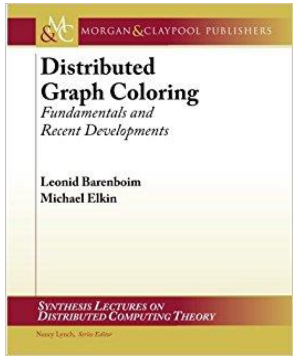
Open Problem 11.4:
Devise or rule out a deterministic $(2\Delta - 1)$-edge-coloring algorithm that runs in polylogarithmic time.

Barenboim, Elkin '13



Linial's Question from the 1980s:
Is there an efficient ($\mathrm{poly} \log n$ rounds) deterministic algorithm for Maximal Independent Set?

Overview of Results & Implications

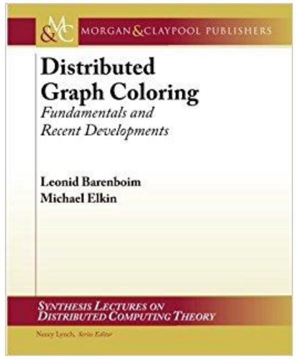


$(2\Delta - 1)$**-Edge-Coloring**
in $O(\log^8 n)$ rounds

Randomized
$(2\Delta - 1)$**-Edge-Coloring**
in $O(\log^8 \log n)$rounds

Open Problem 11.4

Overview of Results & Implications



$(2\Delta - 1)$-Edge-Coloring
in $O(\log^8 n)$ rounds

Randomized
$(2\Delta - 1)$-Edge-Coloring
in $O(\log^8 \log n)$rounds

Open Problem 11.4

Maximal Independent Set and $(\Delta + 1)$-Vertex-Coloring
for graphs with bounded neighborhood independence

Open Problem 11.5

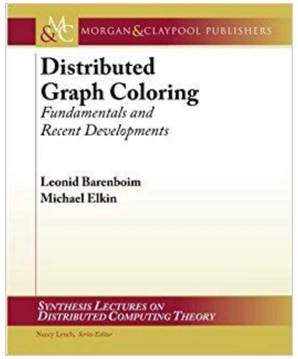
Low-Out-Degree Orientation

almost Open Problem 11.10

$(1 + \varepsilon)$-Approximation of Matching

Overview of Results & Implications

Rank-r-Hypergraph Maximal Matching
in $\mathrm{poly}\, r \cdot \log^{O(\log r)} \Delta \cdot \log n$ rounds



$\mathbf{(2\Delta - 1)}$**-Edge-Coloring**
in $O(\log^8 n)$ rounds

Randomized
$\mathbf{(2\Delta - 1)}$**-Edge-Coloring**
in $O(\log^8 \log n)$rounds

Open Problem 11.4

Maximal Independent Set and $\mathbf{(\Delta + 1)}$**-Vertex-Coloring**
for graphs with bounded neighborhood independence

Open Problem 11.5

Low-Out-Degree Orientation

almost Open Problem 11.10

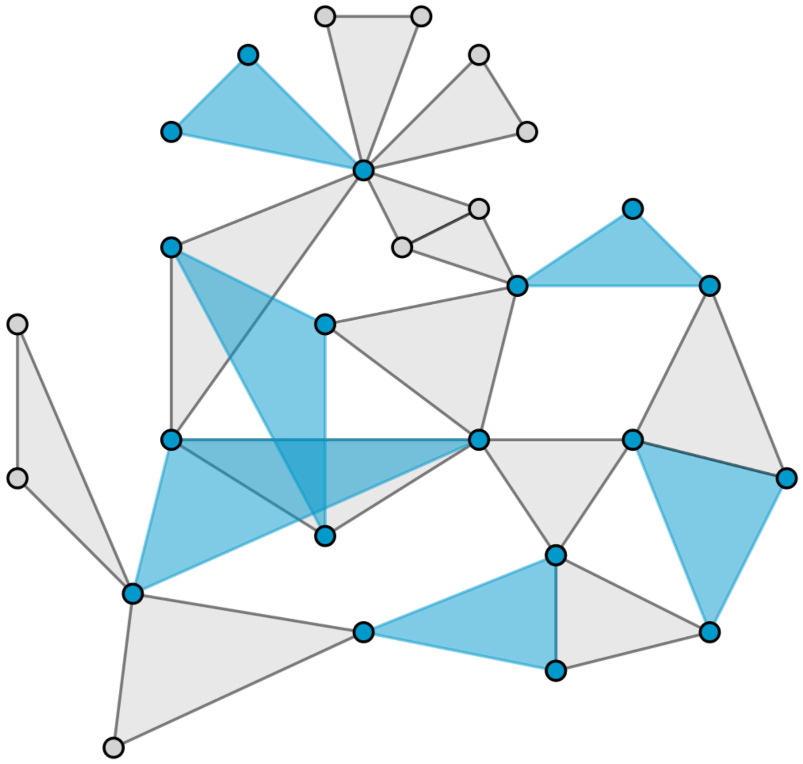
$\mathbf{(1 + \varepsilon)}$**-Approximation of Matching**

I) Formulation as Hypergraph Maximal Matching

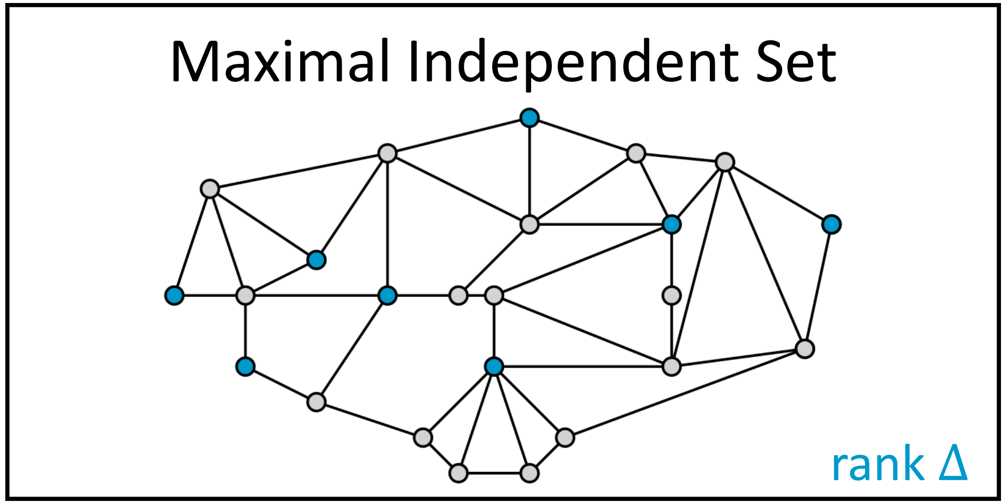
I) Formulation as Hypergraph Maximal Matching

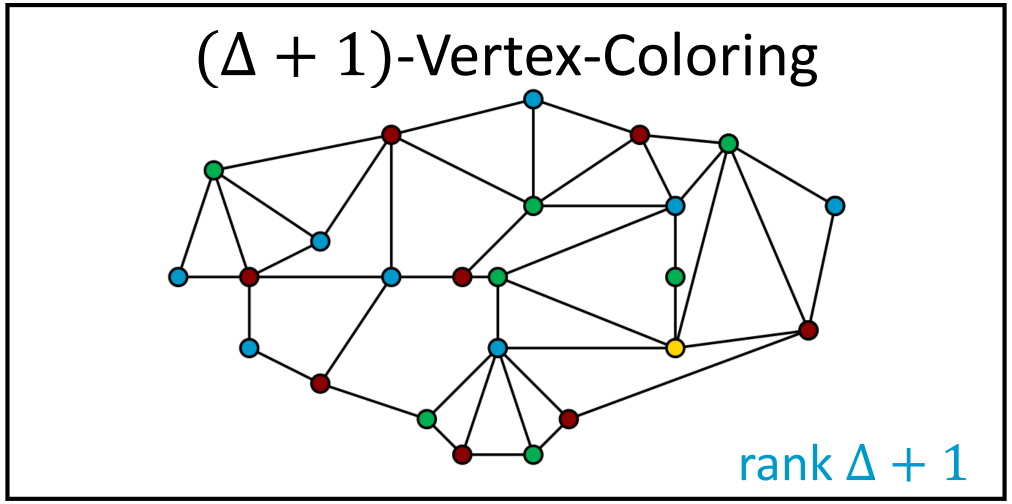
II) Hypergraph Maximal Matching Algorithm

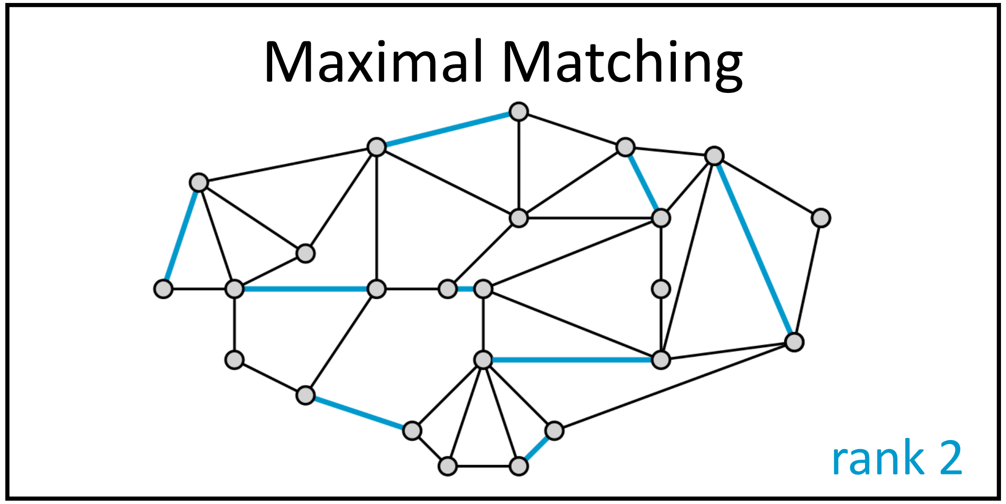
I) Formulation as Hypergraph Maximal Matching

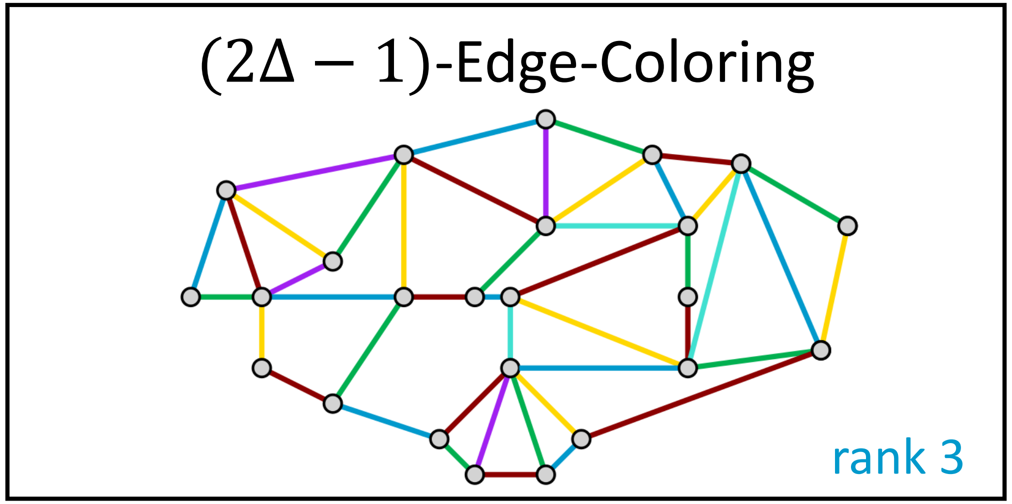


Unified Formulation as Hypergraph Maximal Matching Problem



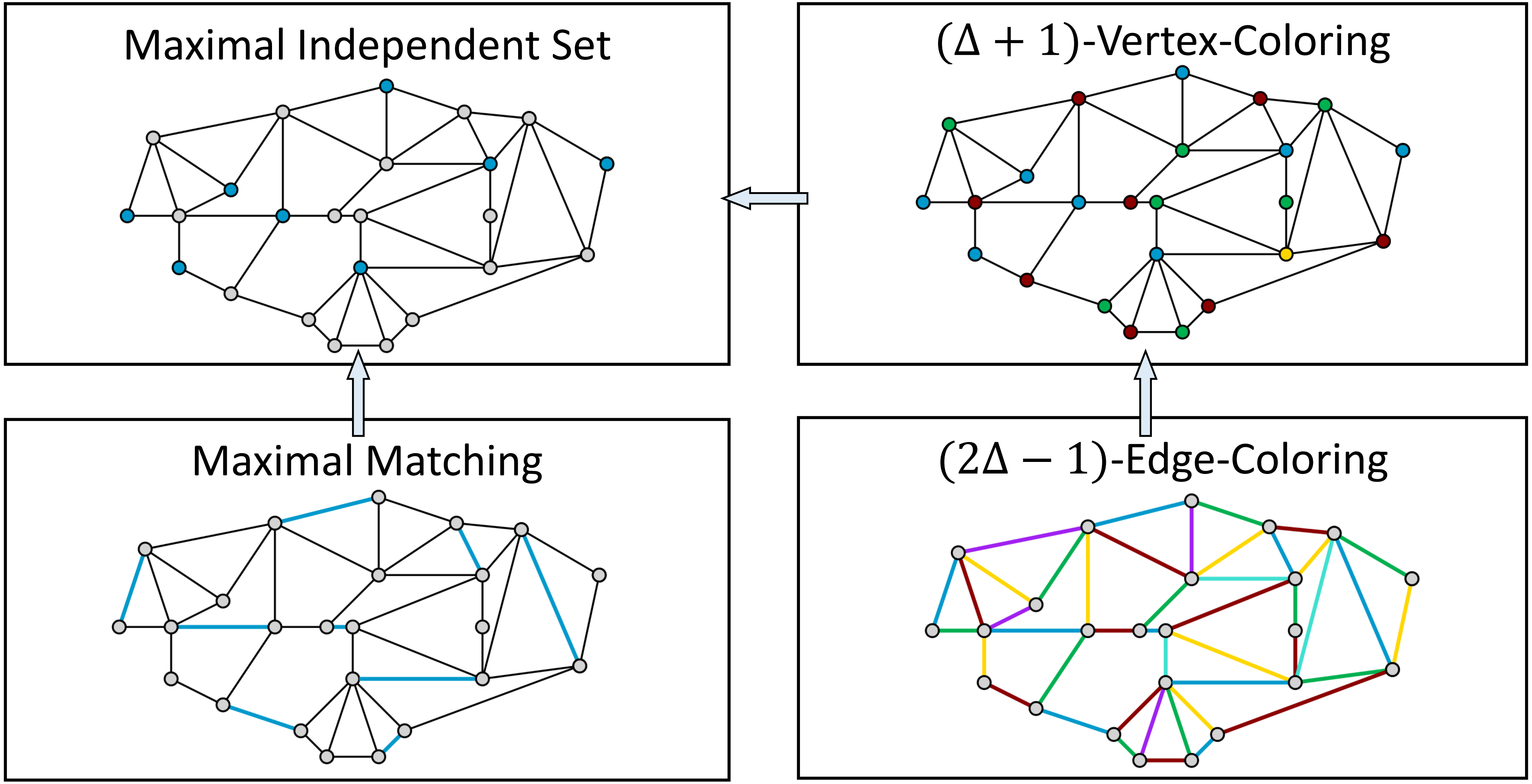






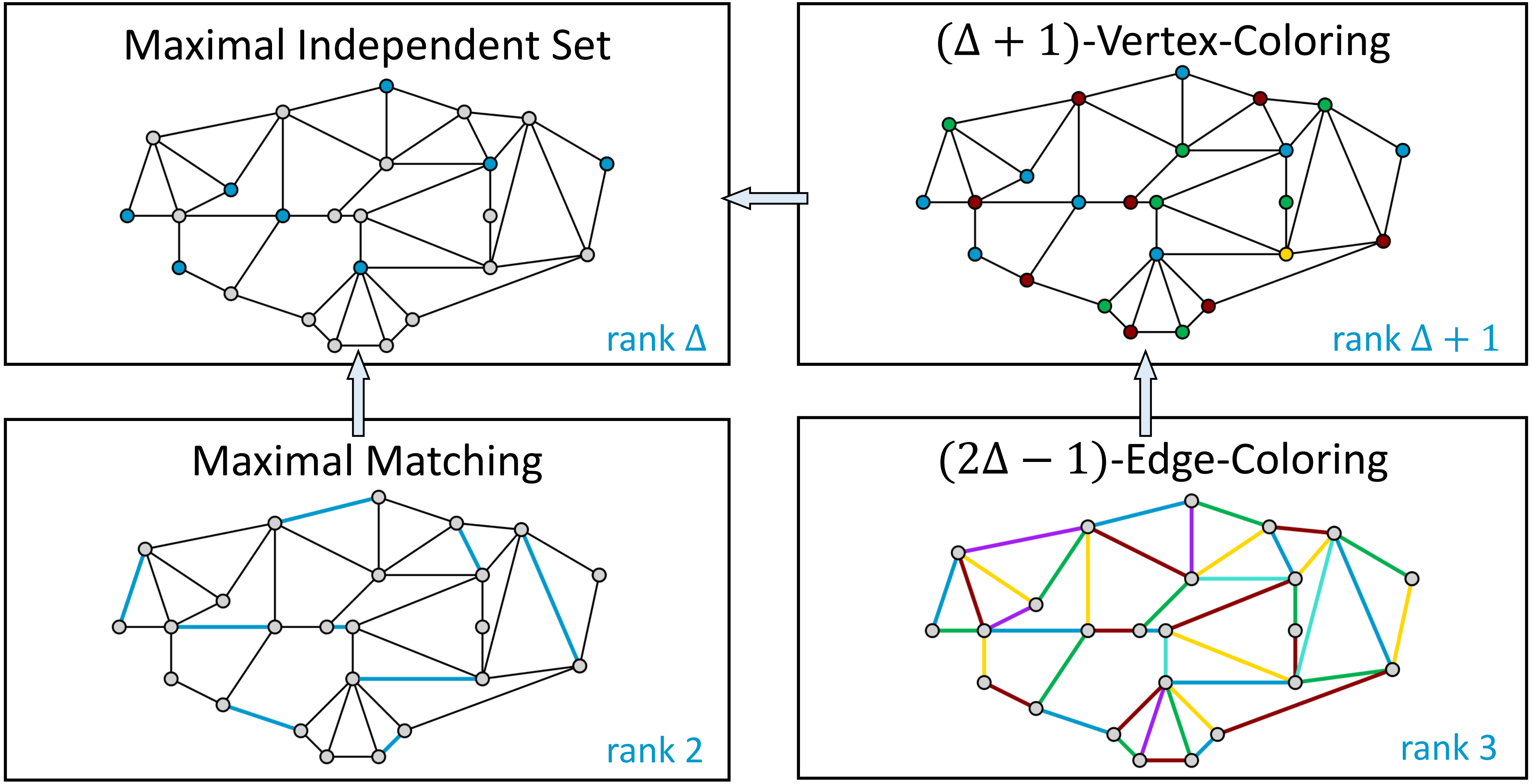
cast classic LOCAL graph problems as hypergraph maximal matching problems (LOCAL reductions)

Unified Formulation as Hypergraph Maximal Matching Problem



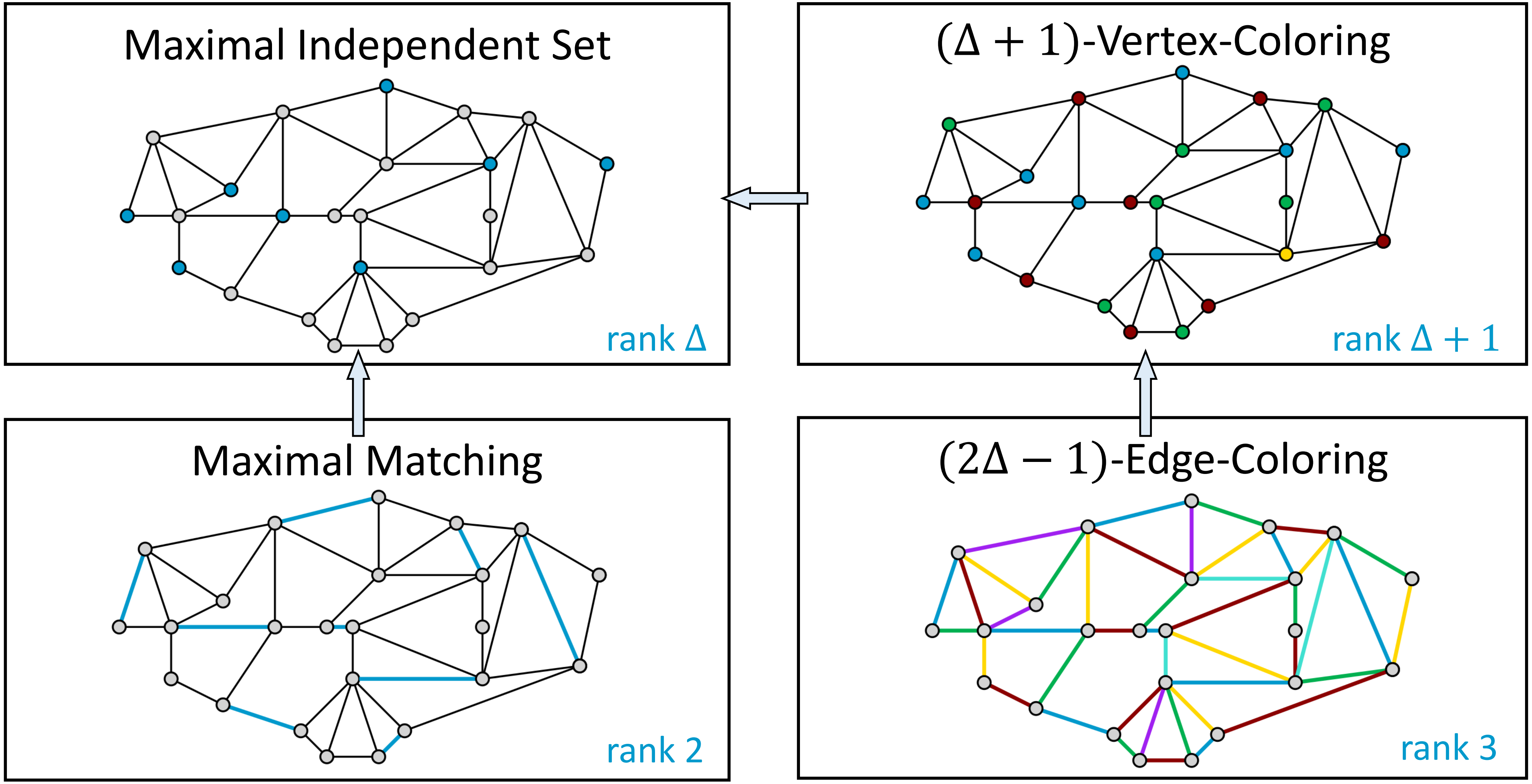
cast classic LOCAL graph problems as hypergraph maximal matching problems (LOCAL reductions)

Unified Formulation as Hypergraph Maximal Matching Problem



cast classic LOCAL graph problems as hypergraph maximal matching problems (LOCAL reductions)

Unified Formulation as Hypergraph Maximal Matching Problem

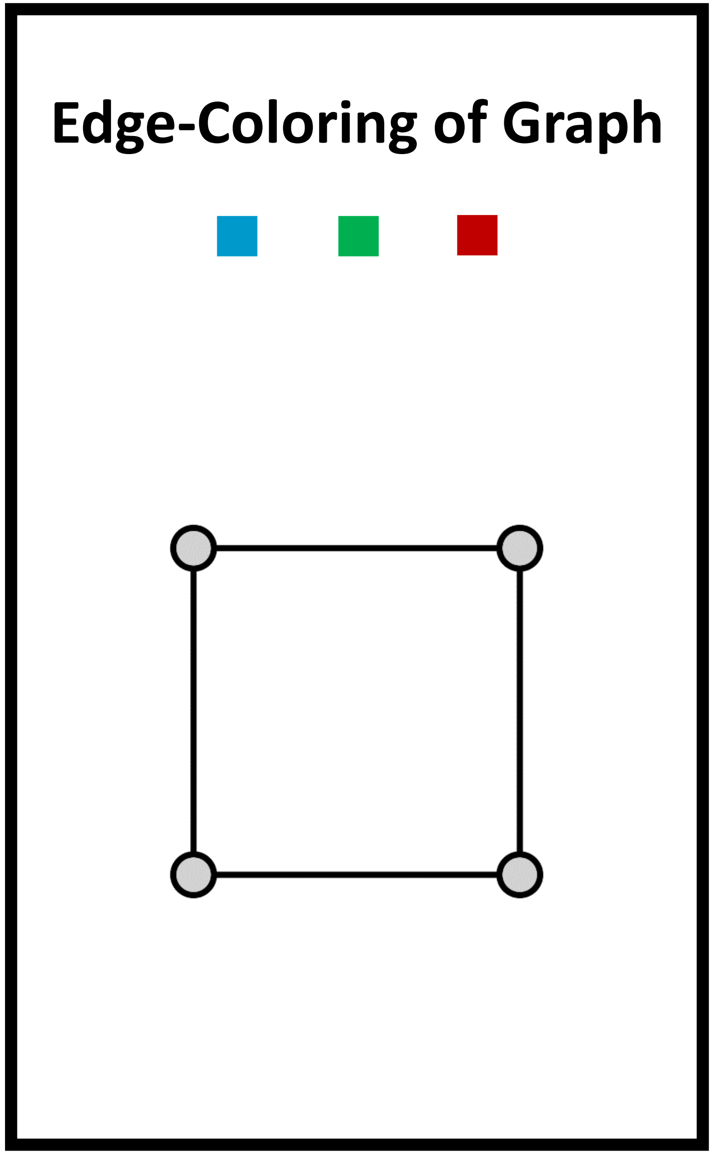


cast classic LOCAL graph problems as hypergraph maximal matching problems (LOCAL reductions)

smooth interpolation between maximal matching and maximal independent set

$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

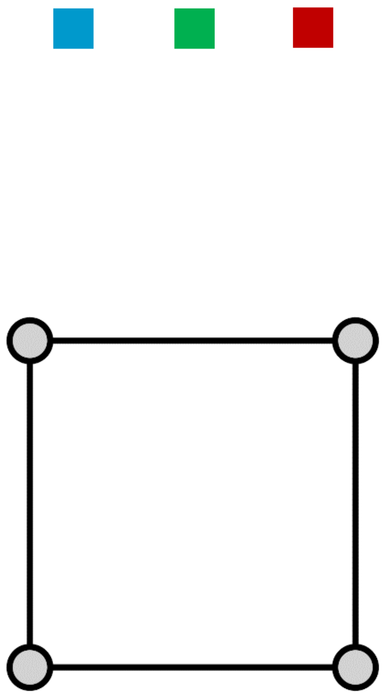
$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching



$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

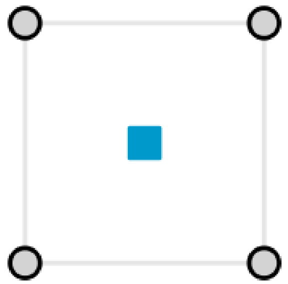
Rank-3-Hypergraph Maximal Matching

Edge-Coloring of Graph

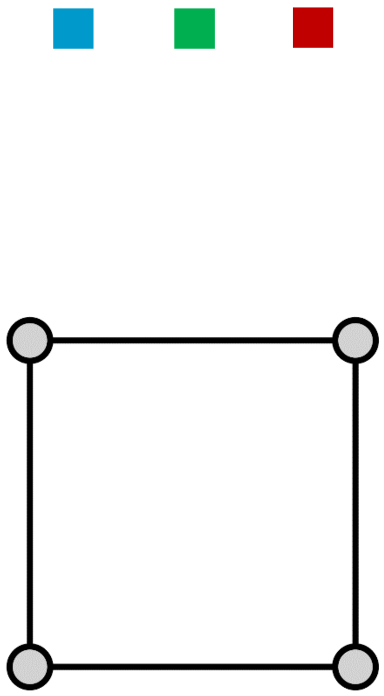


$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

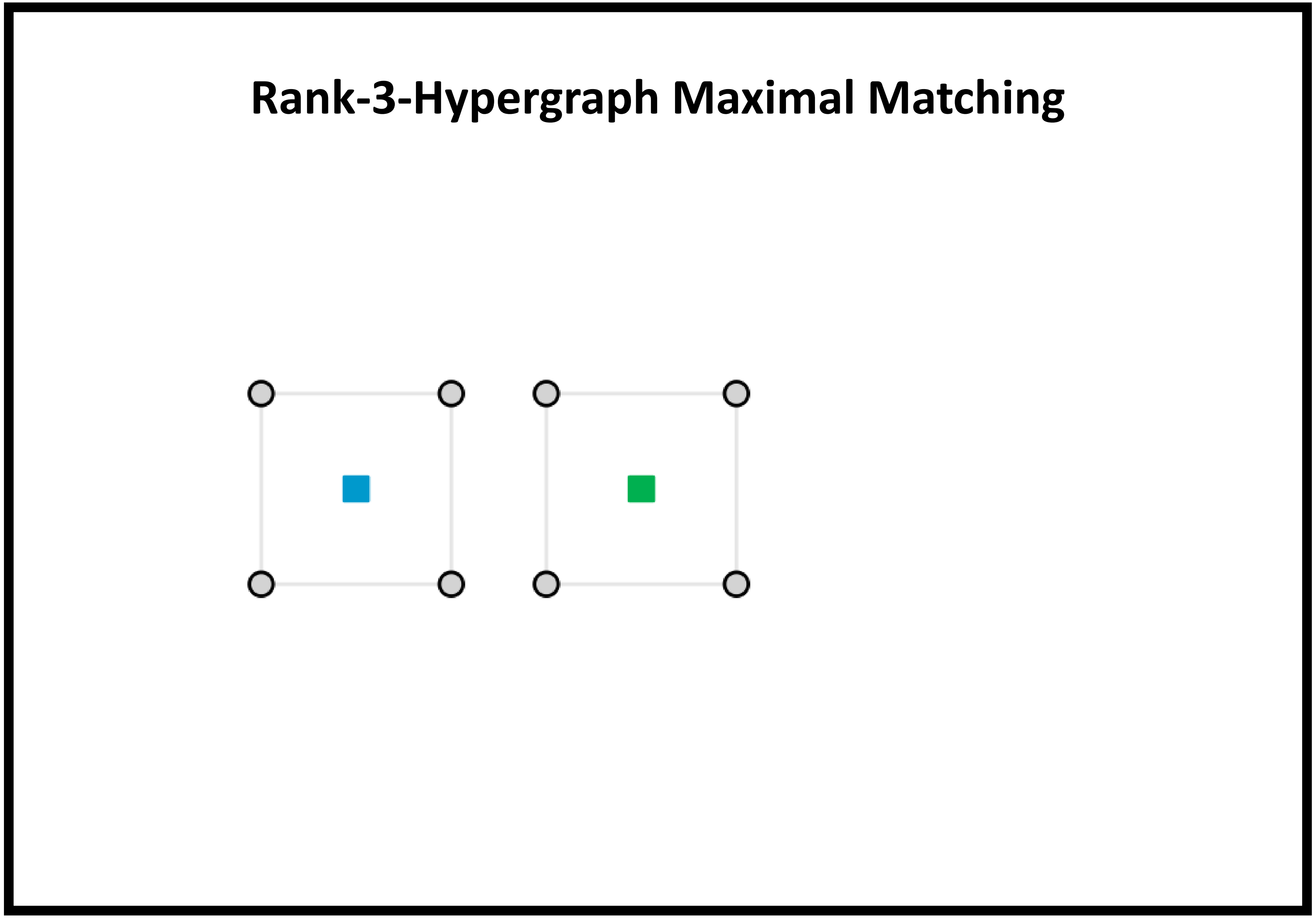
Rank-3-Hypergraph Maximal Matching

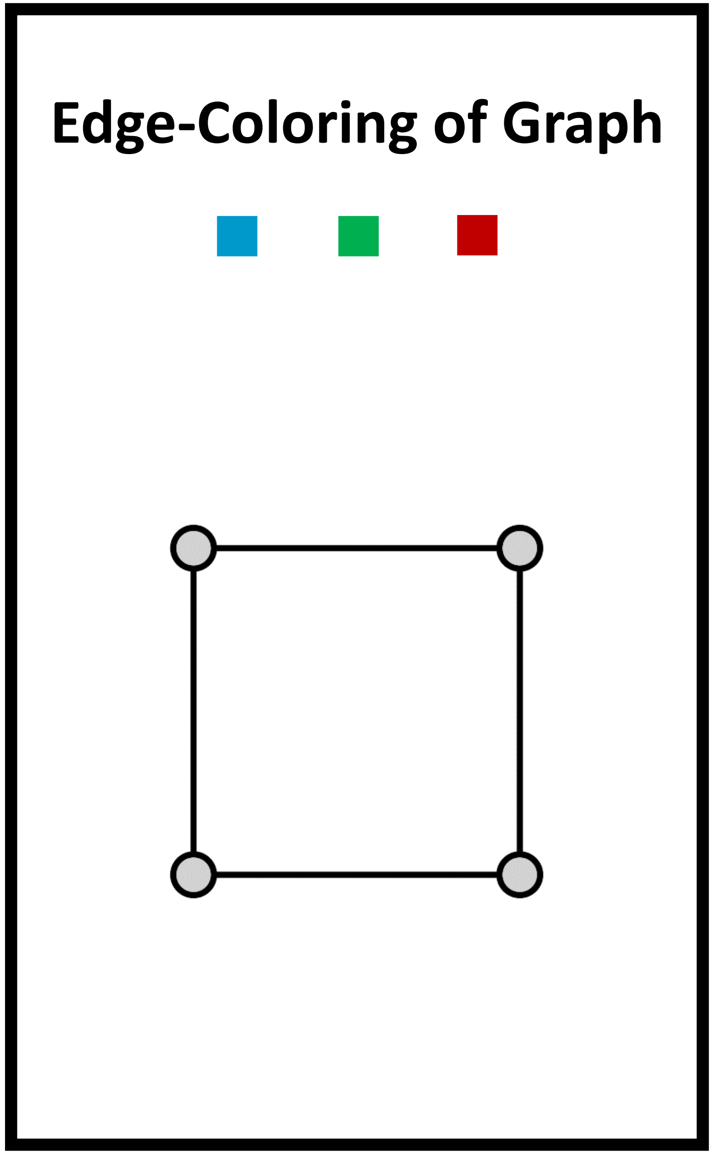


Edge-Coloring of Graph



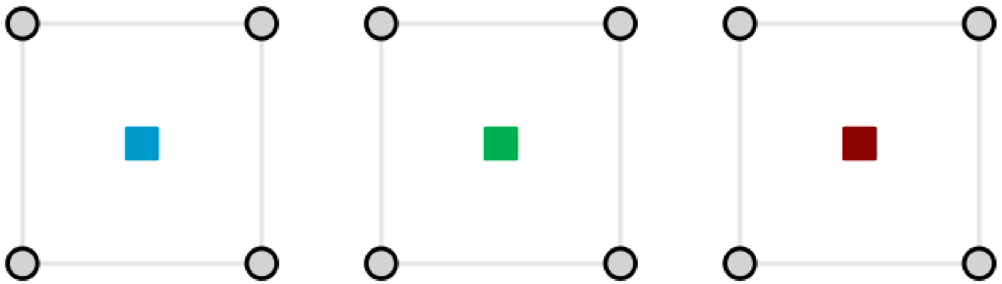
$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching



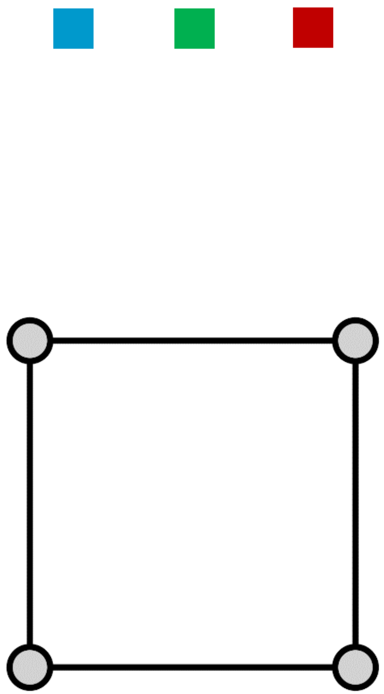


$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

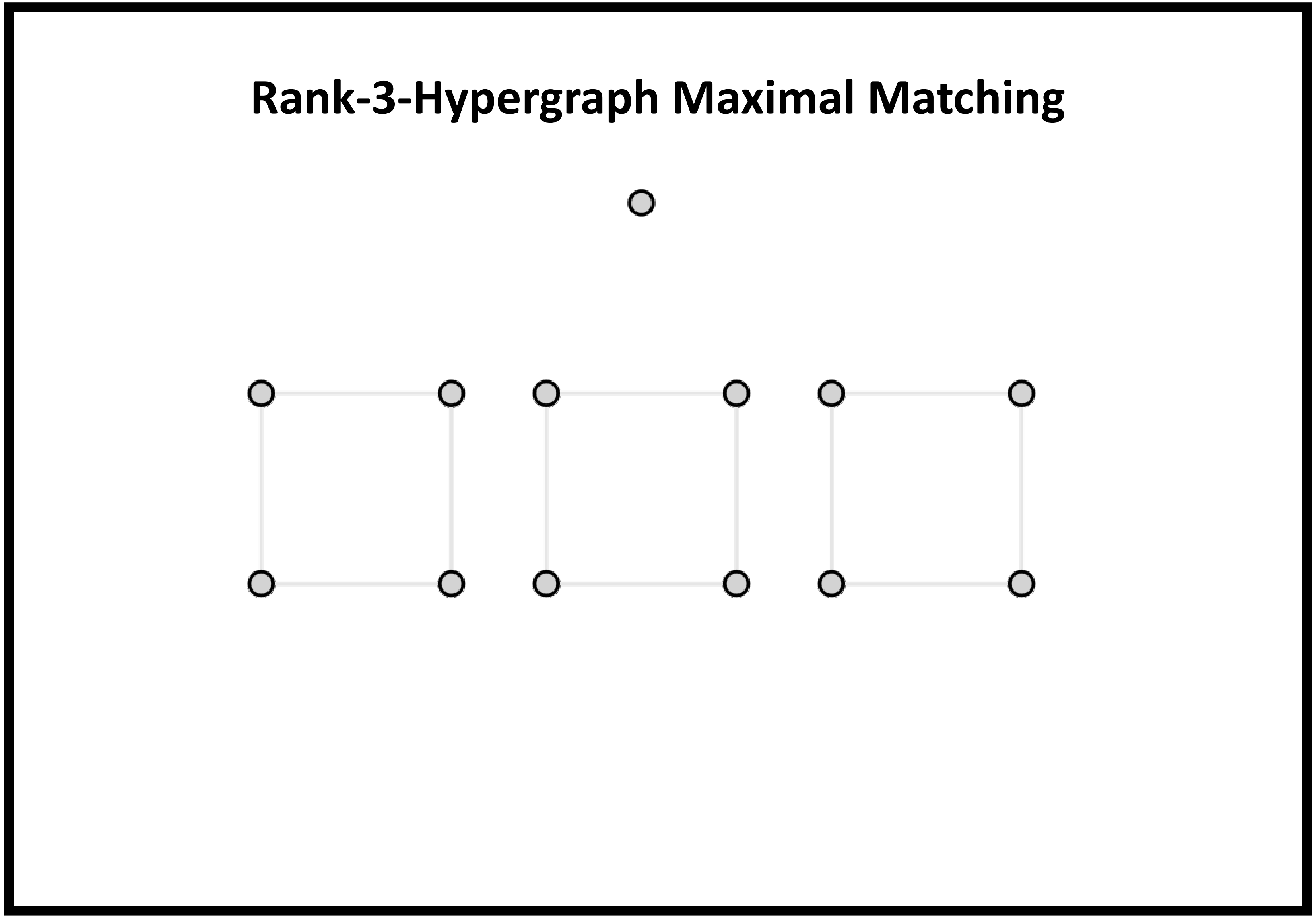
Rank-3-Hypergraph Maximal Matching

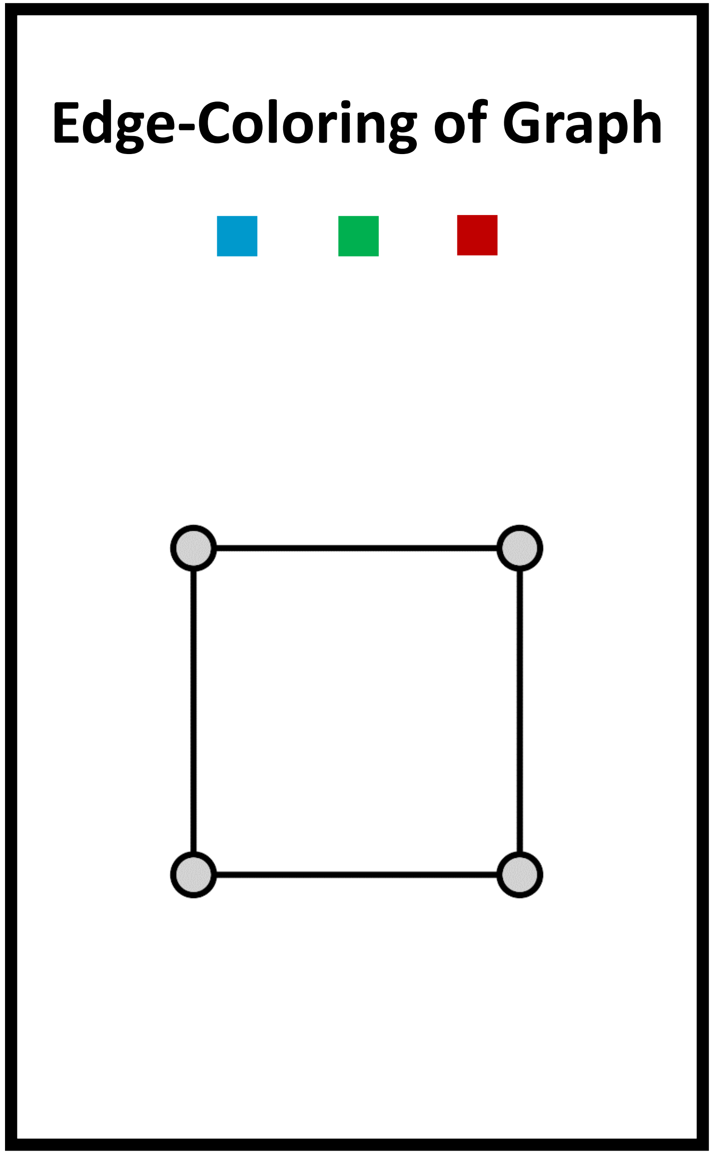


Edge-Coloring of Graph



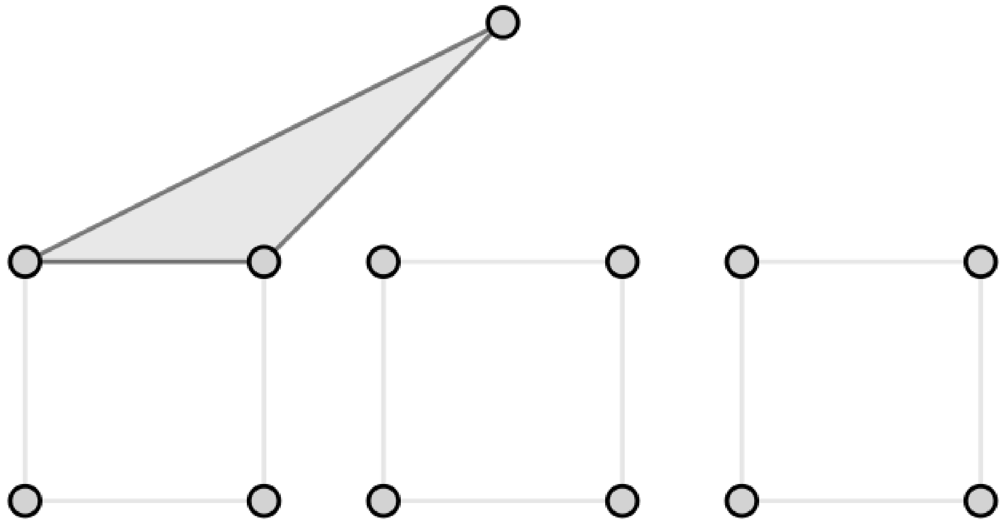
$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching



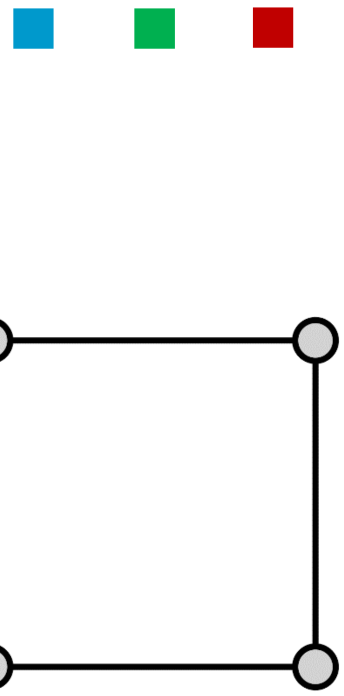


$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

Rank-3-Hypergraph Maximal Matching

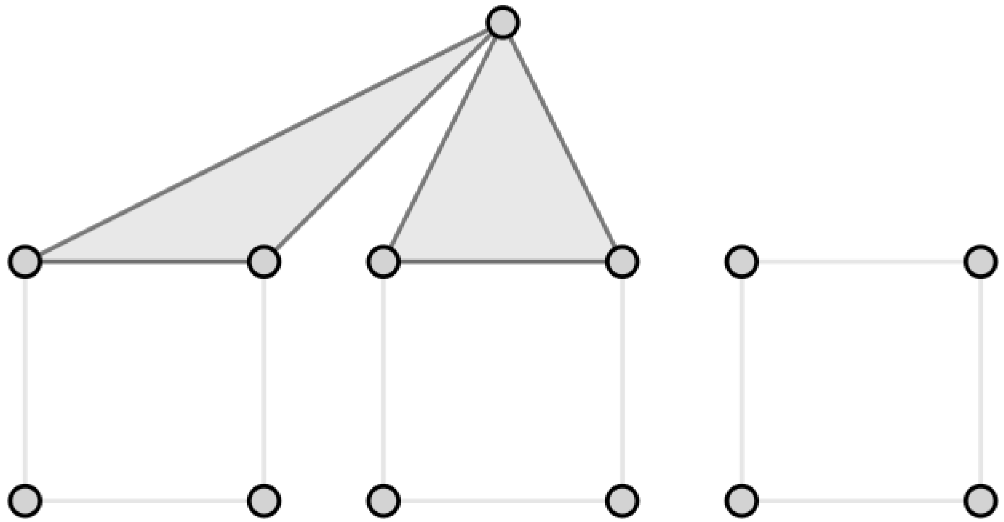


Edge-Coloring of Graph

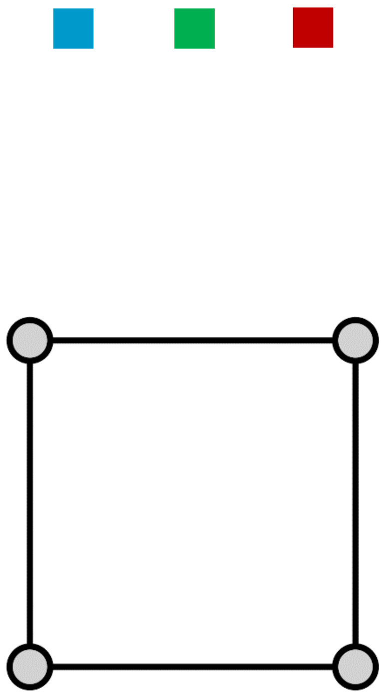


$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

Rank-3-Hypergraph Maximal Matching

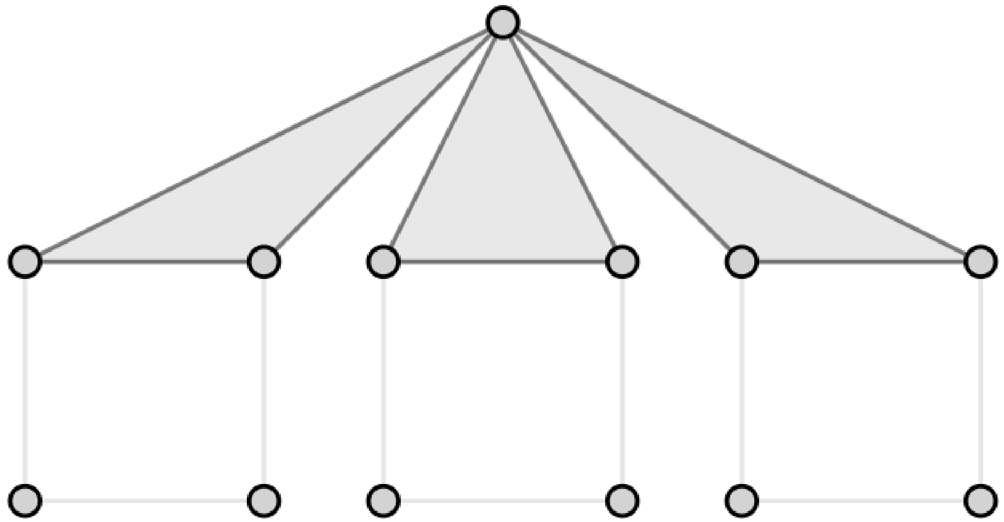


Edge-Coloring of Graph

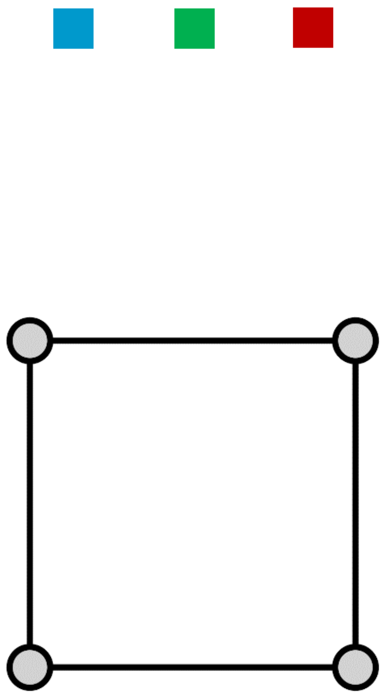


$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

Rank-3-Hypergraph Maximal Matching

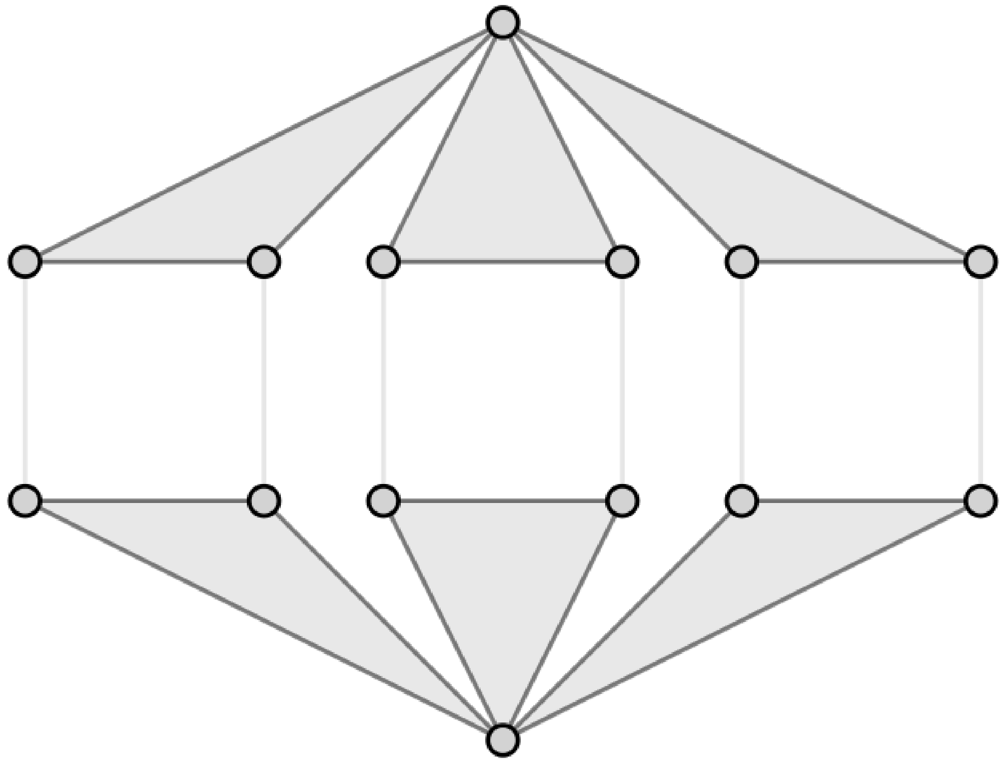


Edge-Coloring of Graph

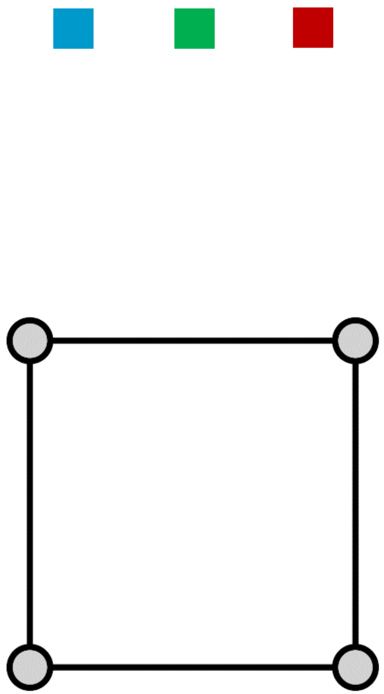


$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

Rank-3-Hypergraph Maximal Matching

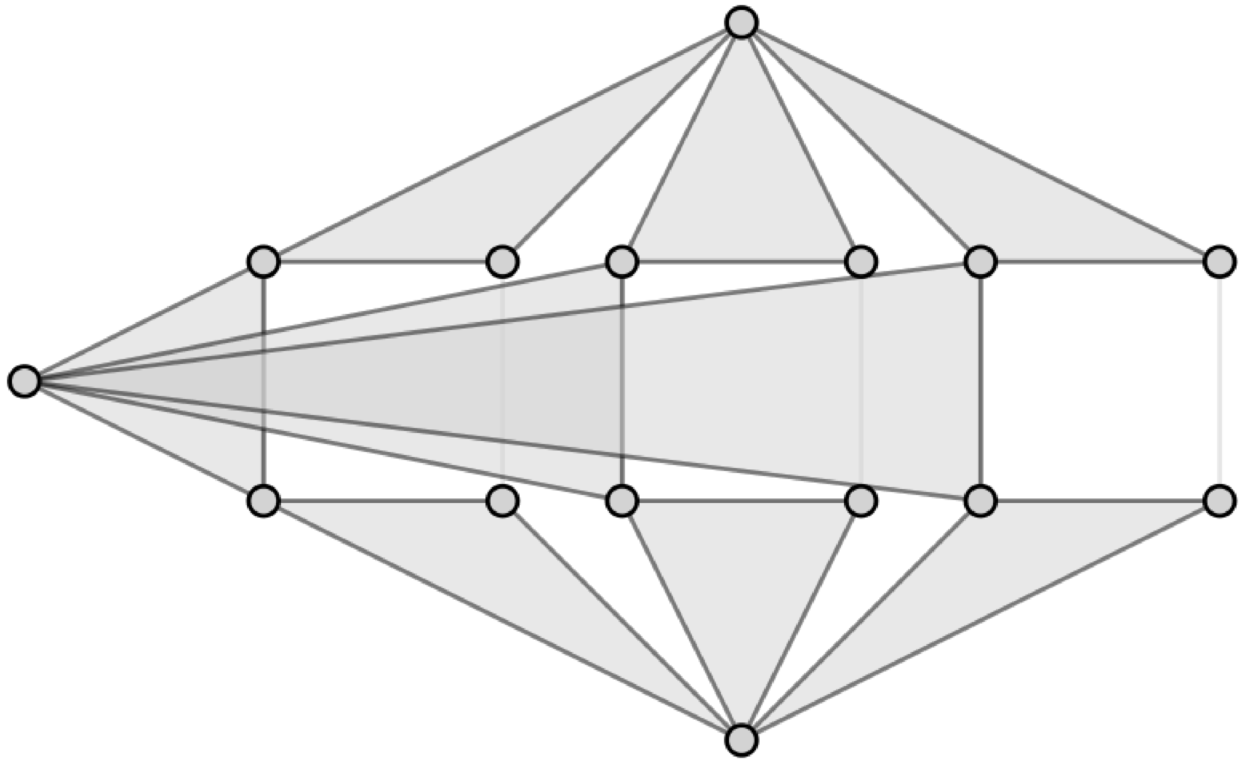


Edge-Coloring of Graph

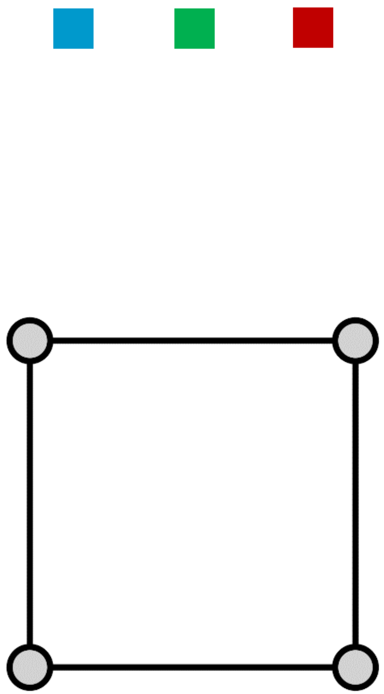


$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

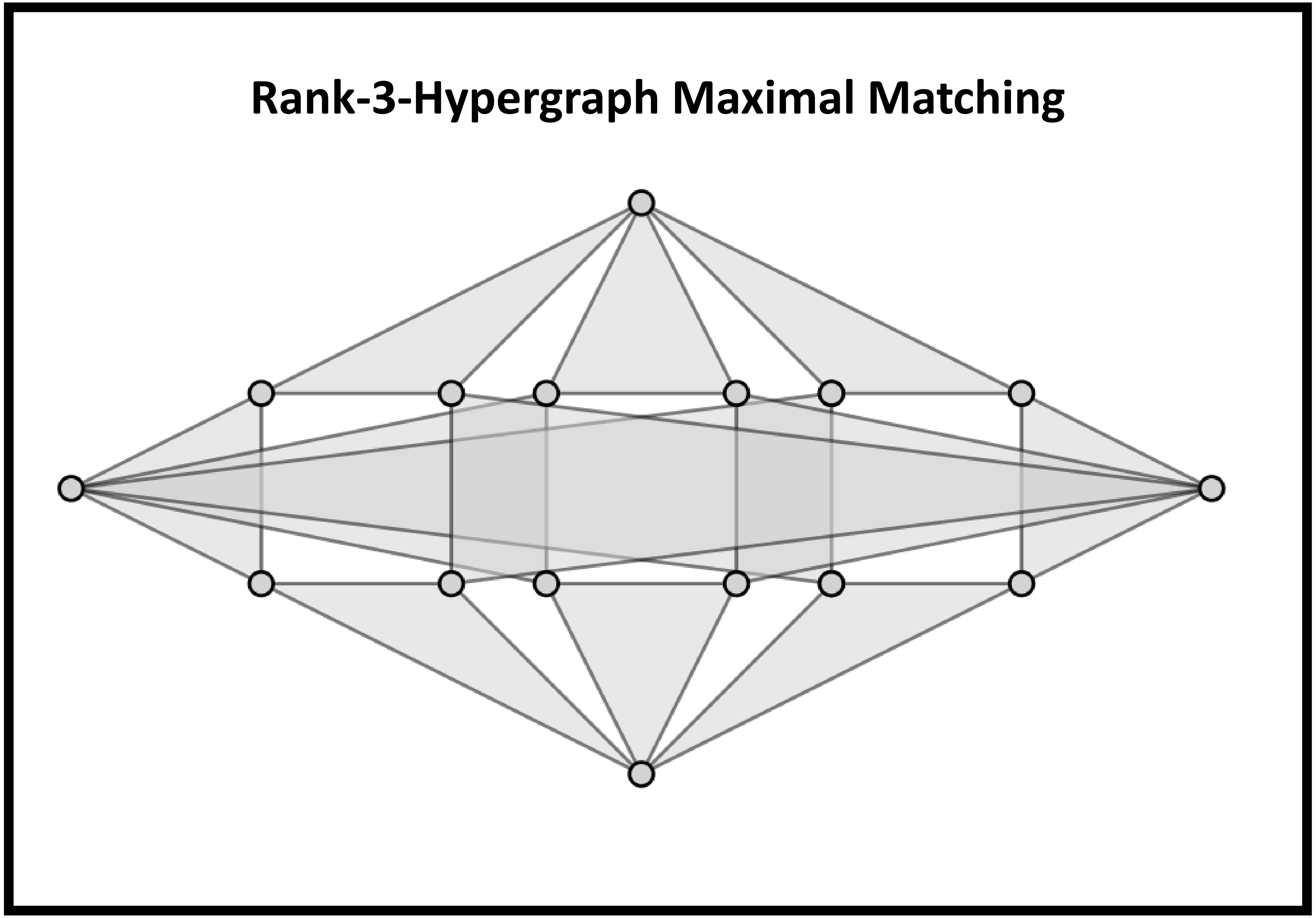
Rank-3-Hypergraph Maximal Matching

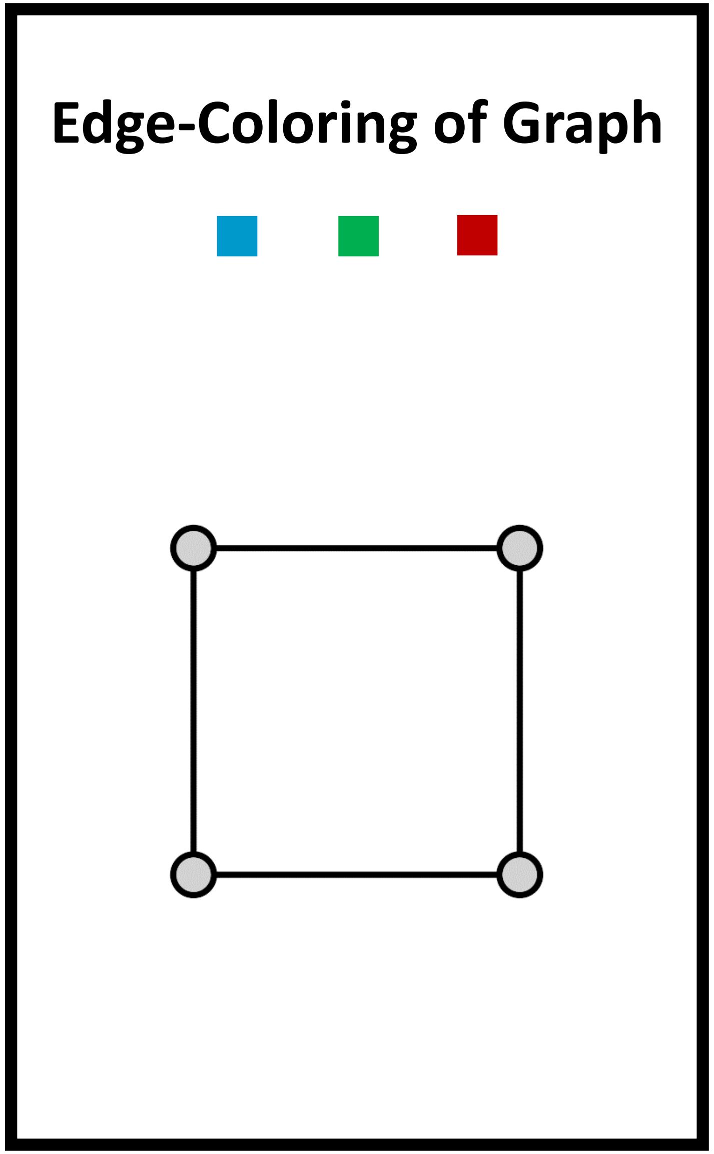


Edge-Coloring of Graph

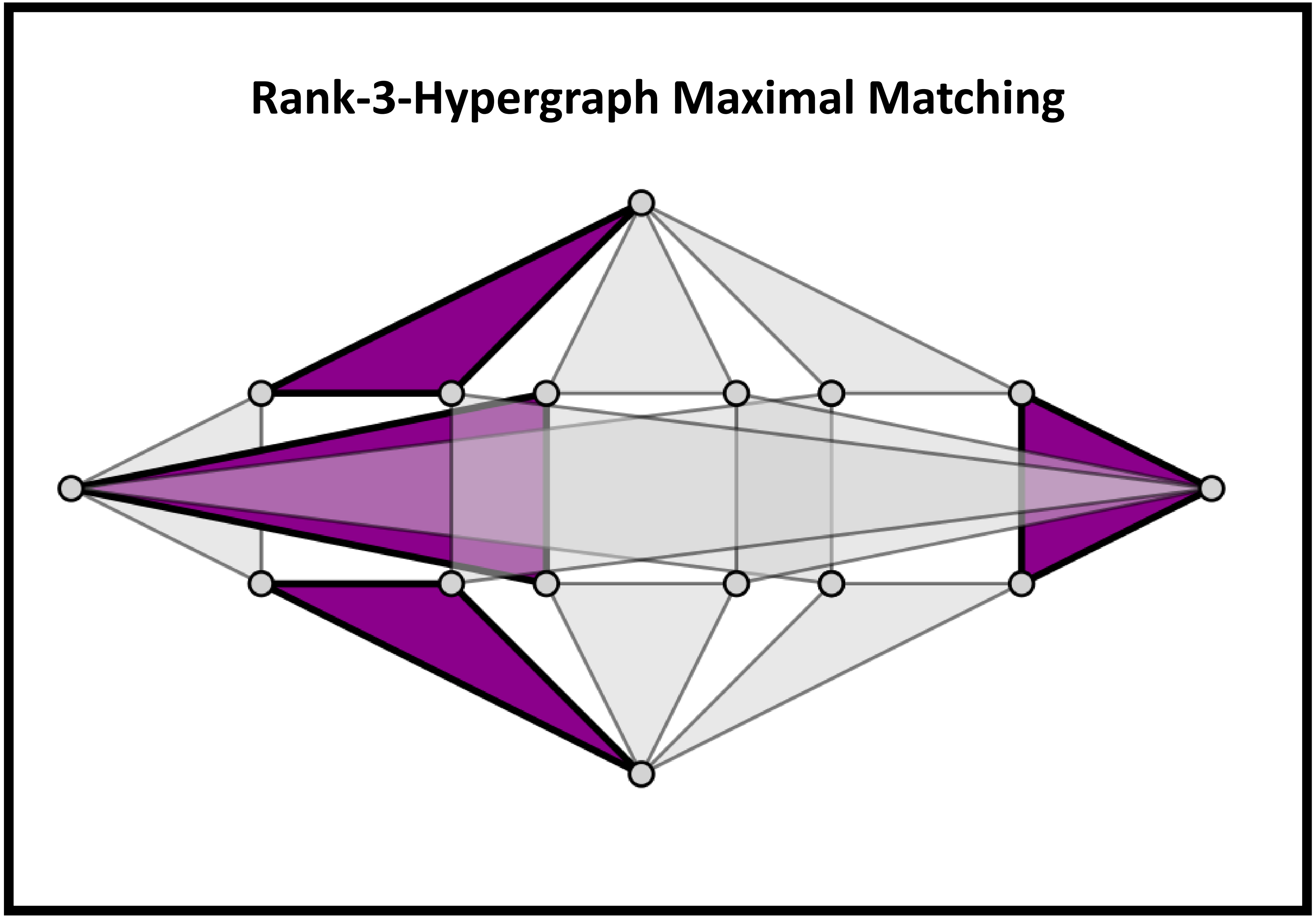


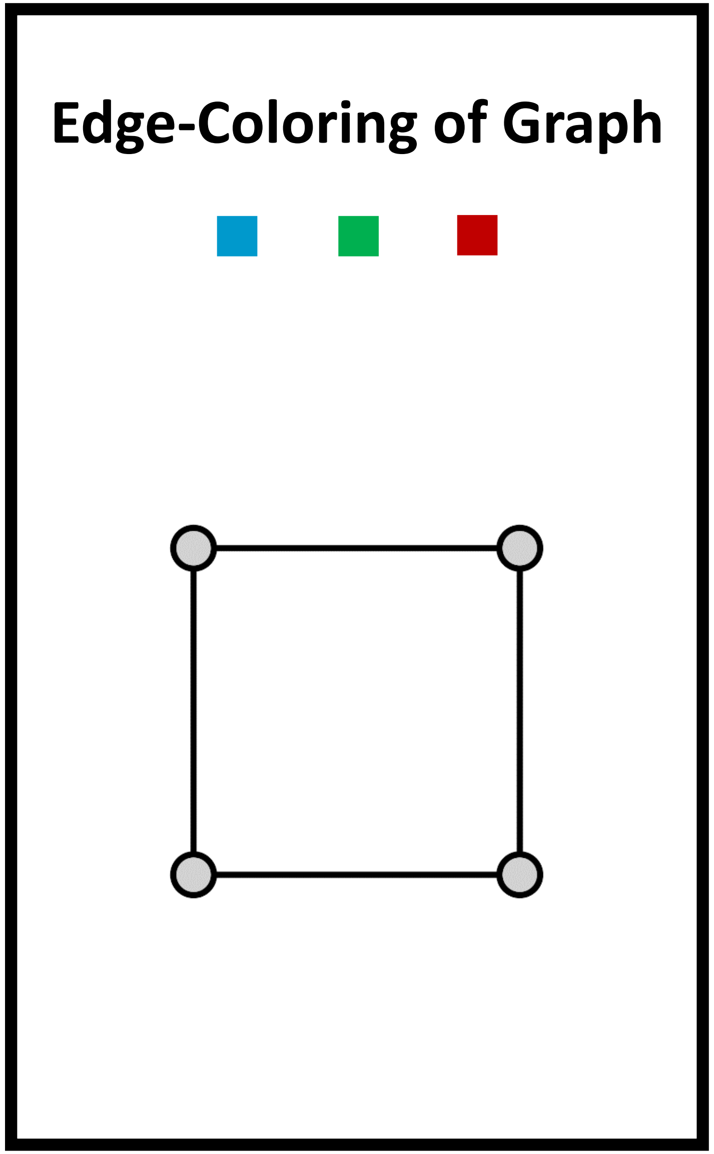
$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching





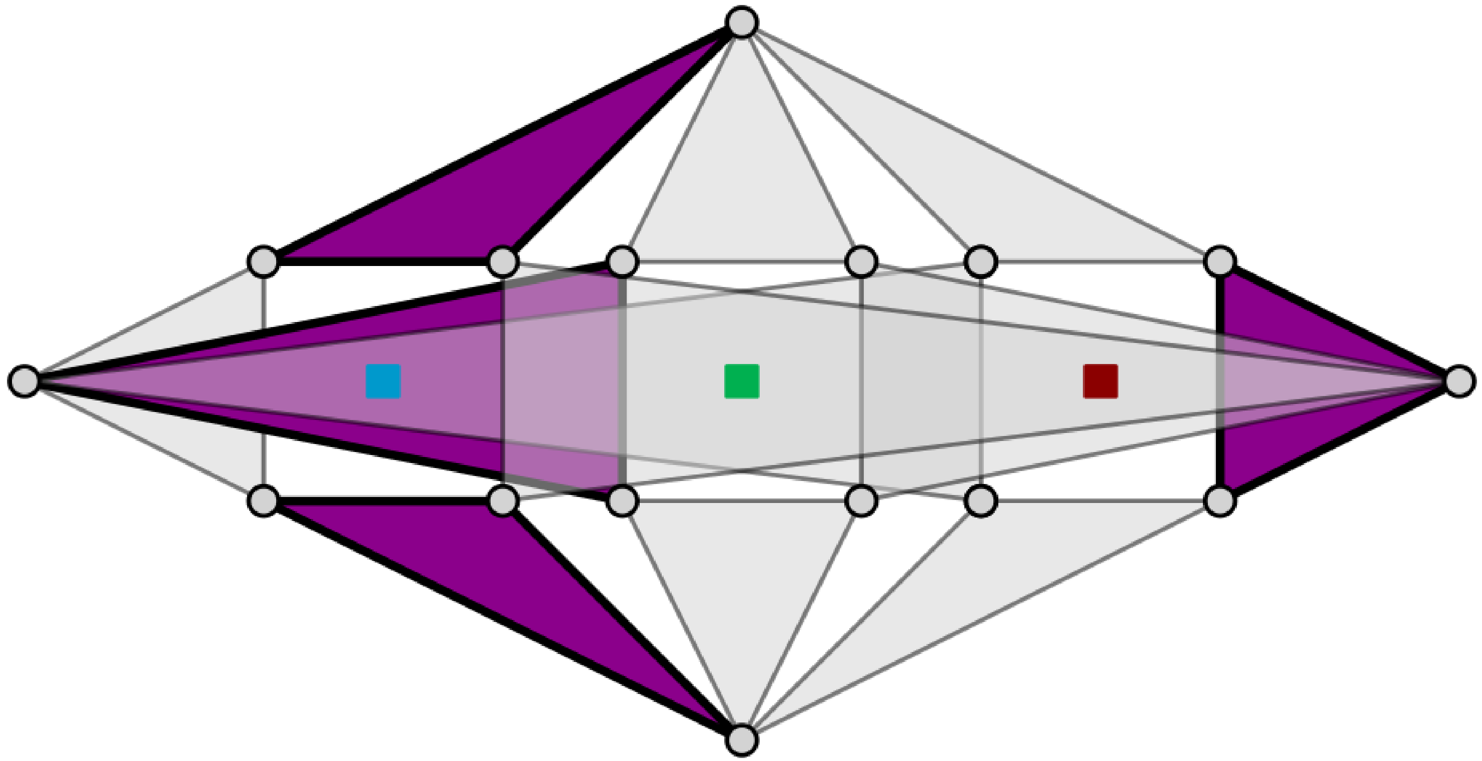
$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching



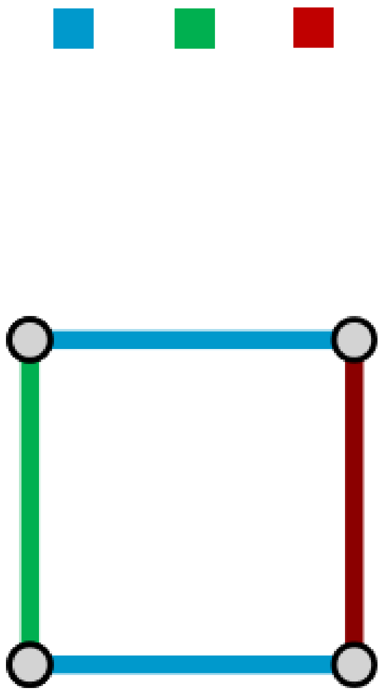


$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

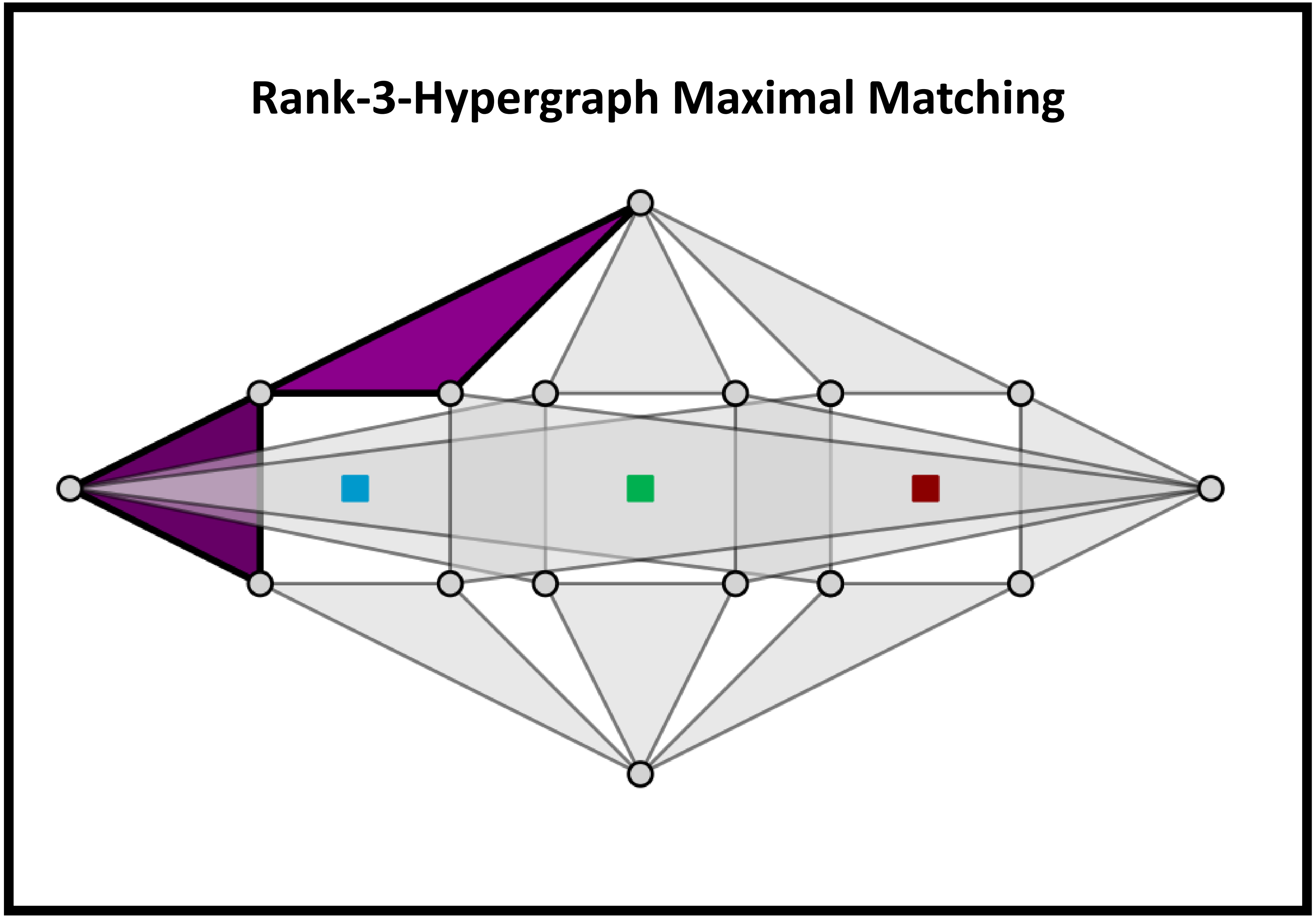
Rank-3-Hypergraph Maximal Matching

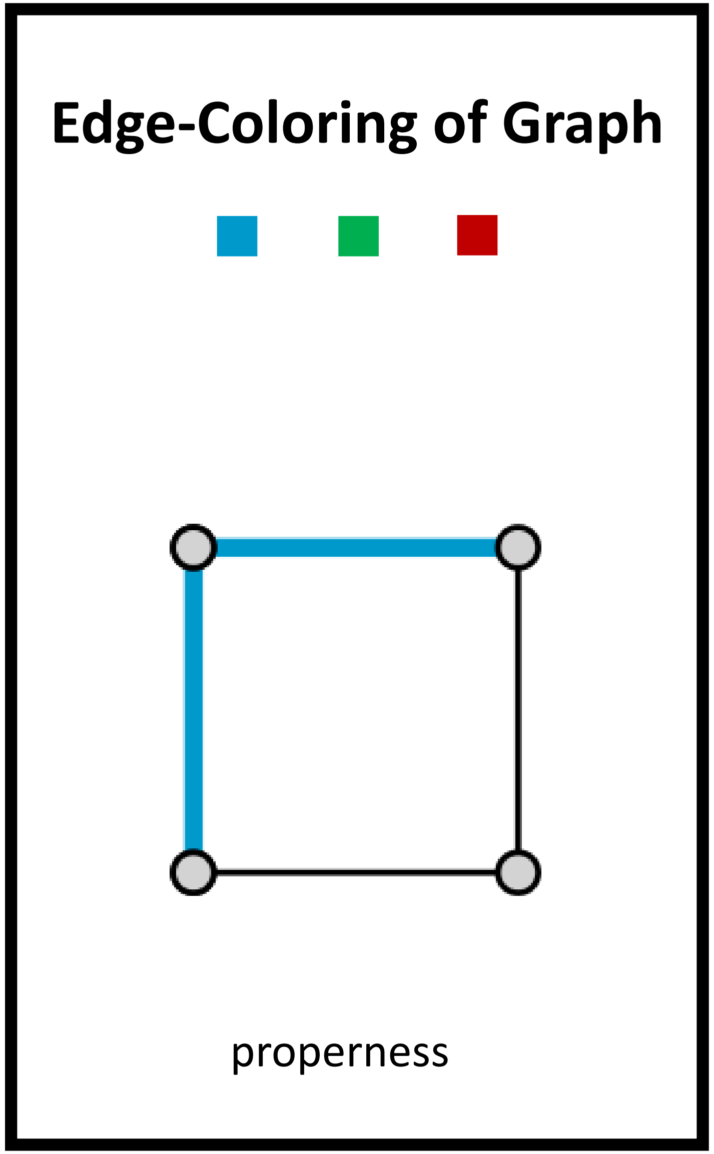


Edge-Coloring of Graph



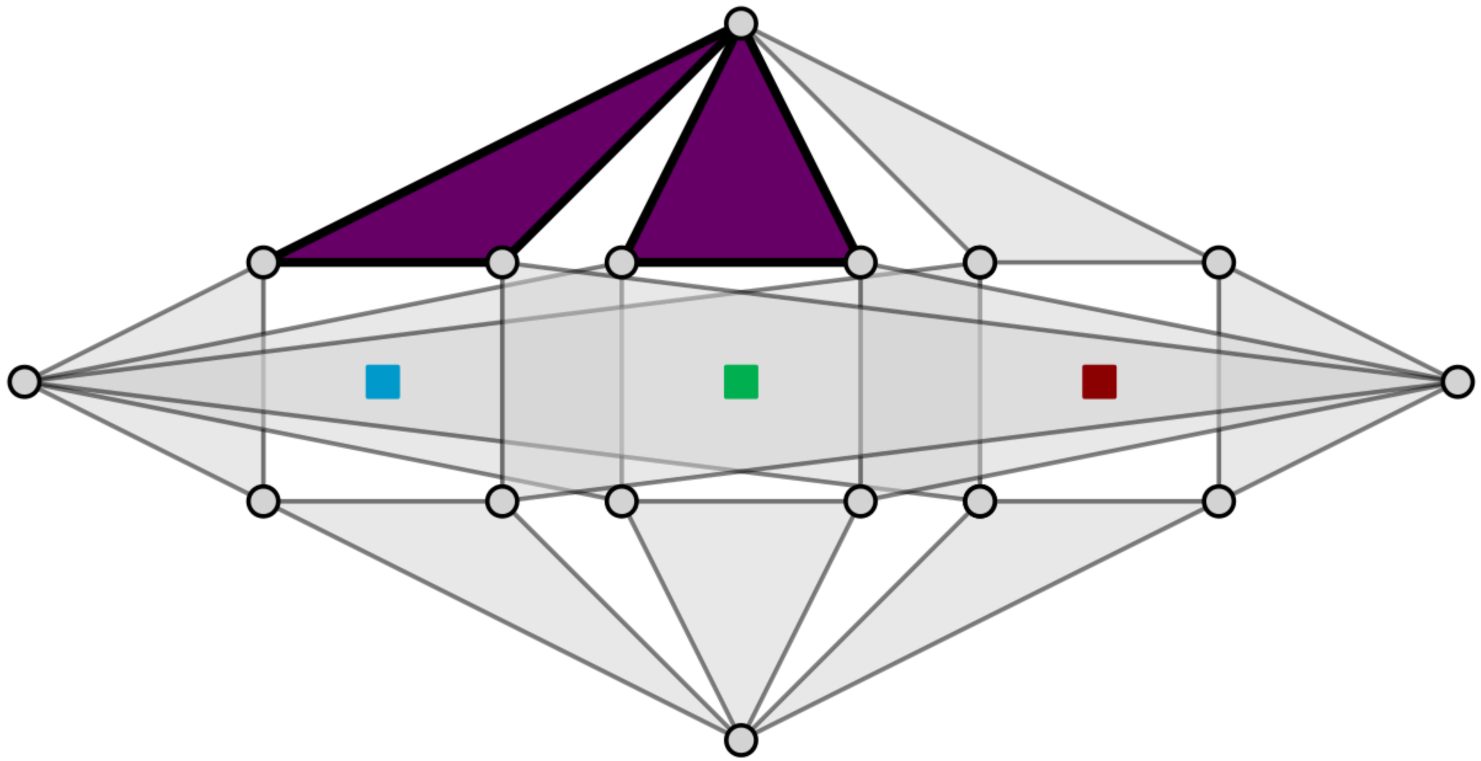
$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching



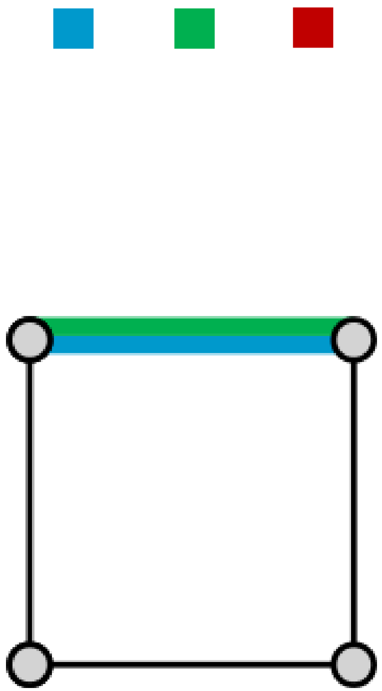


$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

Rank-3-Hypergraph Maximal Matching

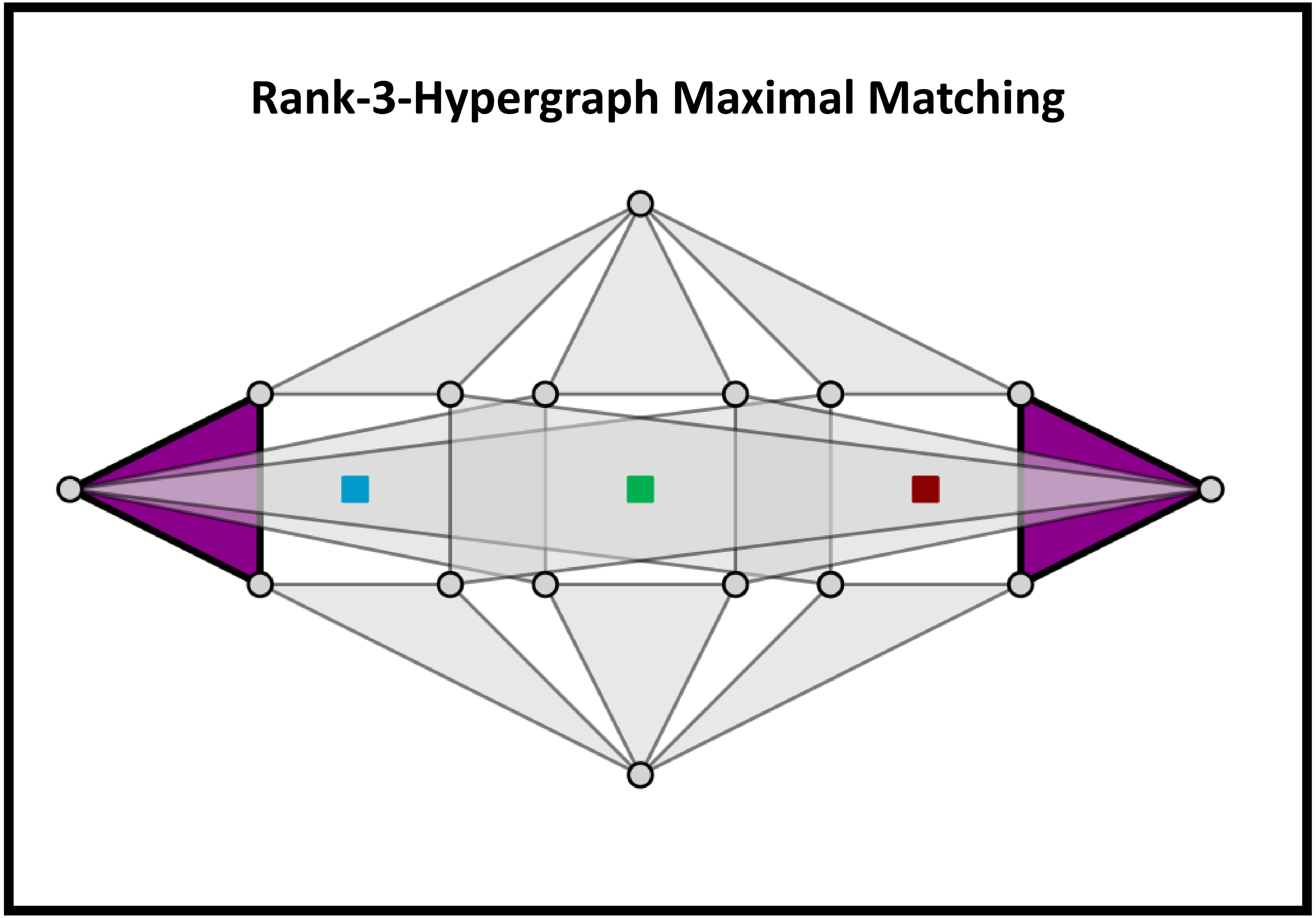


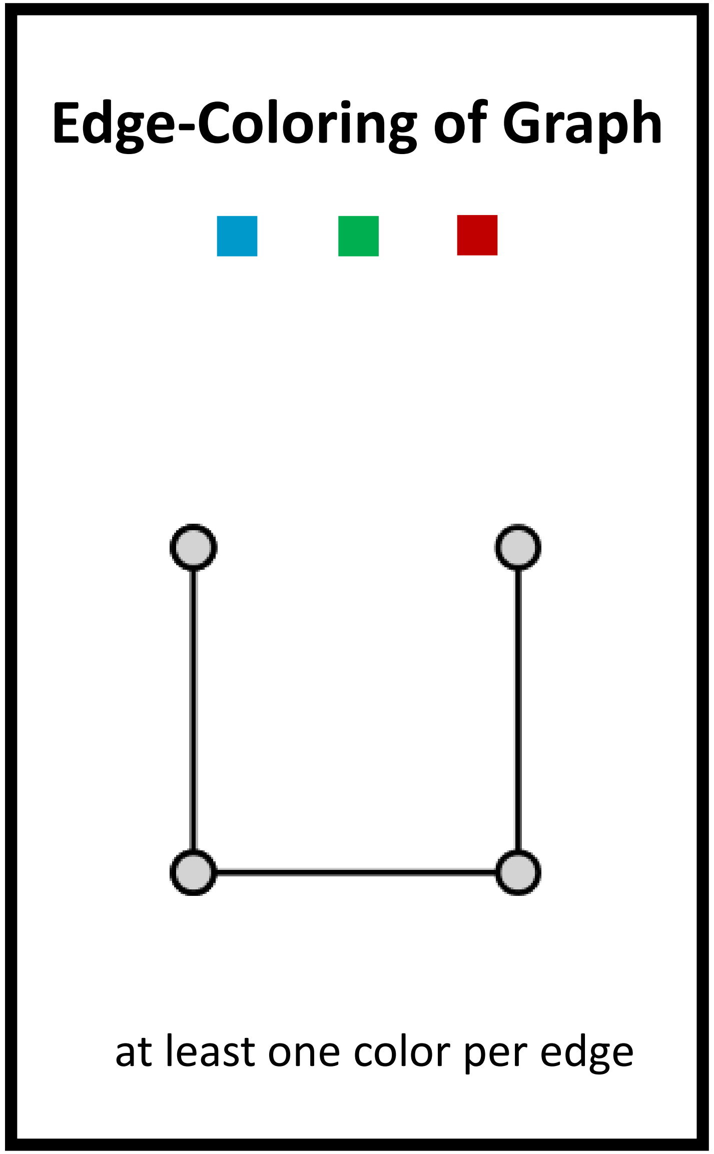
Edge-Coloring of Graph



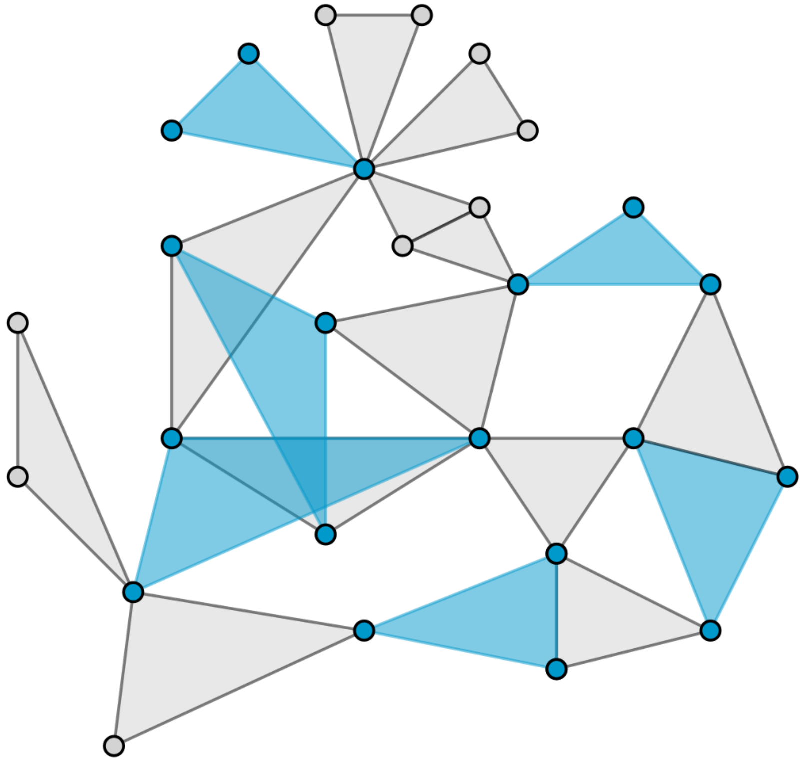
at most one color per edge

$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching





II) Hypergraph Maximal Matching Algorithm



II) Hypergraph Maximal Matching Algorithm

$O(r^2)$-Approximate Maximum Matching

II) Hypergraph Maximal Matching Algorithm

$O(r^2)$-Approximate Maximum Matching

$O(r)$-Approximate Maximum Fractional Matching

II) Hypergraph Maximal Matching Algorithm

$O(r^2)$-Approximate Maximum Matching

$O(r)$-Approximate Maximum Fractional Matching

Rounding Fractional Matching

$(2r)$-Approximate Fractional Matching

Fractional Maximum Matching

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

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

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

$(2r)$-Approximate Fractional Matching

Fractional Maximum Matching

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

value of v

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

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

$(2r)$-Approximate Fractional Matching

Fractional Maximum Matching

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

value of v

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

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

LOCAL Greedy Algorithm

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

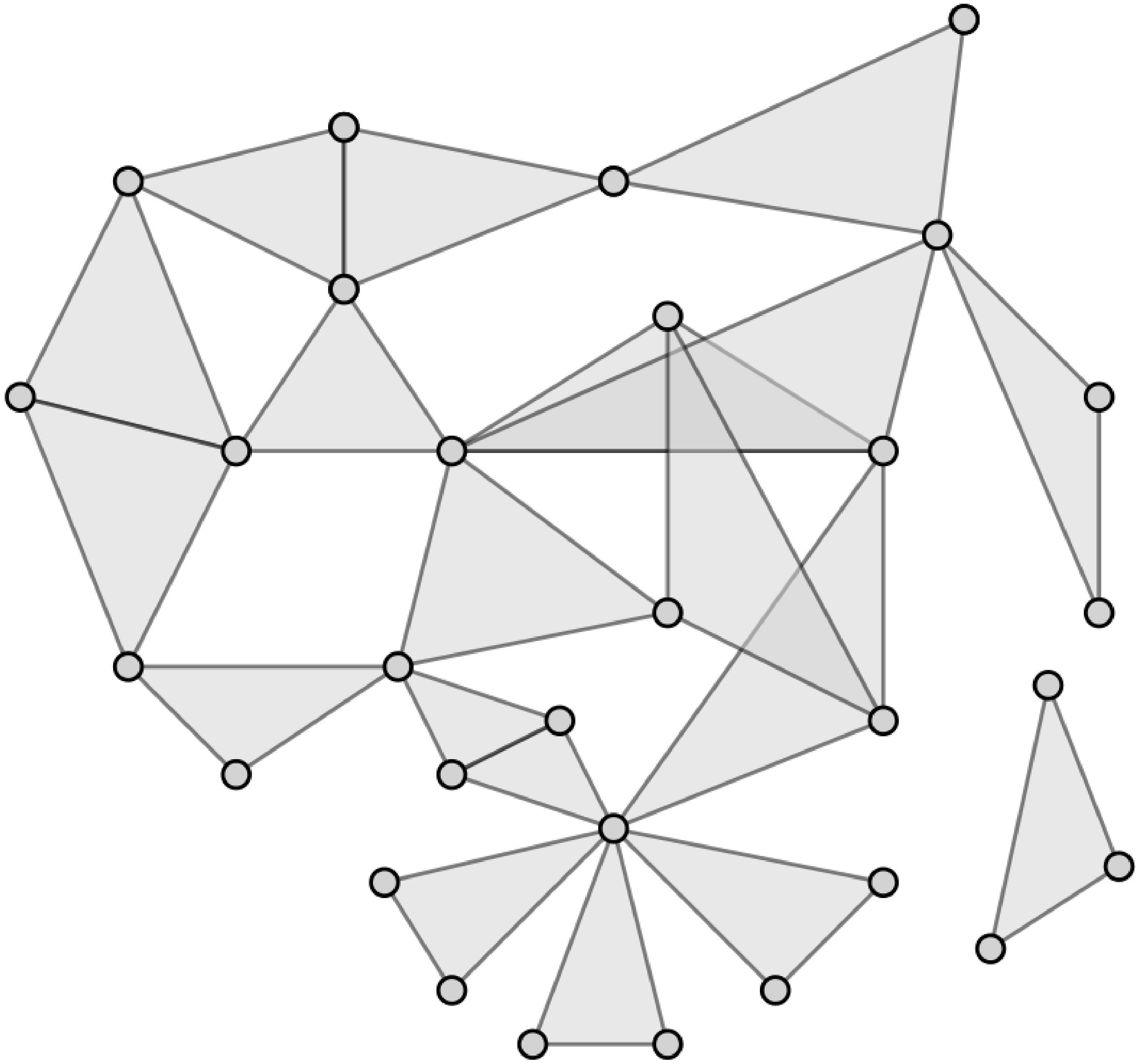
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



Fractional Maximum Matching

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

value of v

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

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

LOCAL Greedy Algorithm

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

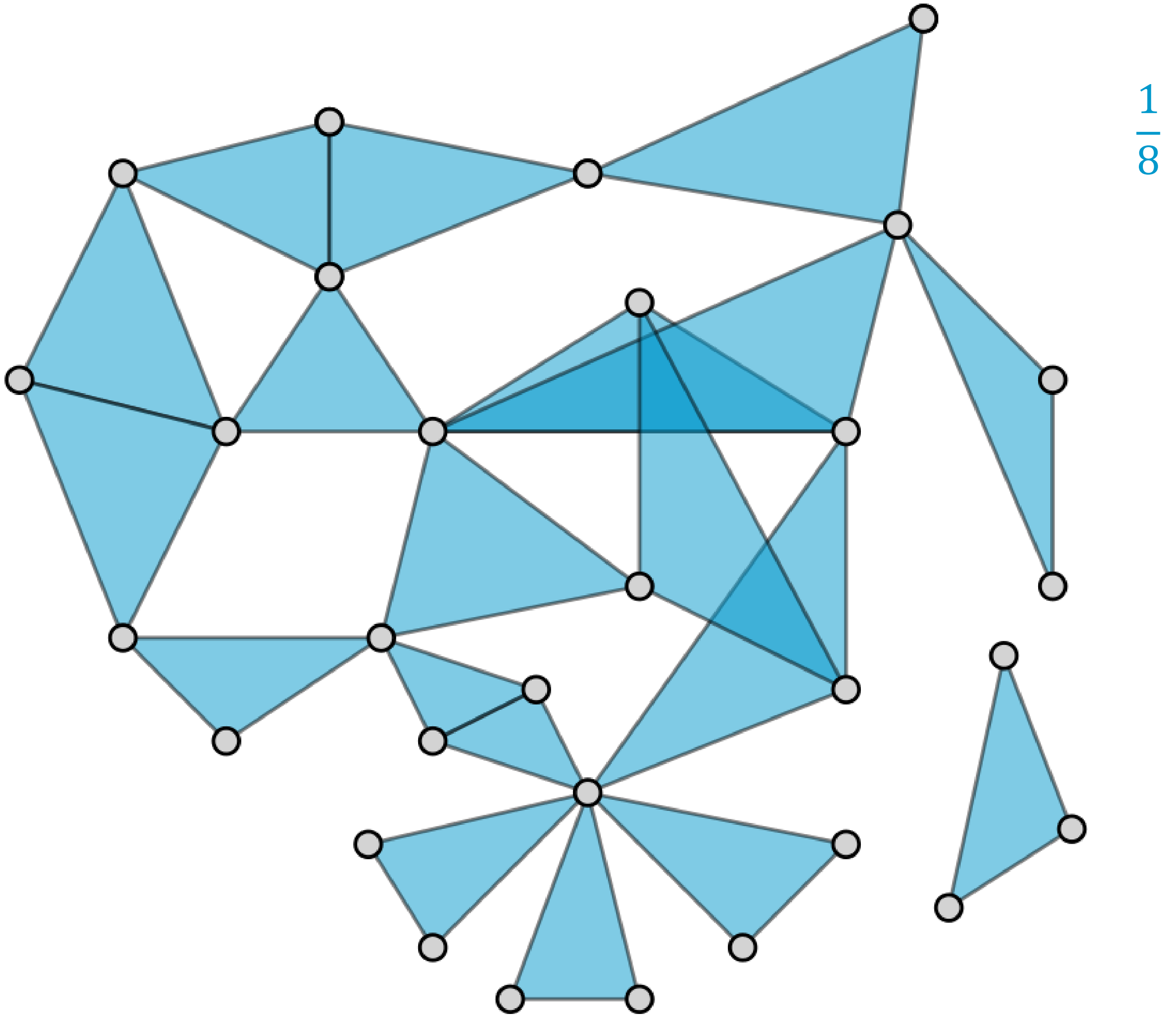
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



Fractional Maximum Matching

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

value of v

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

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

LOCAL Greedy Algorithm

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

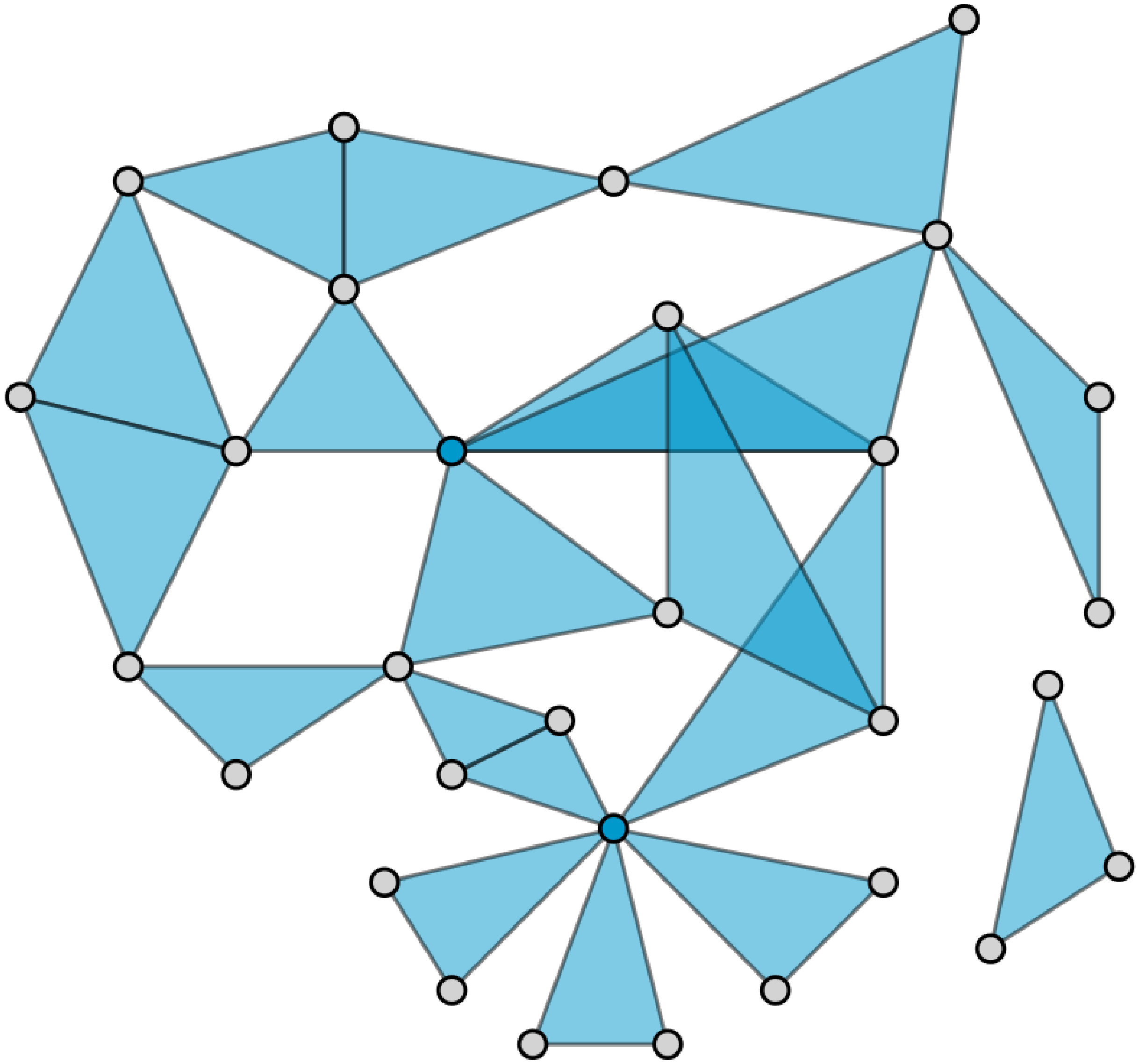
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



$\frac{1}{8}$

Fractional Maximum Matching

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

value of v

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

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

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

LOCAL Greedy Algorithm

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

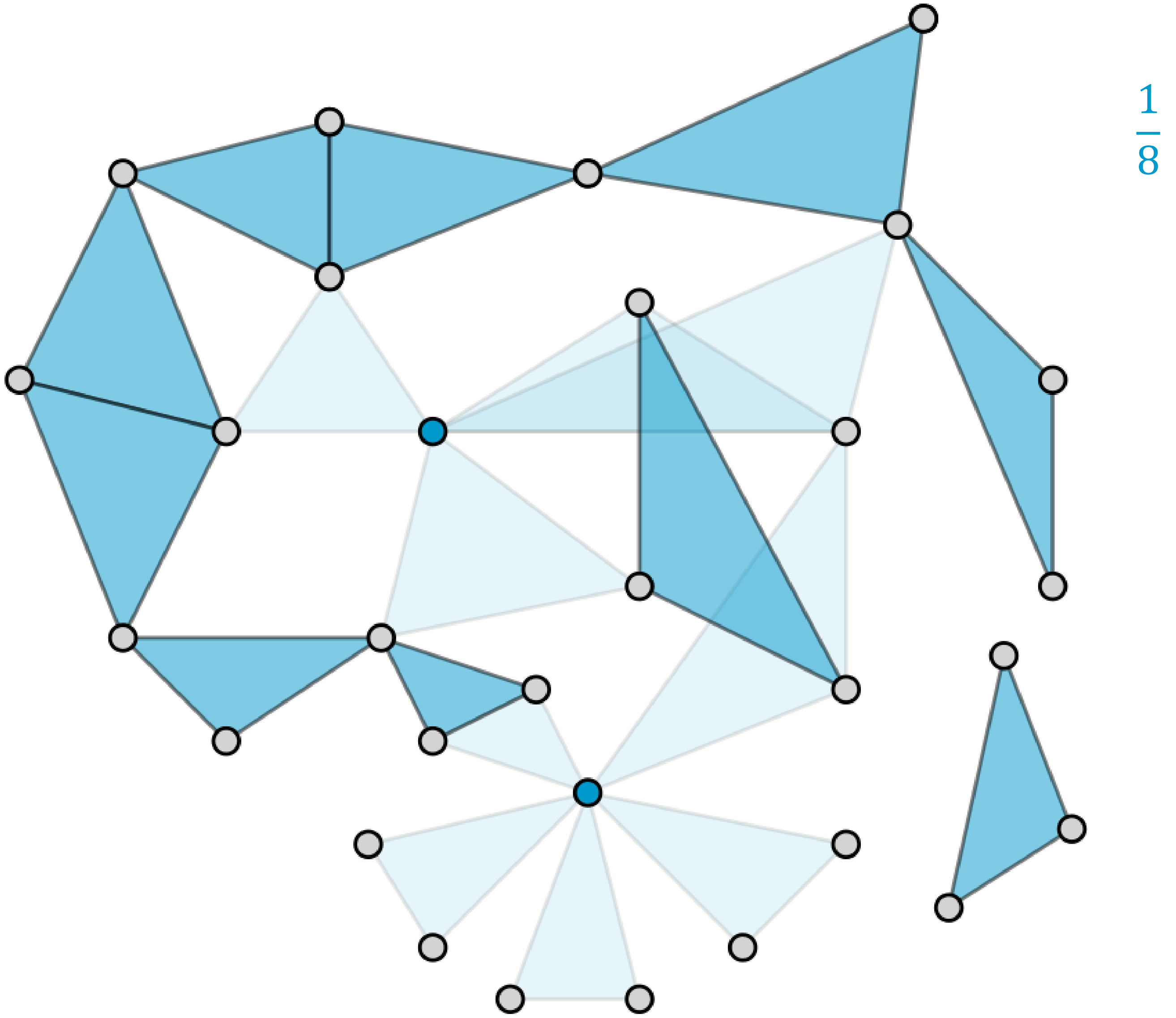
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



Fractional Maximum Matching

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

value of v

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

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

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

LOCAL Greedy Algorithm

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

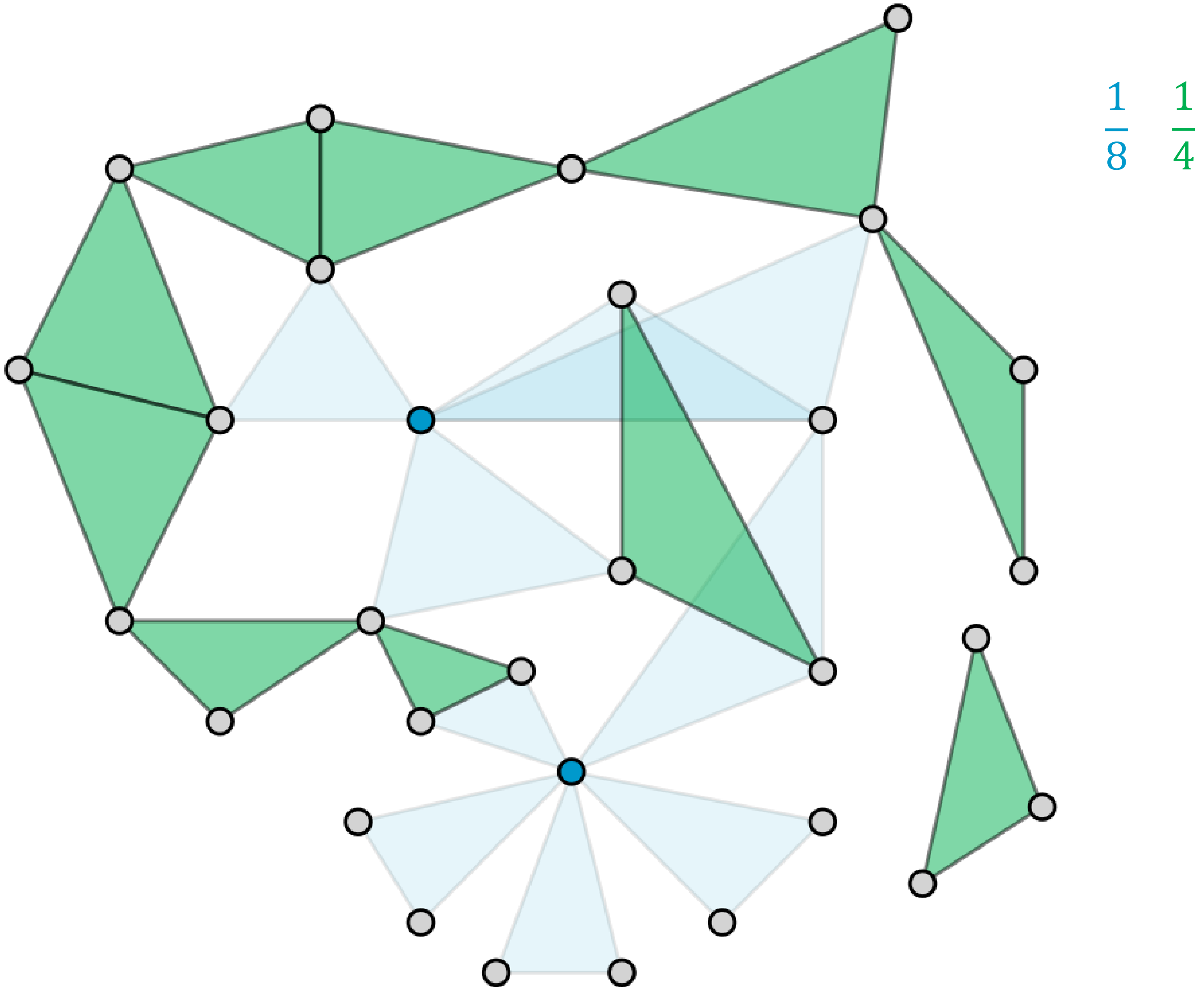
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



Fractional Maximum Matching

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

value of v

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

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

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

LOCAL Greedy Algorithm

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

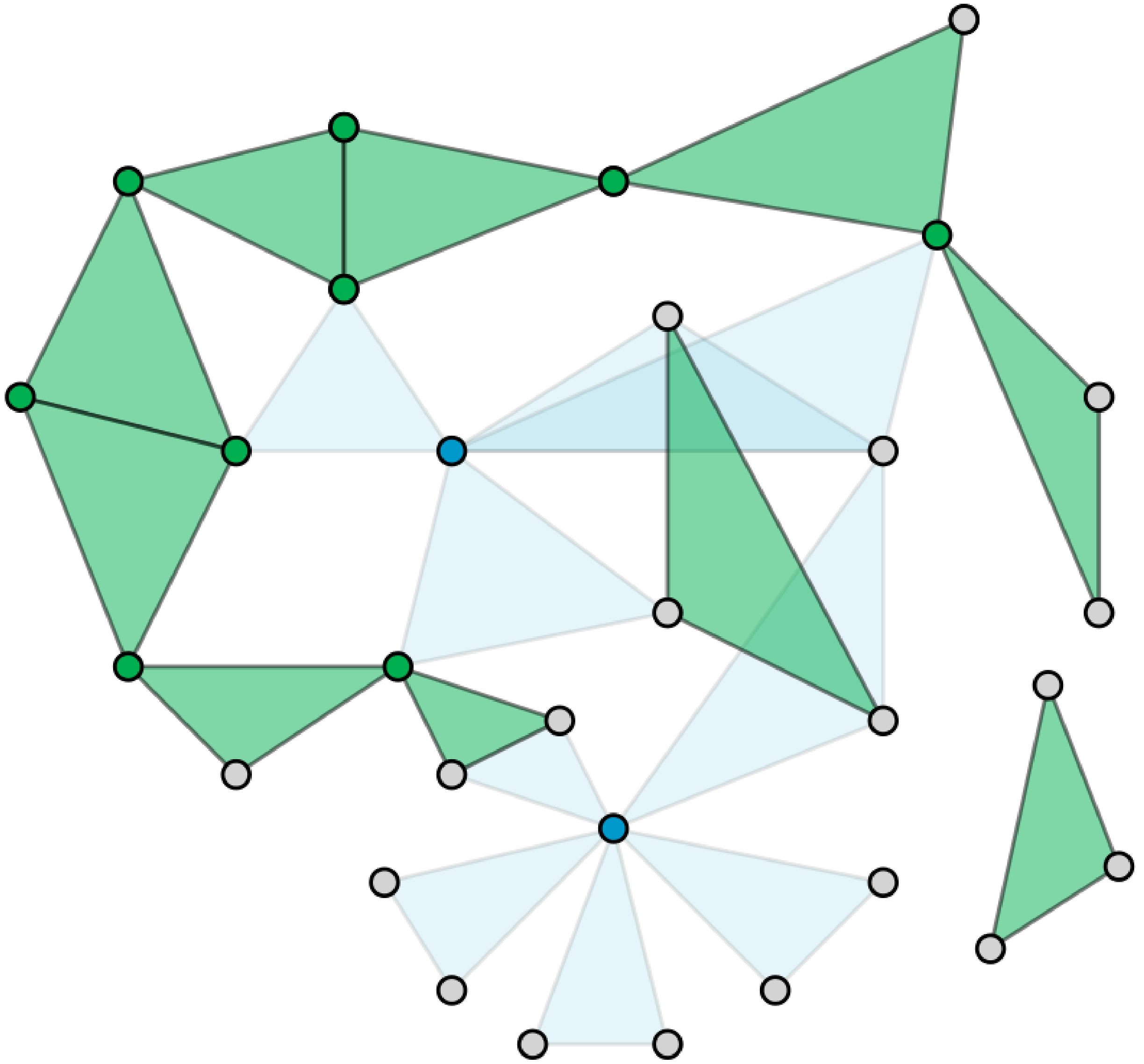
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



Fractional Maximum Matching

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

value of v

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

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

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

LOCAL Greedy Algorithm

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

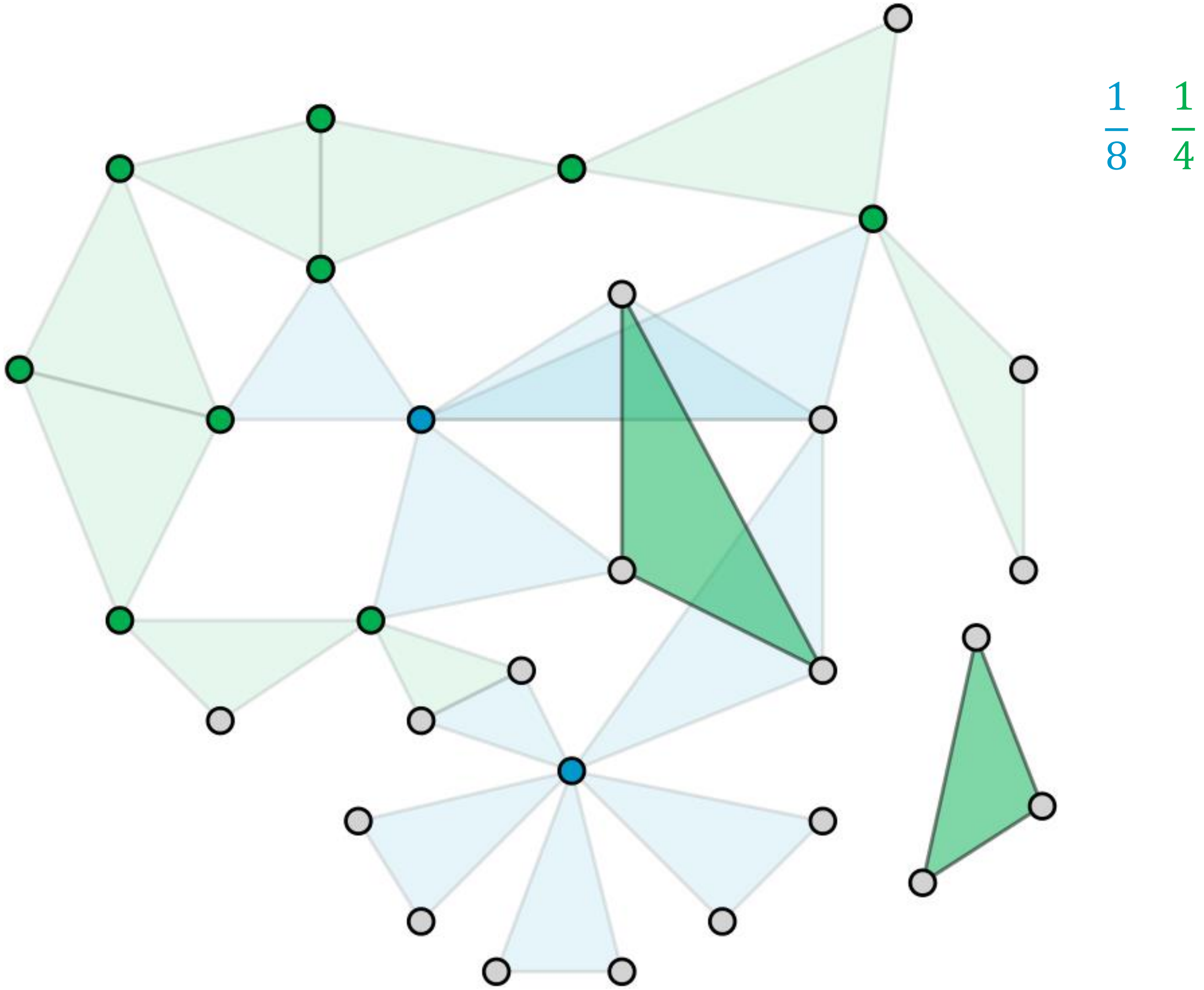
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



Fractional Maximum Matching

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

value of v

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

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

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

LOCAL Greedy Algorithm

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

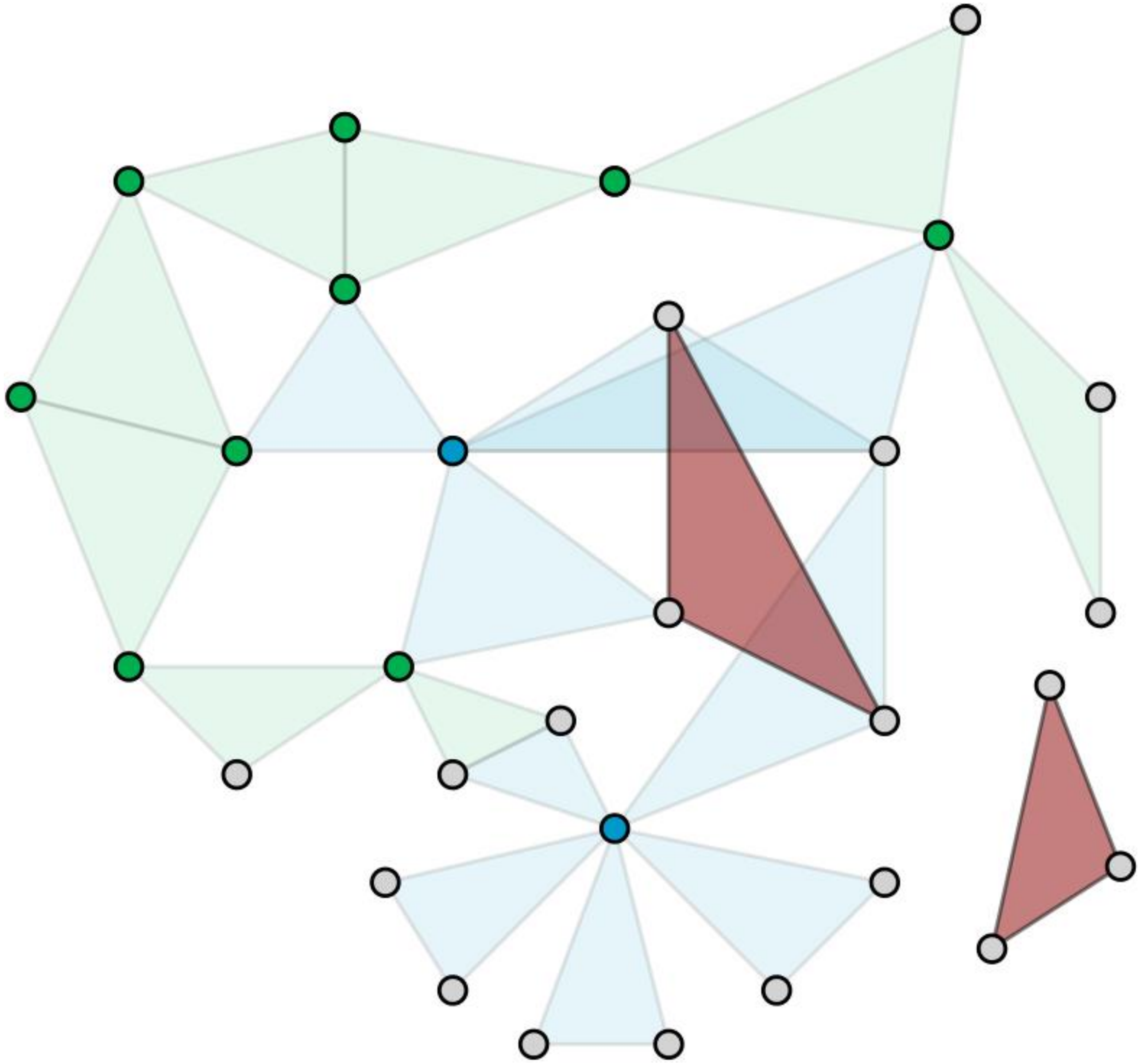
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



Fractional Maximum Matching

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

value of v

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

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

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

LOCAL Greedy Algorithm

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

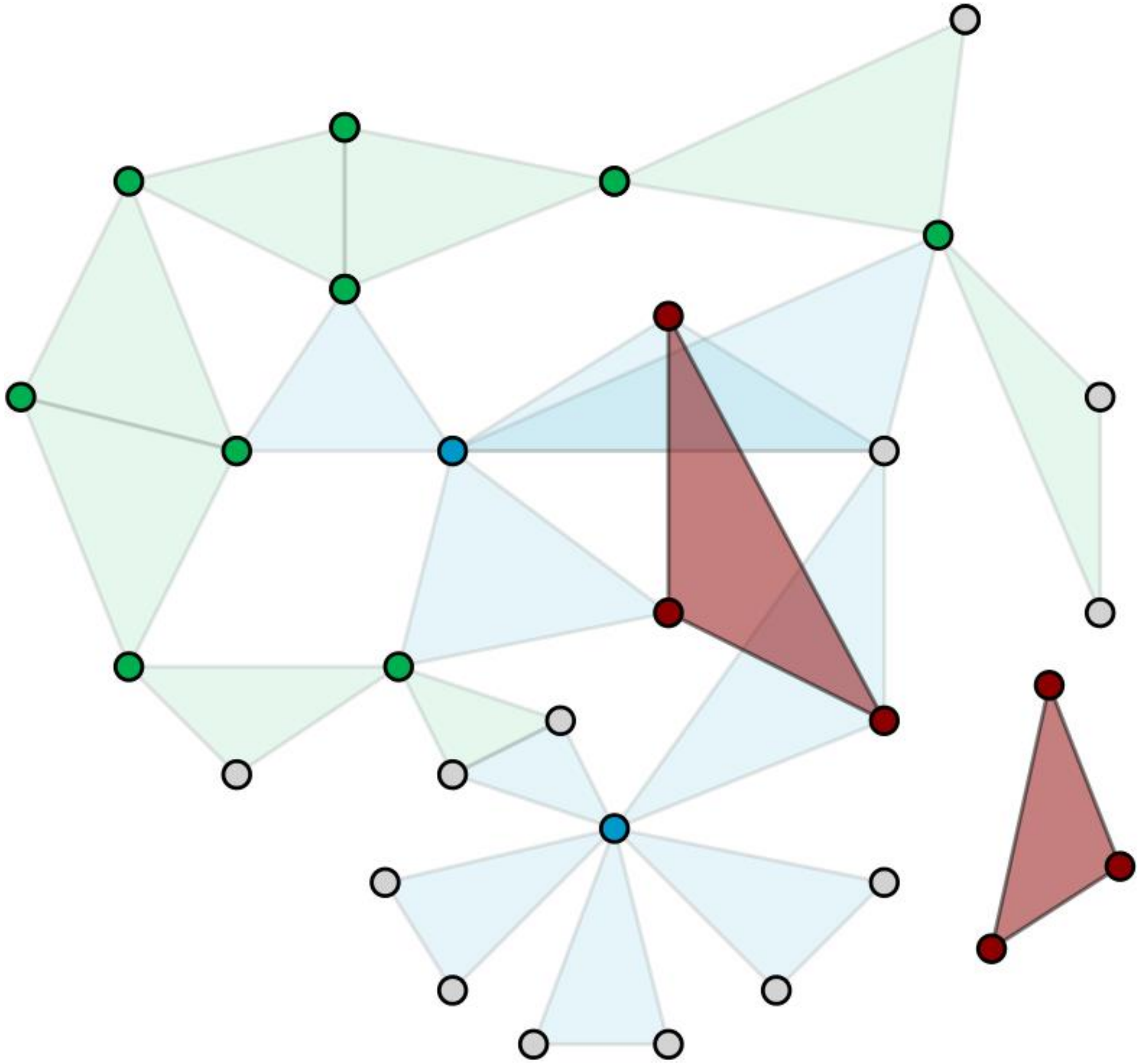
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



Fractional Maximum Matching

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

value of v

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

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

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

LOCAL Greedy Algorithm

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

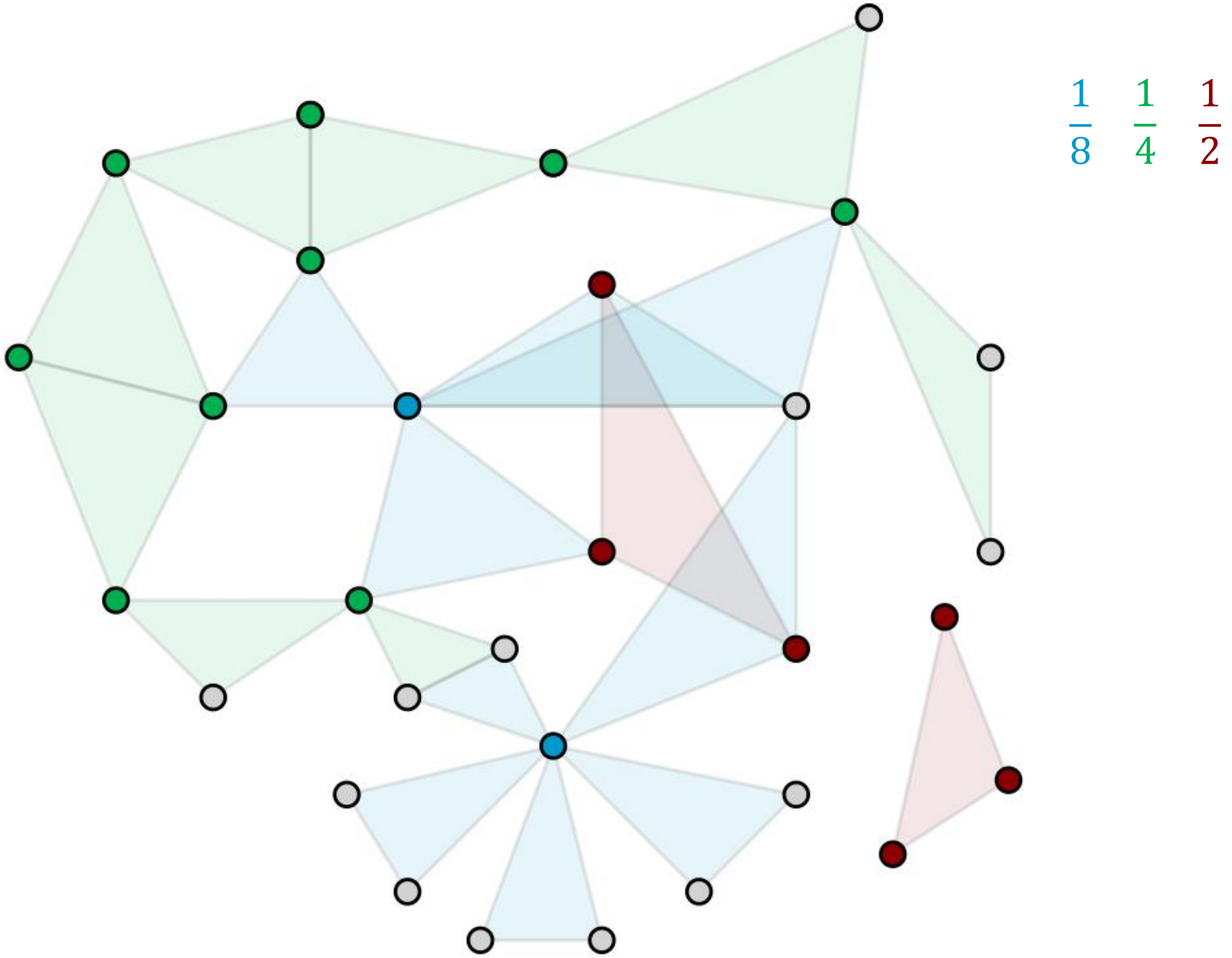
repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges

$(2r)$-Approximate Fractional Matching



Fractional Maximum Matching

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

value of v

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

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

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

LOCAL Greedy Algorithm

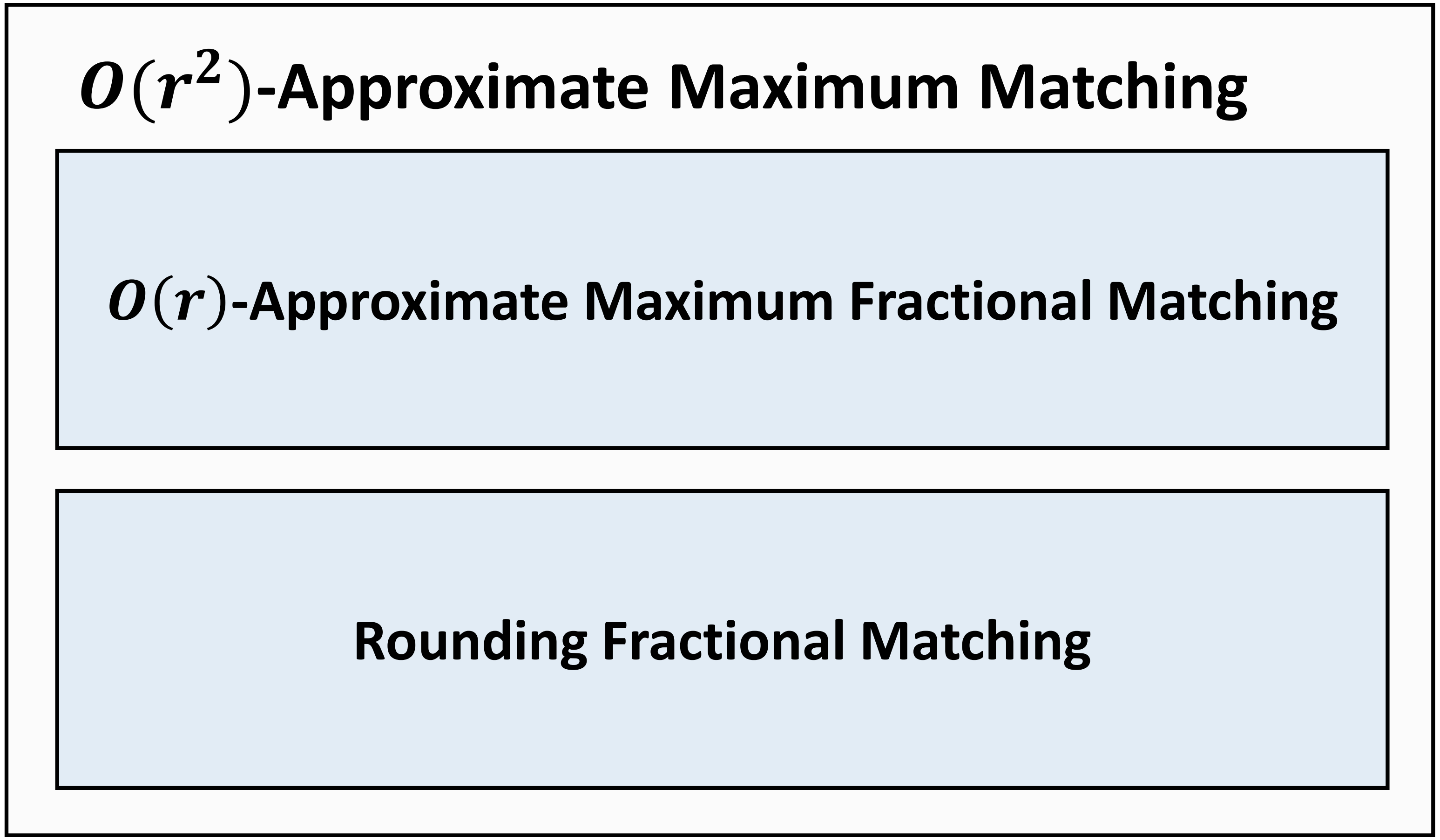
$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

mark half-tight nodes

block their edges

double value of unblocked edges


$O(r^2)$-Approximate Maximum Matching
$O(r)$-Approximate Maximum Fractional Matching
Rounding Fractional Matching

Rounding Fractional Matching

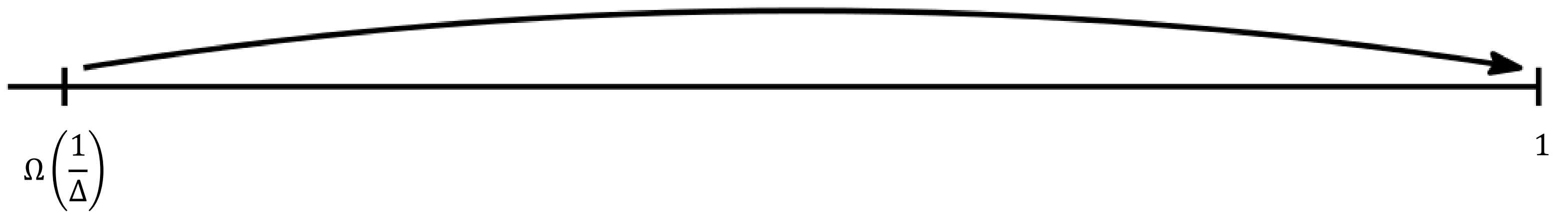
Rounding Fractional Matching

Rounding Fractional Matching

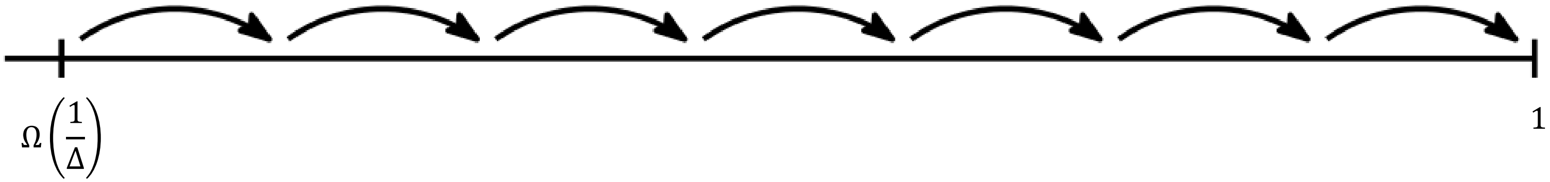
Direct Rounding

Rounding Fractional Matching

Direct Rounding



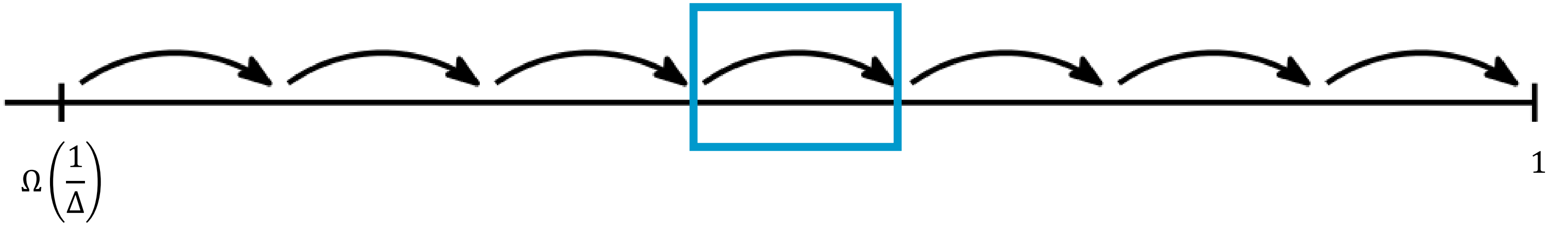
Iterative Rounding



Rounding Fractional Matching

Direct Rounding

Iterative Rounding



Factor-2-Rounding

Factor-2-Rounding

Graphs:

factor-2-rounding with essentially no loss

approach used by Hańćkowiak, Karoński, Panconesi [SODA’98, PODC’99] and F. [DISC’17]

Factor-2-Rounding

Graphs:

factor-2-rounding with essentially no loss

approach used by Hańćkowiak, Karoński, Panconesi [SODA'98, PODC'99] and F. [DISC'17]

Factor-2-Rounding

Graphs:

factor-2-rounding with essentially no loss

approach used by Hańćkowiak, Karoński, Panconesi [SODA'98, PODC'99] and F. [DISC'17]

Hypergraphs?

Challenge:
factor-2-rounding with almost no loss

Factor-2-Rounding

Graphs:

factor-2-rounding with essentially no loss

approach used by Hańćkowiak, Karoński, Panconesi [SODA'98, PODC'99] and F. [DISC'17]

Hypergraphs?

Challenge:
factor-2-rounding with almost no loss

The Only Obstacle
Ghaffari, Kuhn, Maus [STOC'17]

Factor-2-Rounding

Graphs:

factor-2-rounding with essentially no loss

approach used by Hańćkowiak, Karoński, Panconesi [SODA'98, PODC'99] and F. [DISC'17]

Hypergraphs?

Challenge:
factor-2-rounding with almost no loss

Challenge 1:
factor-2-rounding with loss $O(r)$

The Only Obstacle
Ghaffari, Kuhn, Maus [STOC'17]

Factor-2-Rounding

Graphs:

factor-2-rounding with essentially no loss

approach used by Hańćkowiak, Karoński, Panconesi [SODA'98, PODC'99] and F. [DISC'17]

Hypergraphs?

Challenge:
factor-2-rounding with almost no loss

The Only Obstacle
Ghaffari, Kuhn, Maus [STOC'17]

Challenge 1:
factor-2-rounding with loss $O(r)$

Basic Rounding:
factor-L-rounding with loss $O(r)$

Basic Rounding

Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0

Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

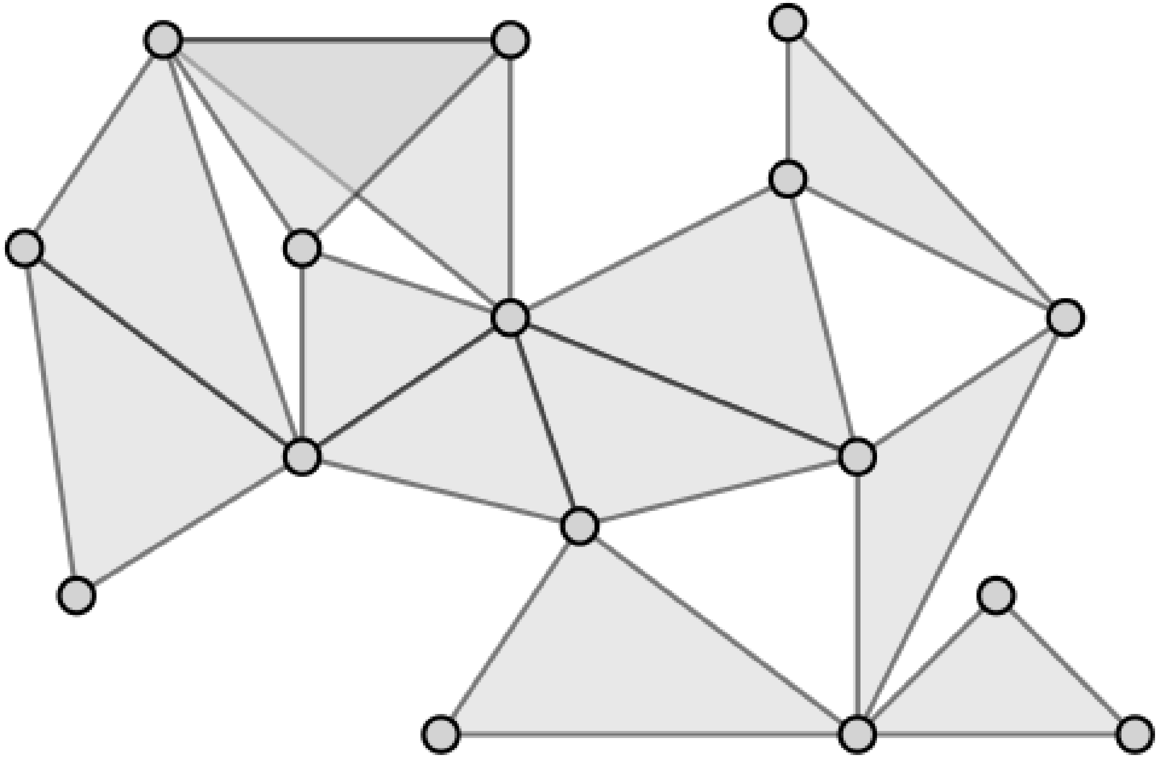
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

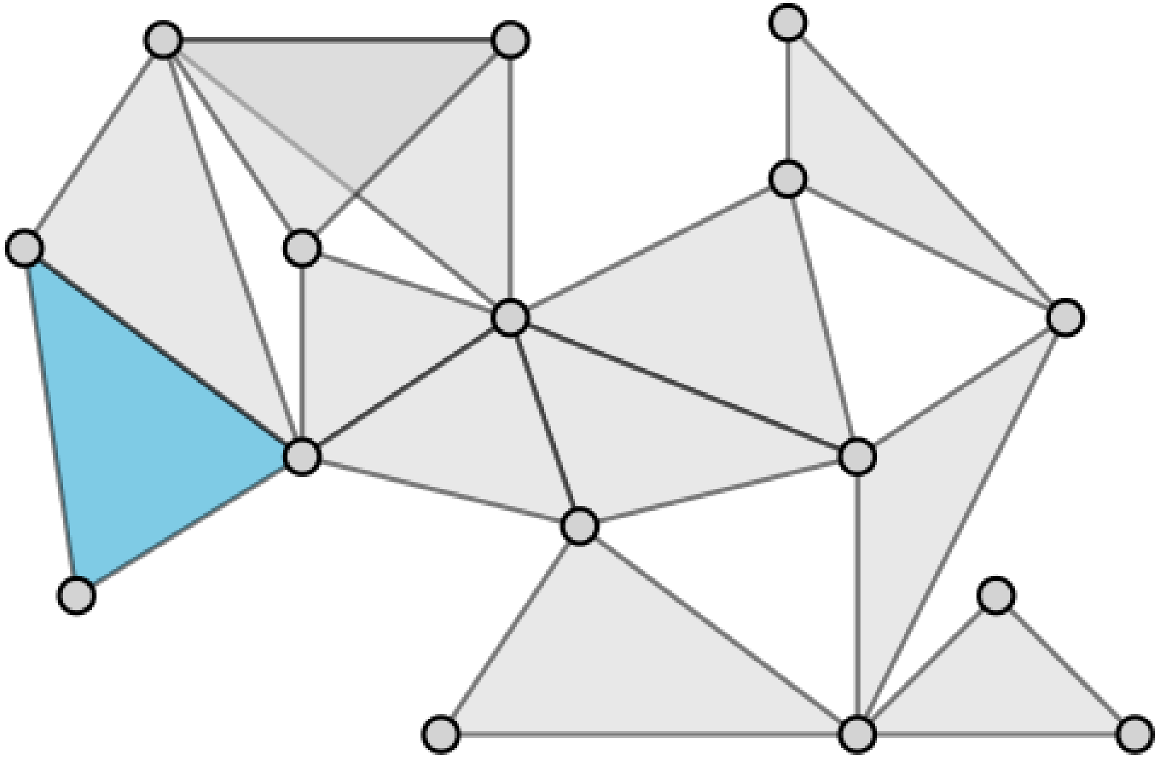
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

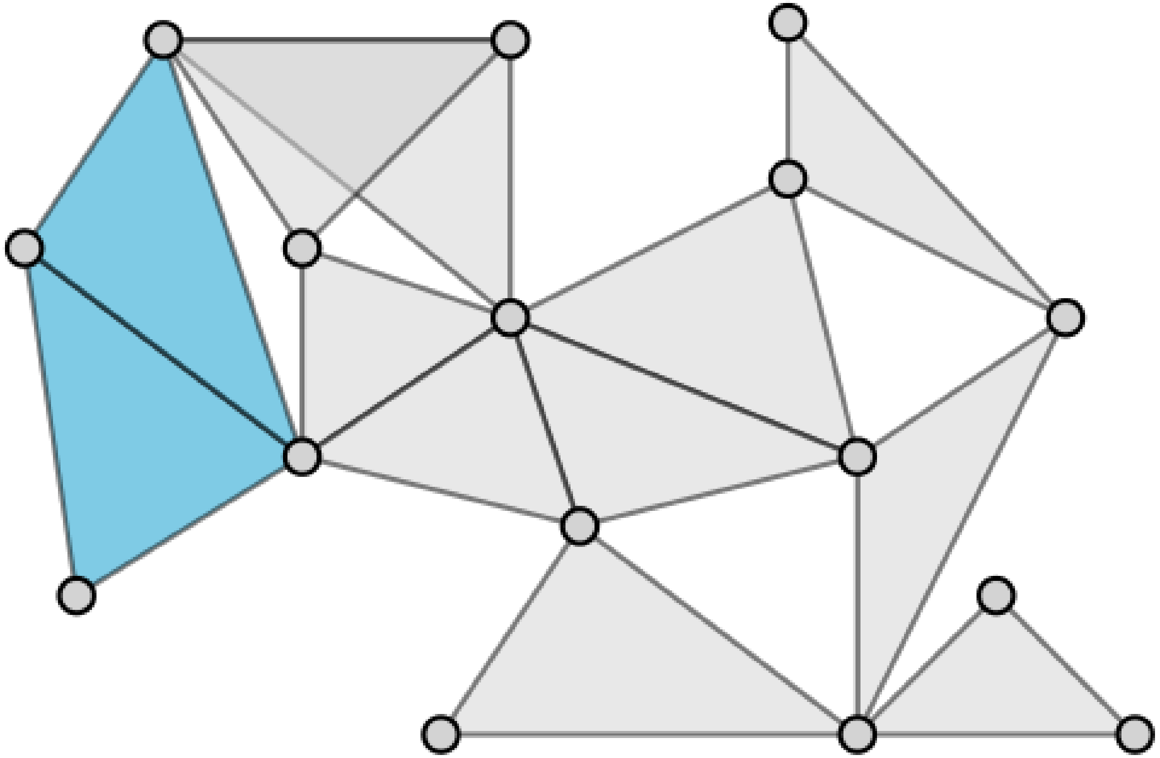
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

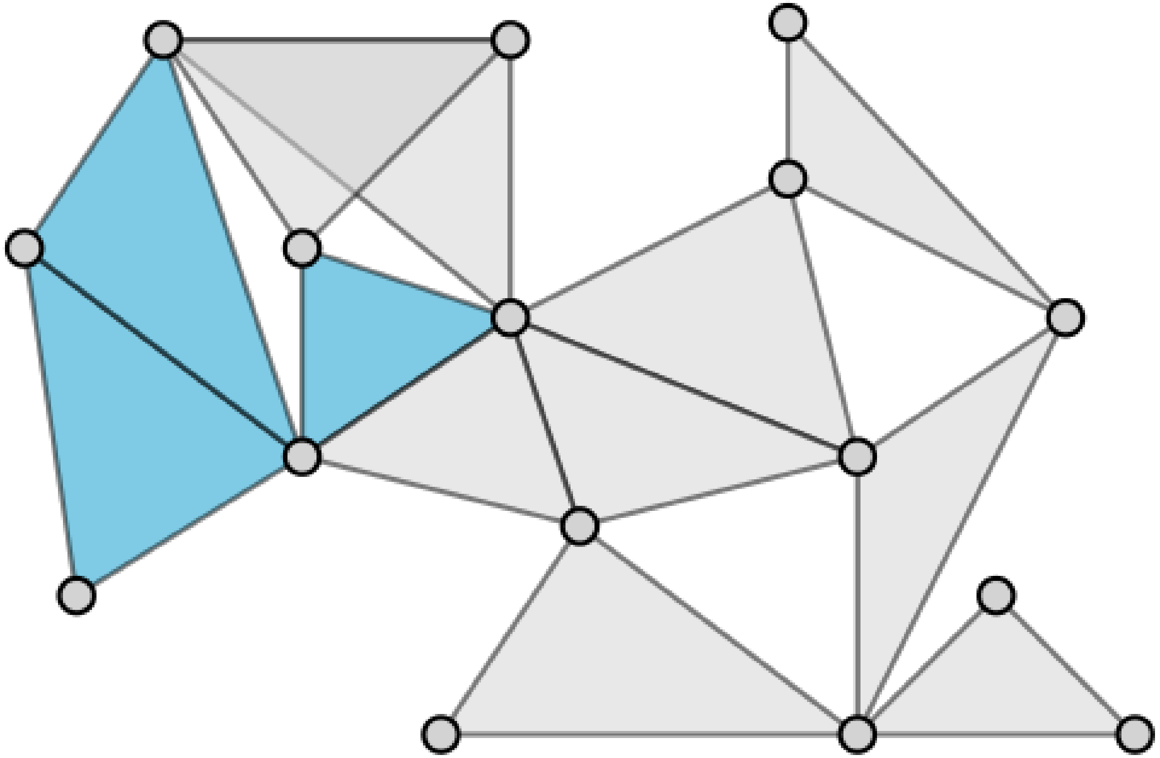
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

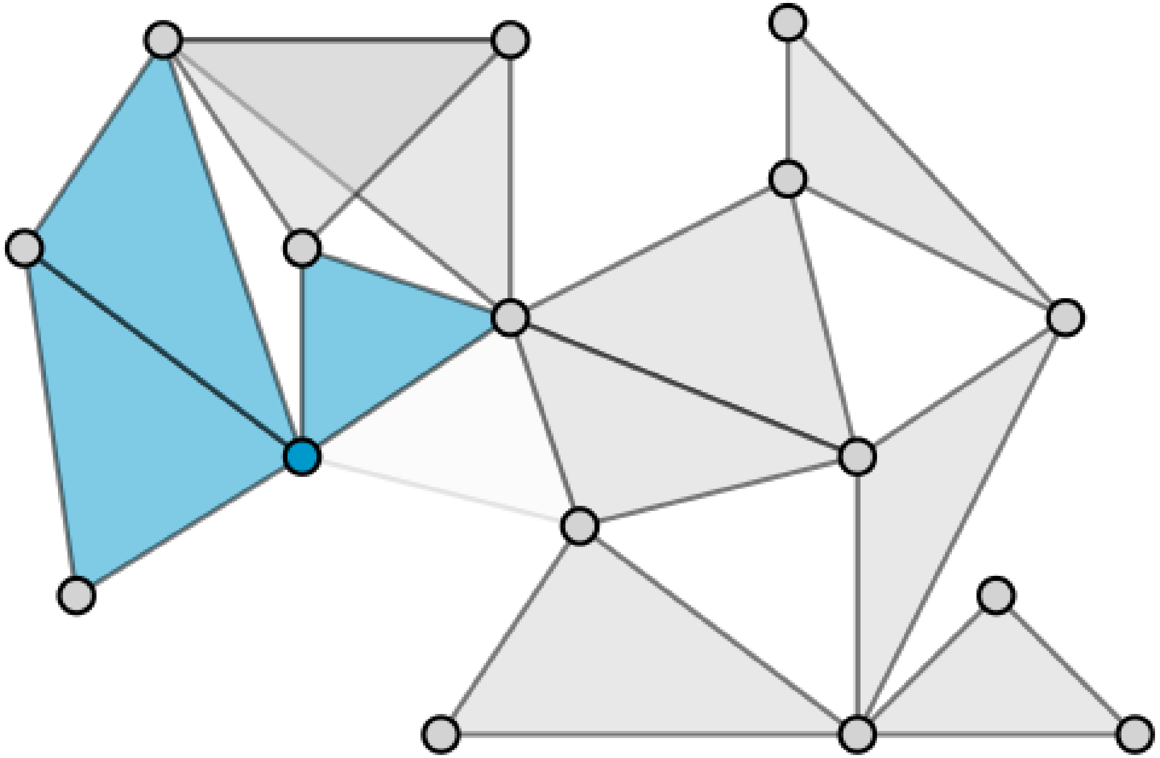
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

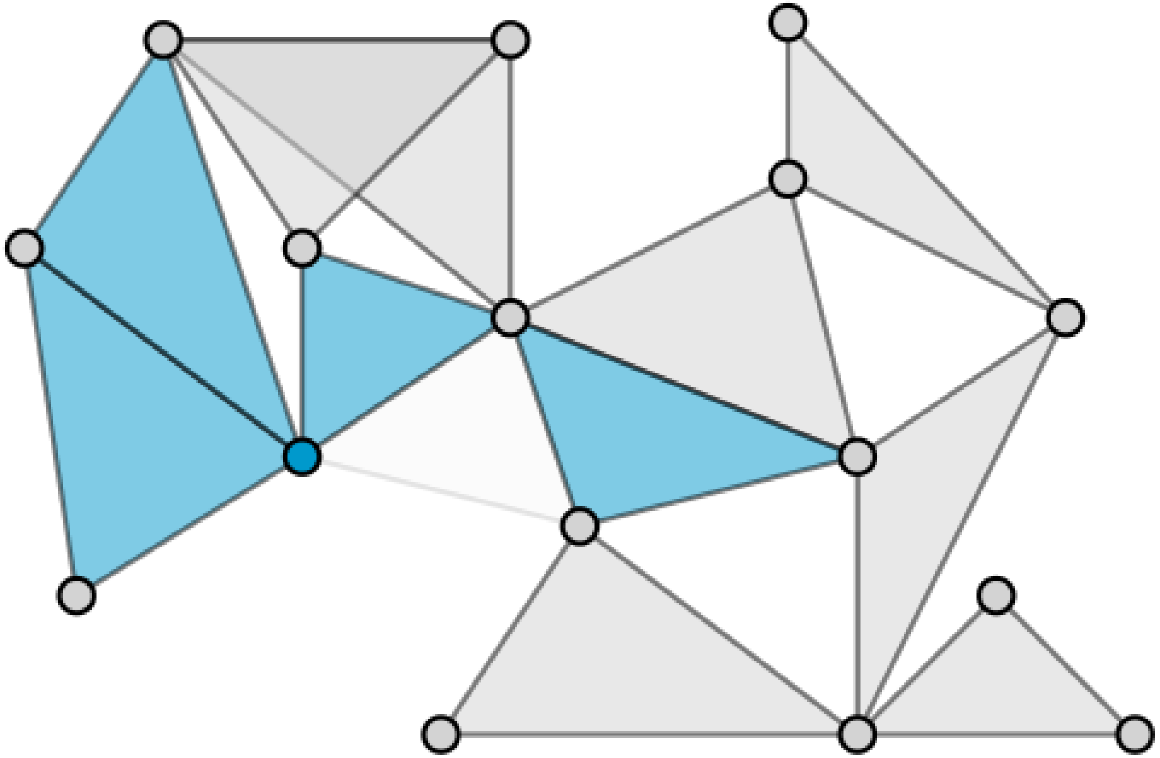
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

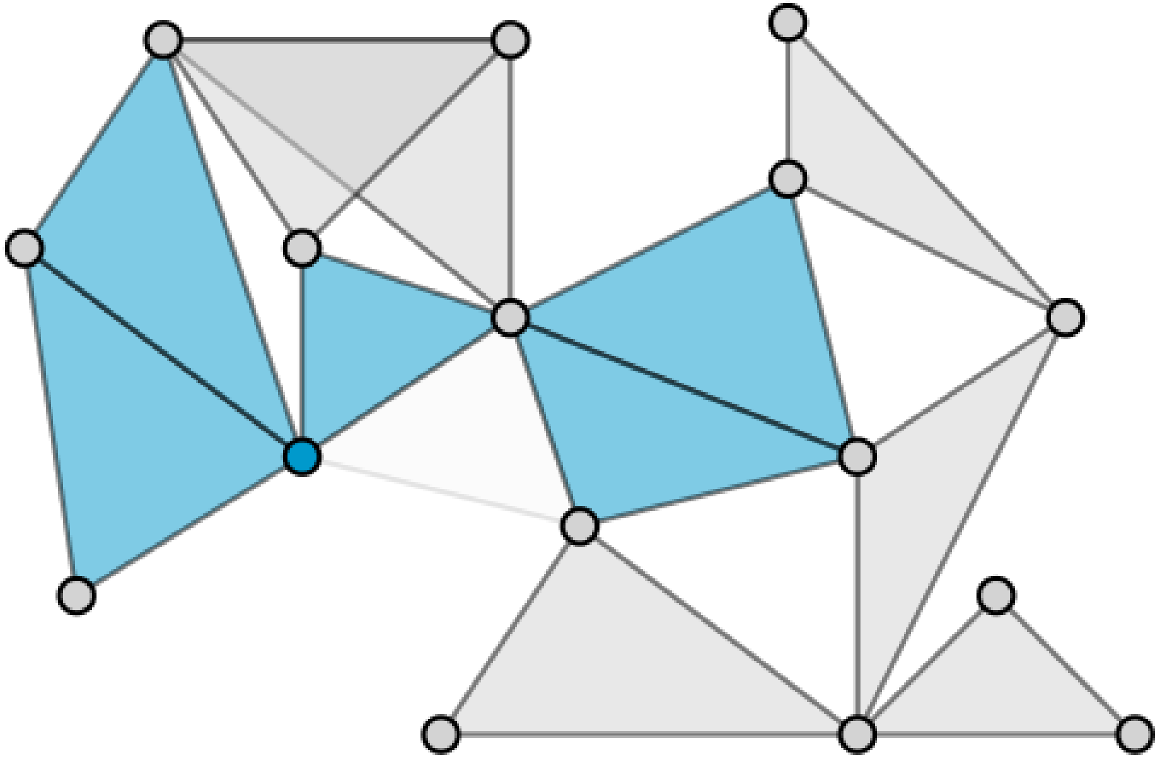
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

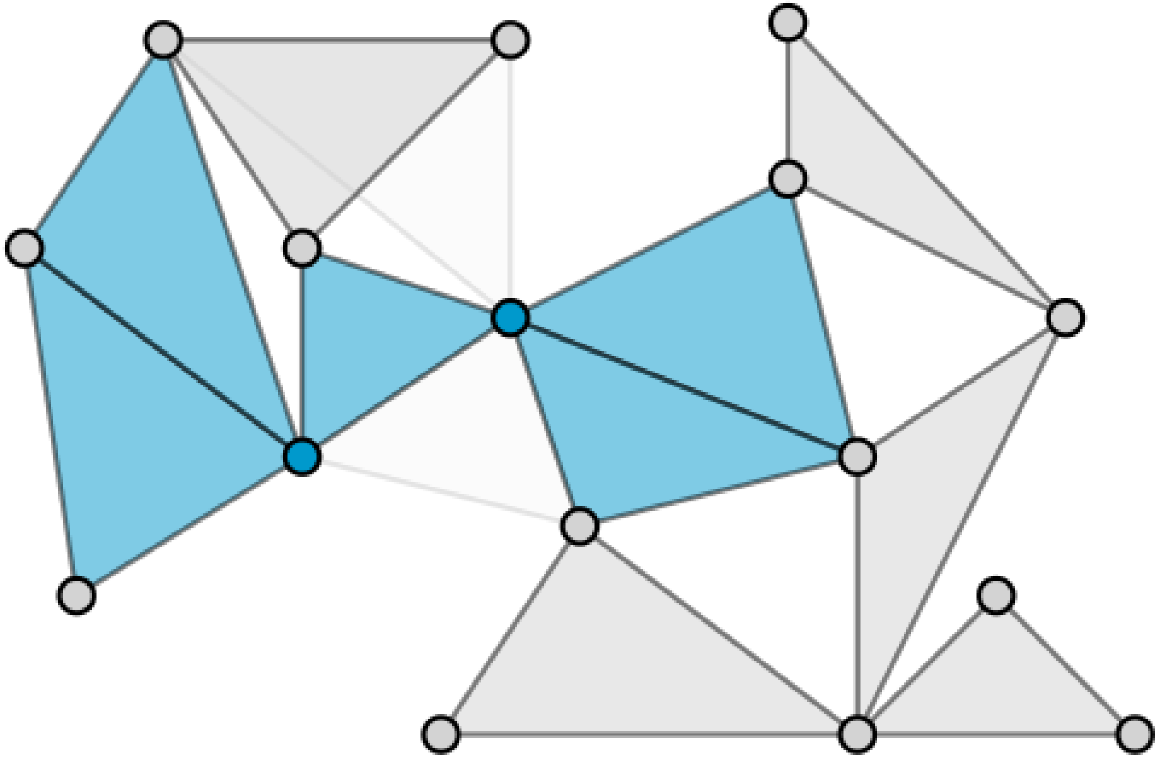
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

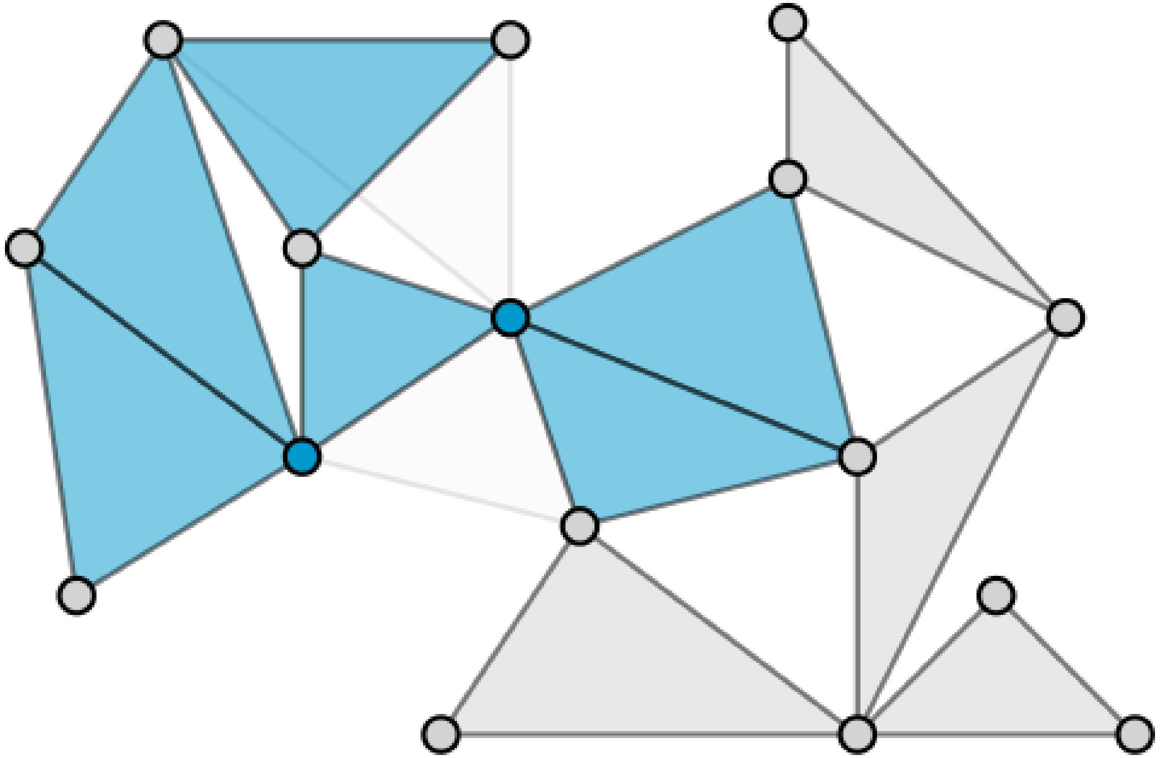
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

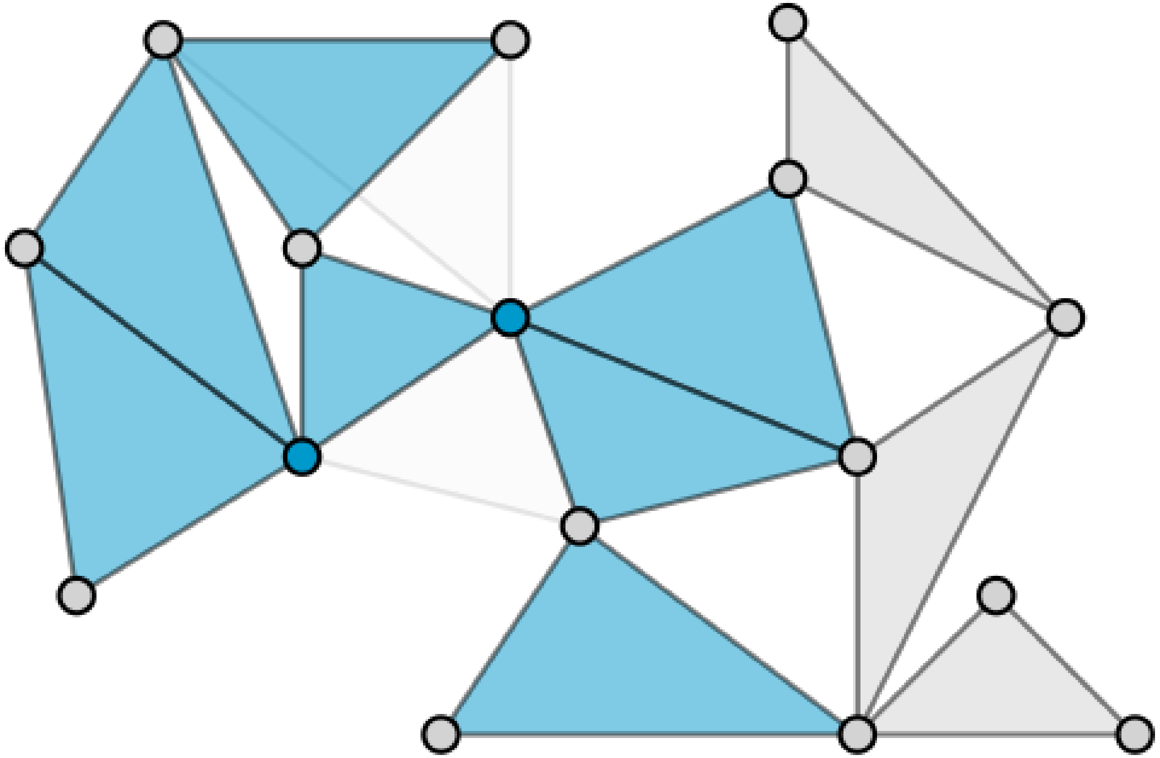
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

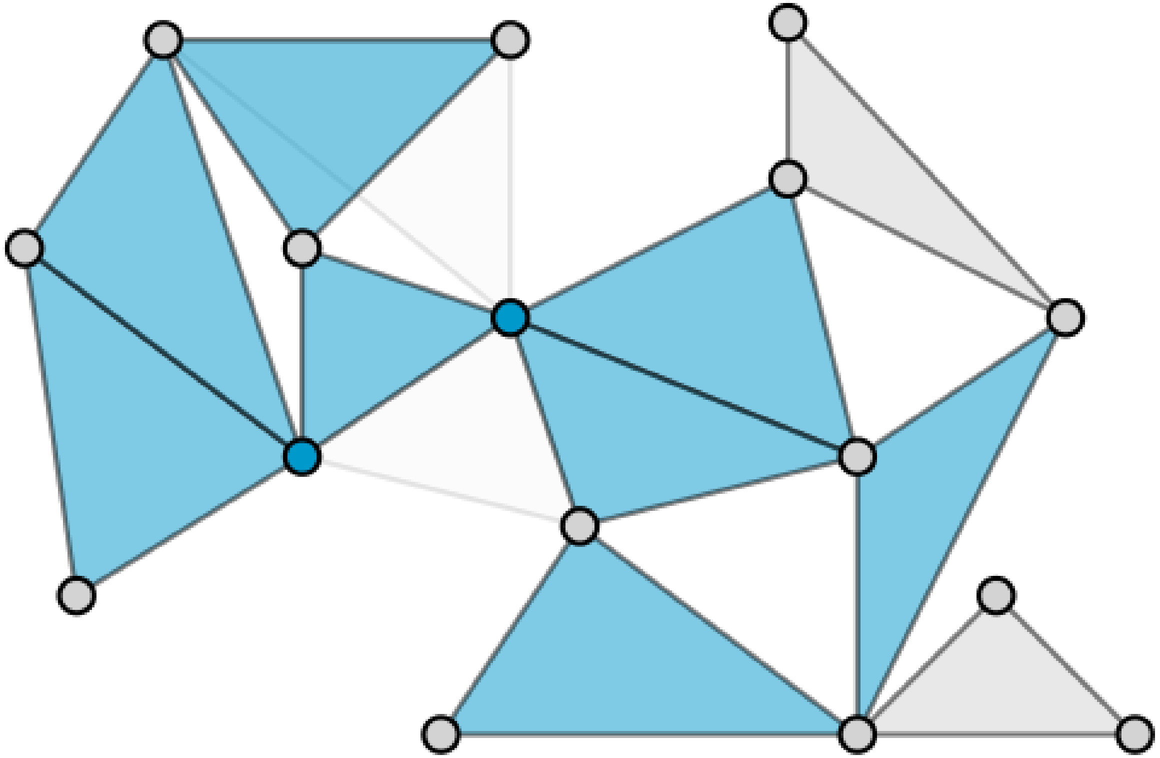
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

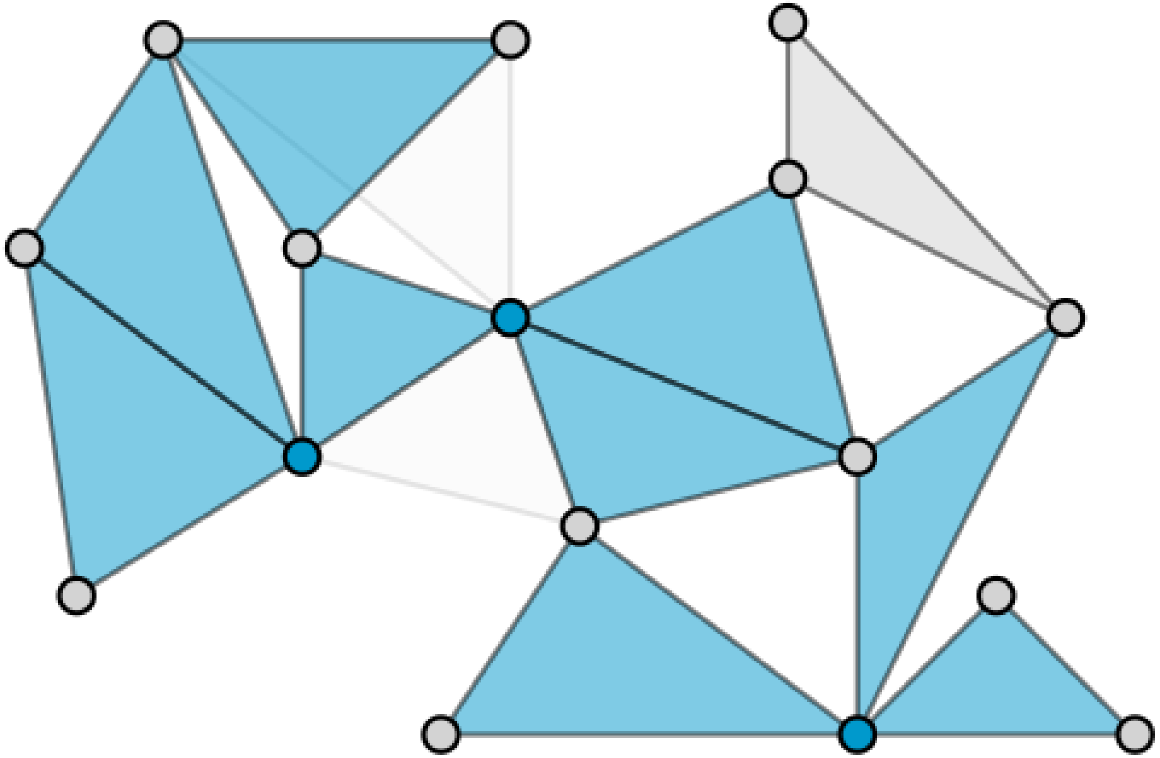
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

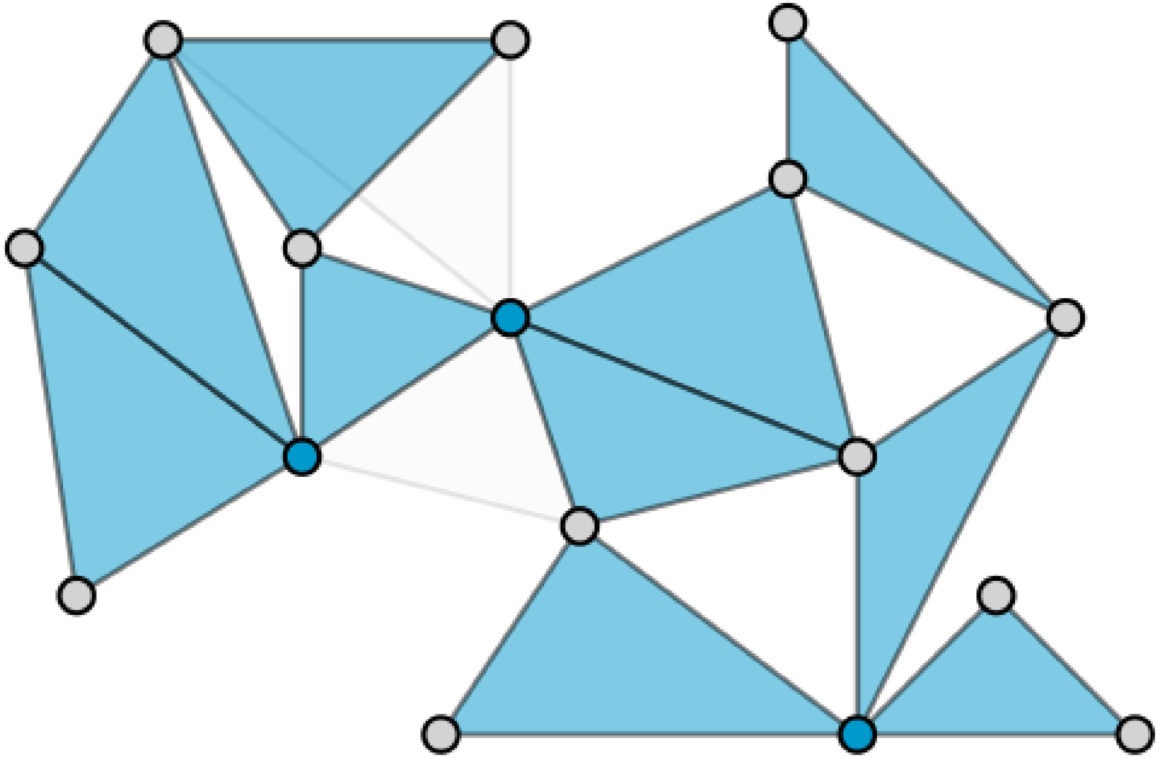
  set value to $\frac{L}{d}$

  mark tight nodes

  block their edges

for all blocked edges with value $< \frac{L}{d}$

  set value to 0



Basic Rounding

Sequential Greedy Factor-2-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

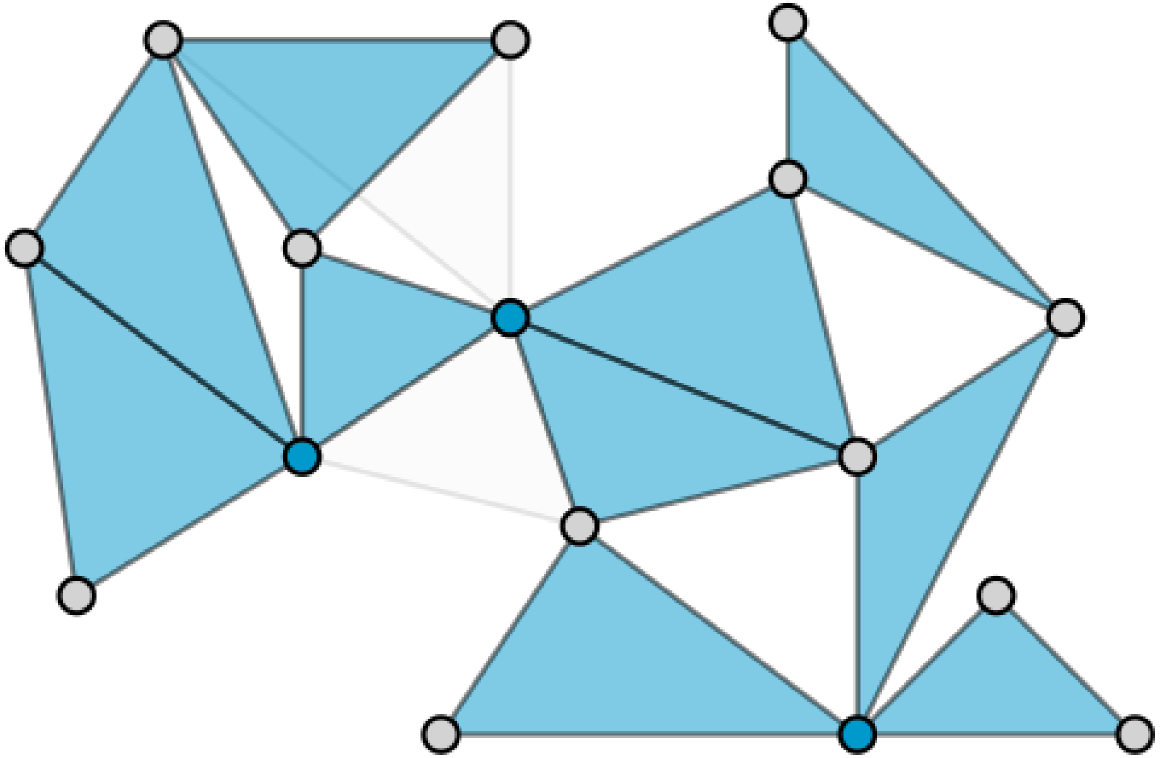
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



$O(r)$ loss

Basic Rounding

LOCAL Greedy Factor-L-Rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

$\frac{d}{2L}$ - Defective $O(L^2 r^2)$-Edge-Coloring

for each color class

  mark half-tight nodes

  block their edges

  set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

  set value to 0

Basic Rounding

LOCAL Greedy Factor-L-Rounding

from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

$\frac{d}{2L}$ - Defective $O(L^2 r^2)$-Edge-Coloring

for each color class

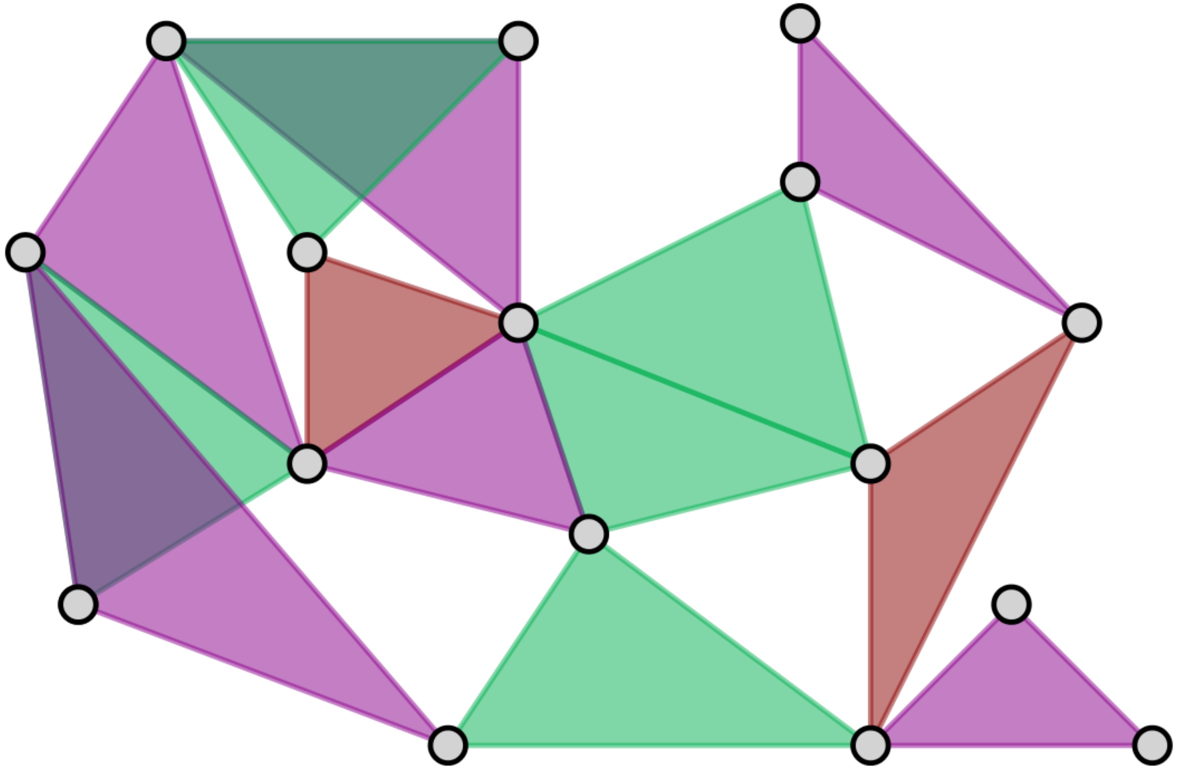
    mark half-tight nodes

    block their edges

    set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

    set value to 0



Basic Rounding

LOCAL Greedy Factor-L-Rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

$\frac{d}{2L}$ - Defective $O(L^2r^2)$-Edge-Coloring

for each color class

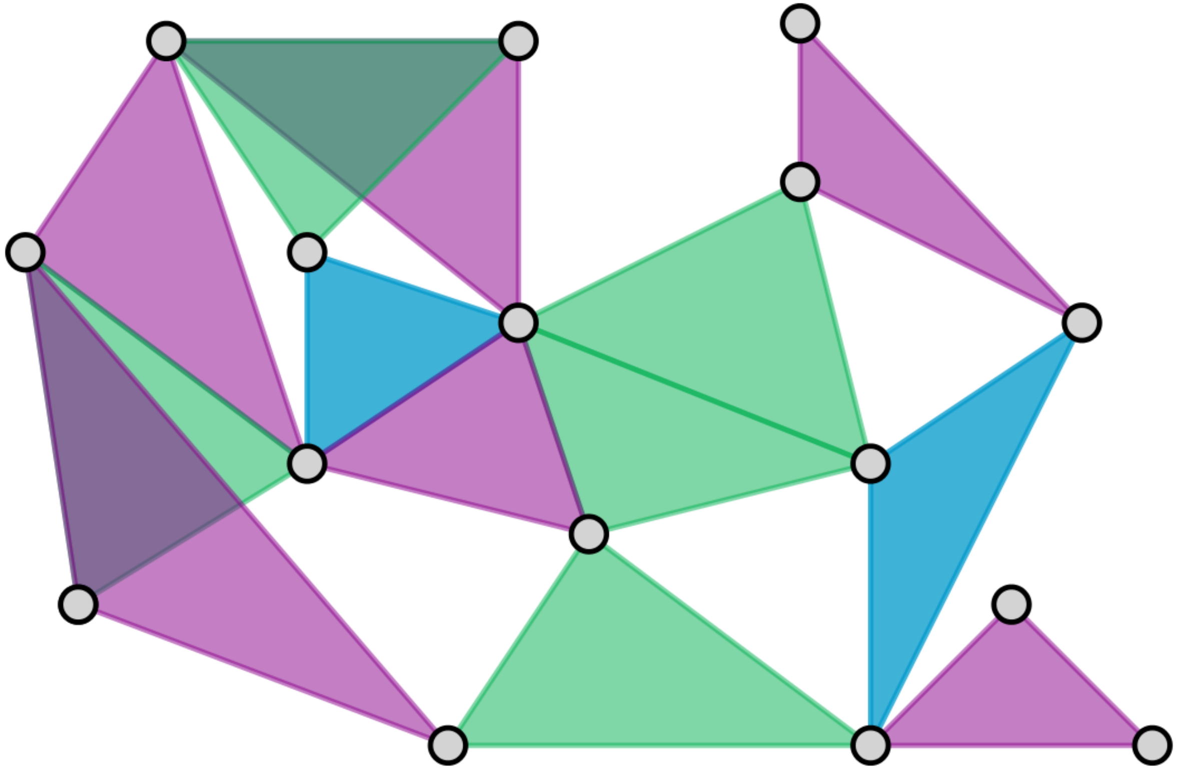
  mark half-tight nodes

  block their edges

  set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

  set value to 0



Basic Rounding

LOCAL Greedy Factor-L-Rounding

from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

$\frac{d}{2L}$ - Defective $O(L^2 r^2)$-Edge-Coloring

for each color class

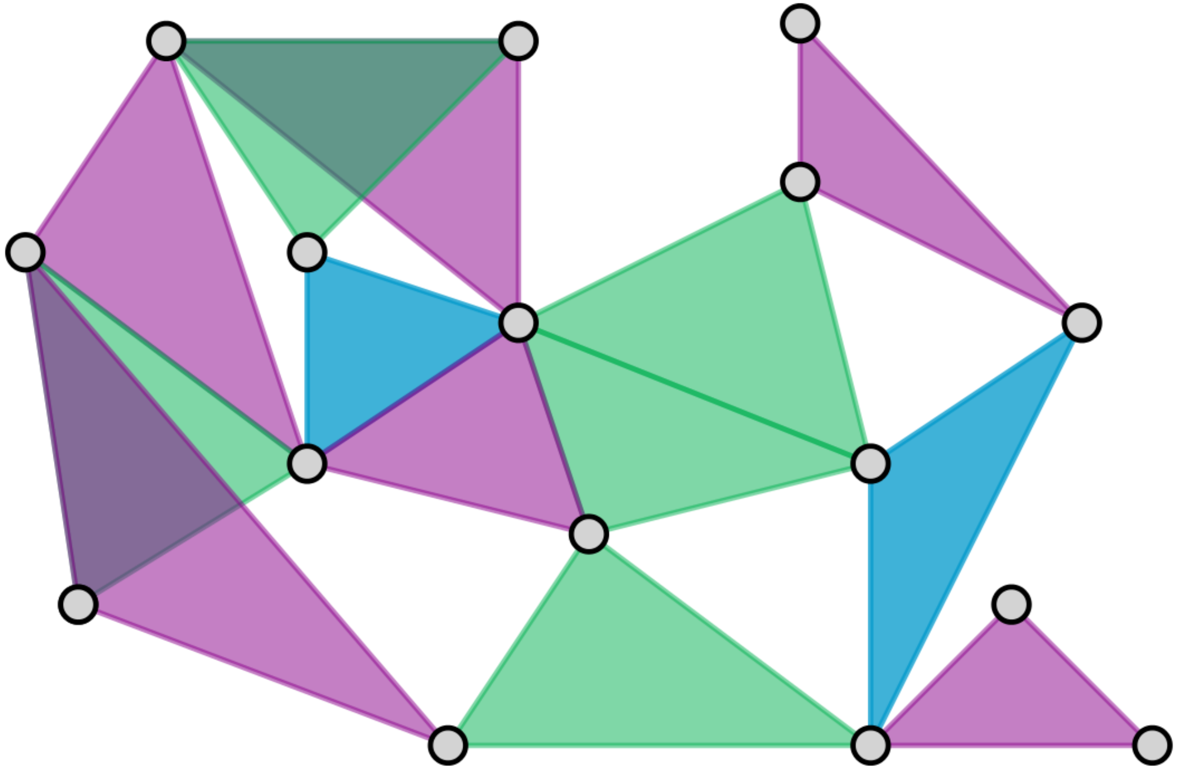
  mark half-tight nodes

  block their edges

  set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

  set value to 0



In each step, value of a node increased by at most $+\frac{d}{2L} \cdot \frac{L}{d} = \frac{1}{2}$

$O(\log \Delta + (L \cdot r)^2)$ rounds using Defective-Coloring Algorithm by Kuhn [SPAA’09]

Basic Rounding

LOCAL Greedy Factor-L-Rounding

from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

$\frac{d}{2L}$ - Defective $O(L^2 r^2)$-Edge-Coloring

for each color class

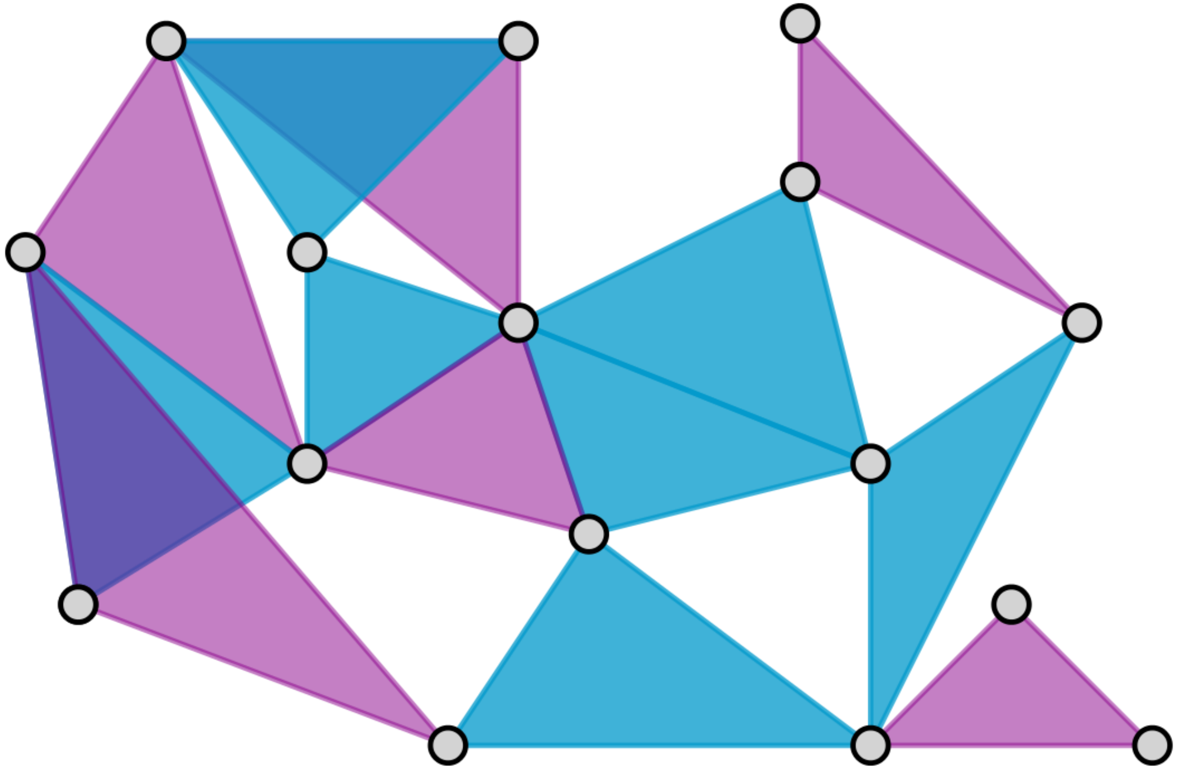
  mark half-tight nodes

  block their edges

  set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

  set value to 0



In each step, value of a node increased by at most $+\frac{d}{2L} \cdot \frac{L}{d} = \frac{1}{2}$

$O(\log \Delta + (L \cdot r)^2)$ rounds using Defective-Coloring Algorithm by Kuhn [SPAA'09]

Basic Rounding

LOCAL Greedy Factor-L-Rounding

from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

$\frac{d}{2L}$ - Defective $O(L^2 r^2)$-Edge-Coloring

for each color class

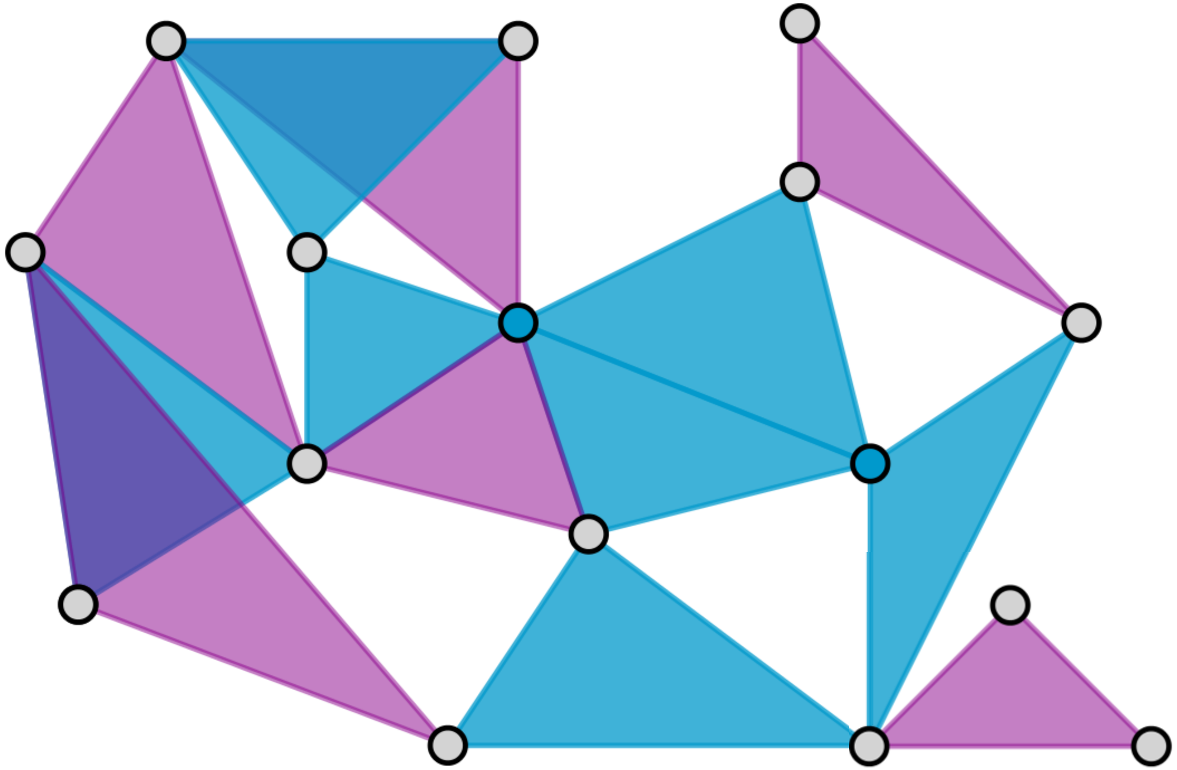
  mark half-tight nodes

  block their edges

  set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

  set value to 0



In each step, value of a node increased by at most $+\frac{d}{2L} \cdot \frac{L}{d} = \frac{1}{2}$

$O(\log \Delta + (L \cdot r)^2)$ rounds using Defective-Coloring Algorithm by Kuhn [SPAA'09]

Basic Rounding

LOCAL Greedy Factor-L-Rounding

from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

$\frac{d}{2L}$ - Defective $O(L^2 r^2)$-Edge-Coloring

for each color class

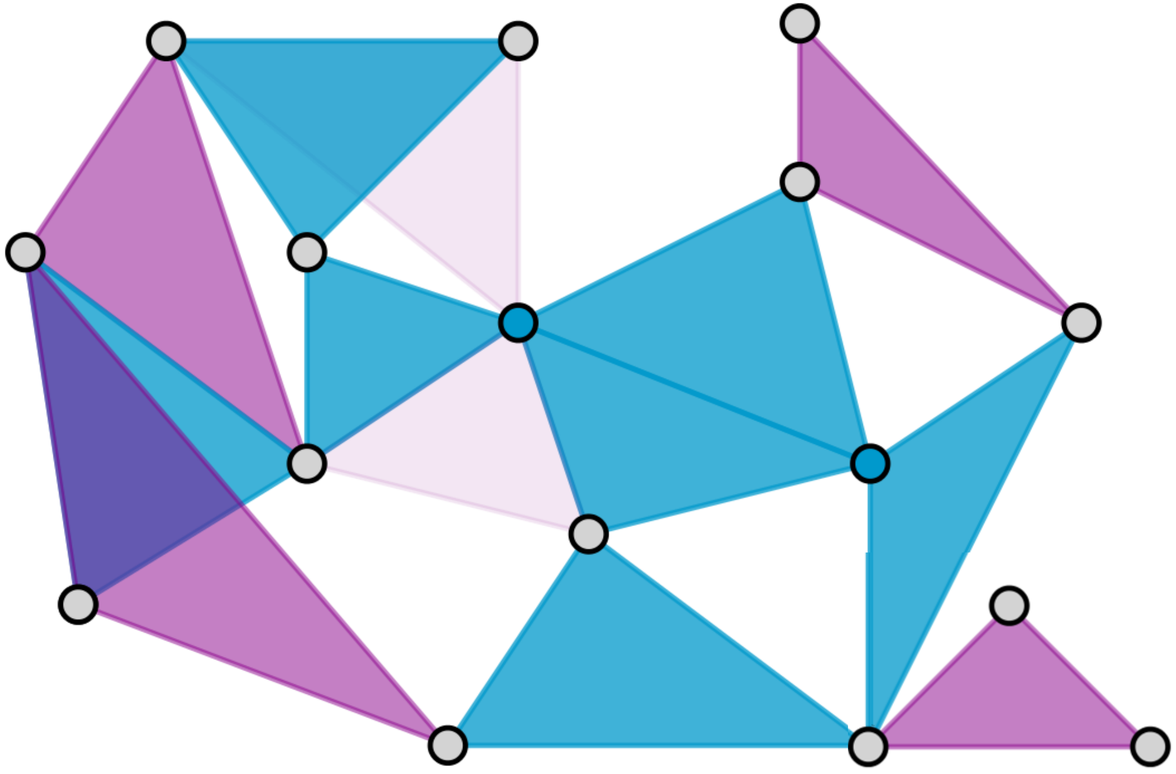
  mark half-tight nodes

  block their edges

  set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

  set value to 0



In each step, value of a node increased by at most $+\frac{d}{2L} \cdot \frac{L}{d} = \frac{1}{2}$

$O(\log \Delta + (L \cdot r)^2)$ rounds using Defective-Coloring Algorithm by Kuhn [SPAA'09]

Basic Rounding

LOCAL Greedy Factor-L-Rounding

from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

$\frac{d}{2L}$ - Defective $O(L^2 r^2)$-Edge-Coloring

for each color class

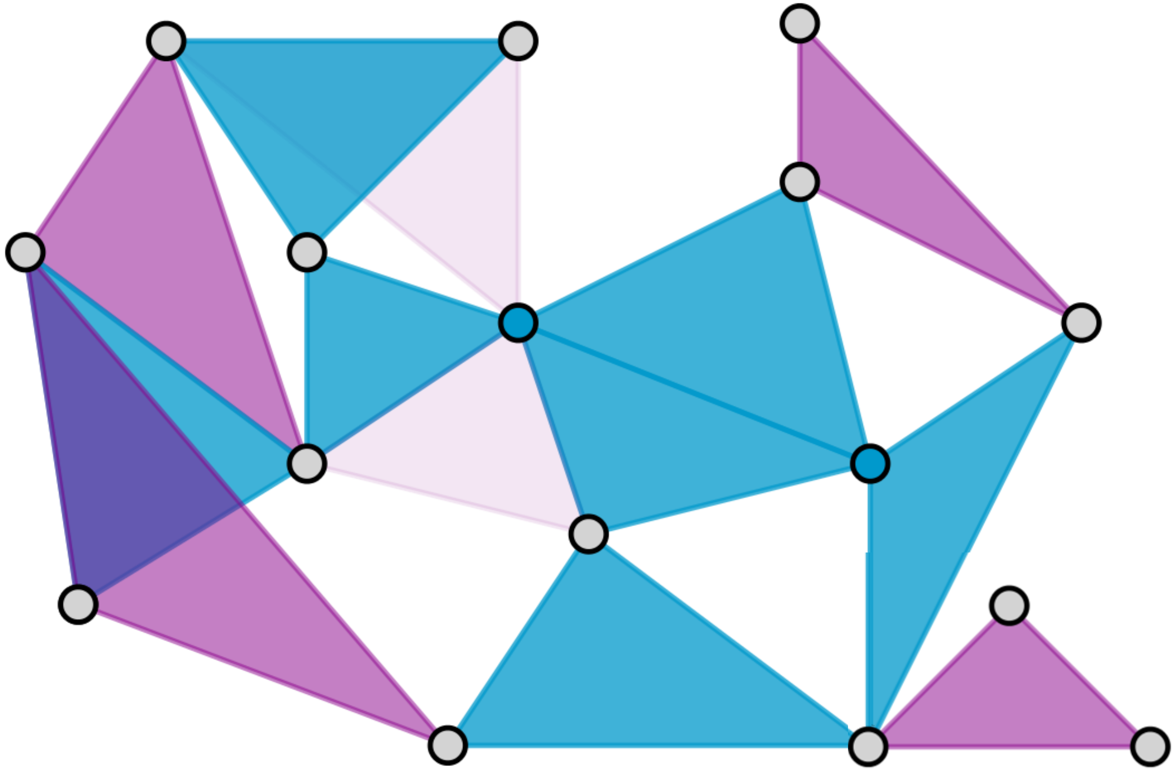
    mark half-tight nodes

    block their edges

    set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

    set value to 0



In each step, value of a node increased by at most $+\frac{d}{2L} \cdot \frac{L}{d} = \frac{1}{2}$

$O(\log \Delta + (L \cdot r)^2)$ rounds using Defective-Coloring Algorithm by Kuhn [SPAA'09]

Basic Rounding

Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

LOCAL Greedy Factor-L-Rounding

from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

$\frac{d}{2L}$ - Defective $O(L^2 r^2)$-Edge-Coloring

for each color class

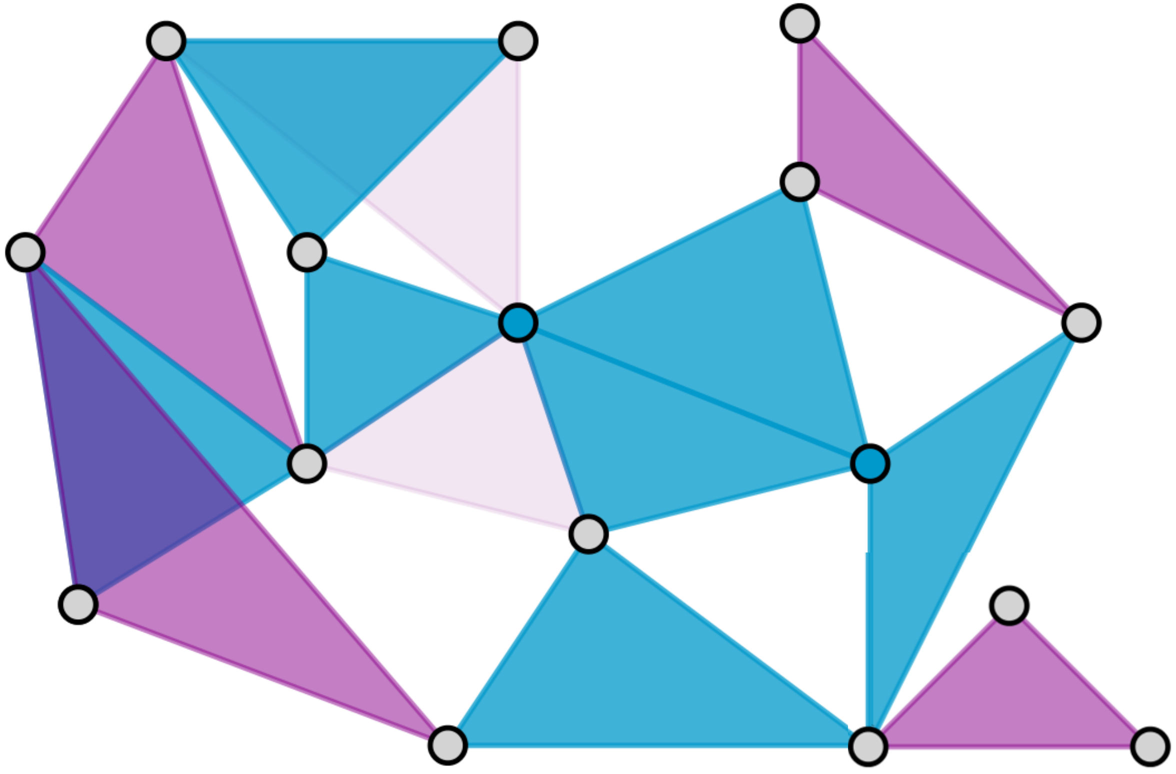
 mark half-tight nodes

 block their edges

 set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

 set value to 0



In each step, value of a node increased by at most $+\frac{d}{2L} \cdot \frac{L}{d} = \frac{1}{2}$

$O(\log \Delta + (L \cdot r)^2)$ rounds using Defective-Coloring Algorithm by Kuhn [SPAA'09]

Rounding

Basic Rounding:

Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

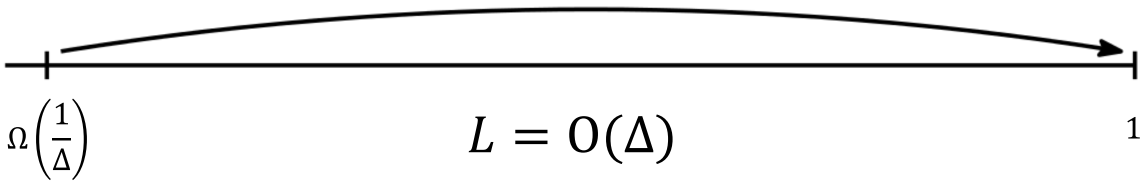
Rounding

Basic Rounding:
Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds



Rounding

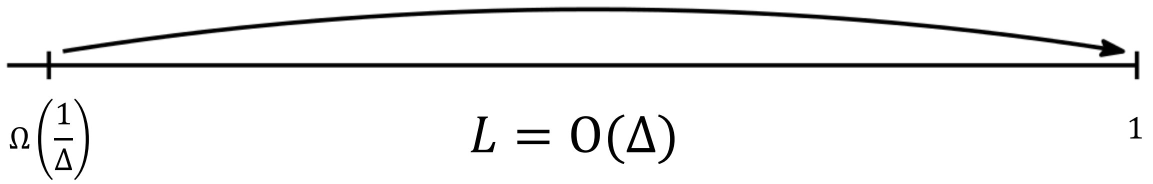
Basic Rounding:

Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds



too slow!

Rounding

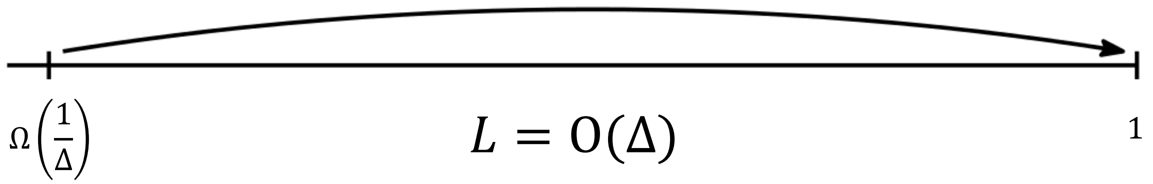
Basic Rounding:

Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

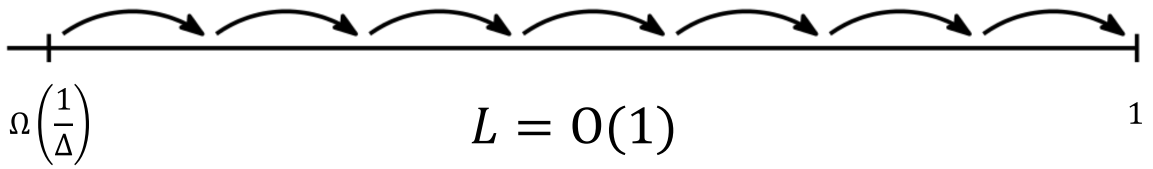


too slow!

Iterative Basic Rounding

$r^{\Theta(\log \Delta)}$ loss

$O\left(\log \Delta \cdot (\log \Delta + r^2)\right)$ rounds



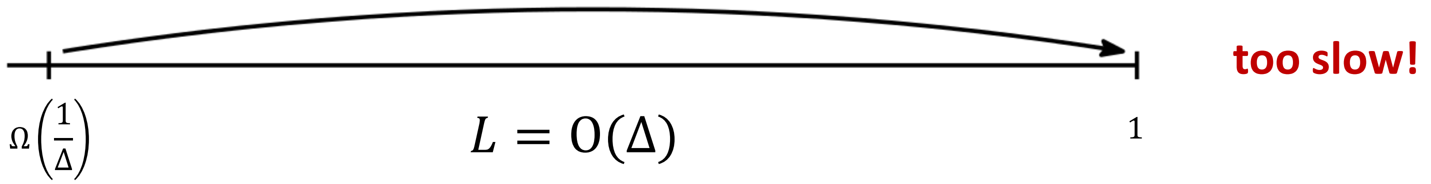
Rounding

Basic Rounding:
Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

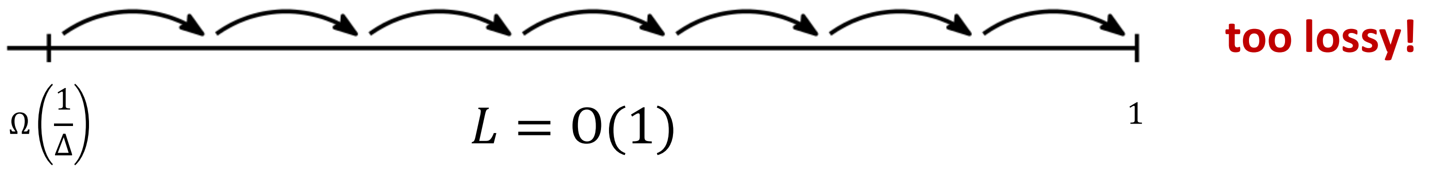
$O(\log \Delta + (\Delta \cdot r)^2)$ rounds



Iterative Basic Rounding

$r^{\Theta(\log \Delta)}$ loss

$O\left(\log \Delta \cdot (\log \Delta + r^2)\right)$ rounds



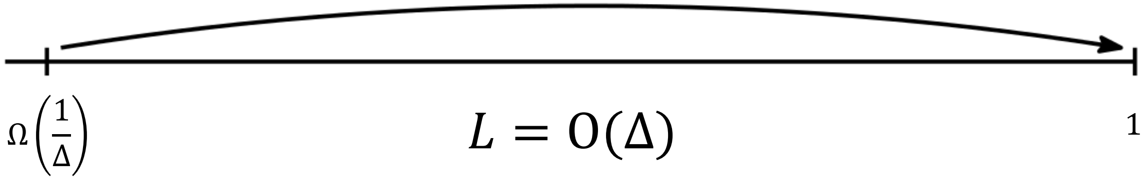
Rounding

Basic Rounding:
Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

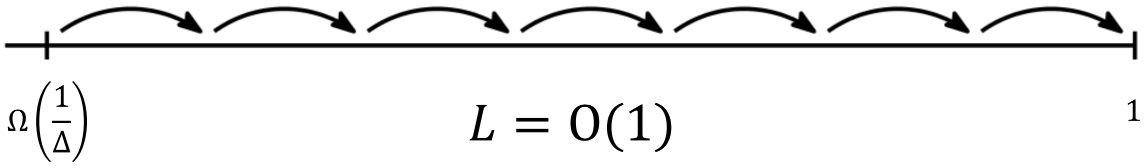


too slow!

Iterative Basic Rounding

$r^{\Theta(\log \Delta)}$ loss

$O\big(\log \Delta \cdot (\log \Delta + r^2)\big)$ rounds



too lossy!

Challenge 2:
repeated application

Rounding

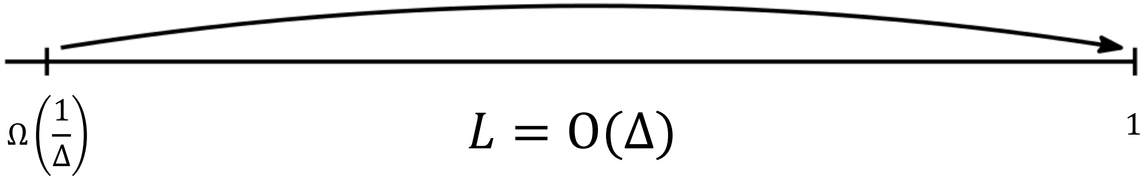
Basic Rounding:

Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

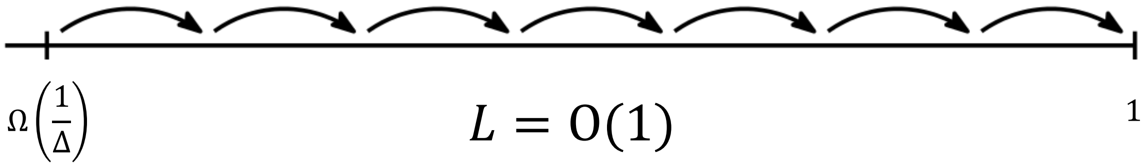


too slow!

Iterative Basic Rounding

$r^{\Theta(\log \Delta)}$ loss

$O\big(\log \Delta \cdot (\log \Delta + r^2)\big)$ rounds



too lossy!

Recursive Rounding

&

Iterative Refilling

Rounding

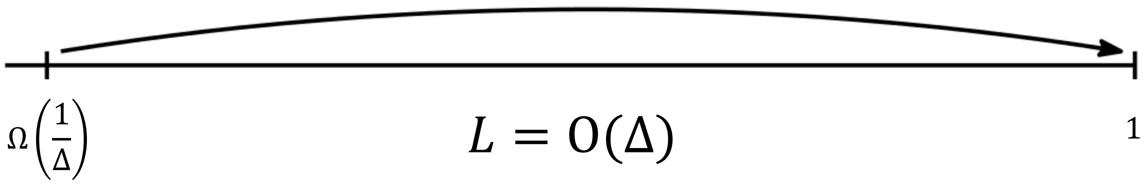
Basic Rounding:

Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

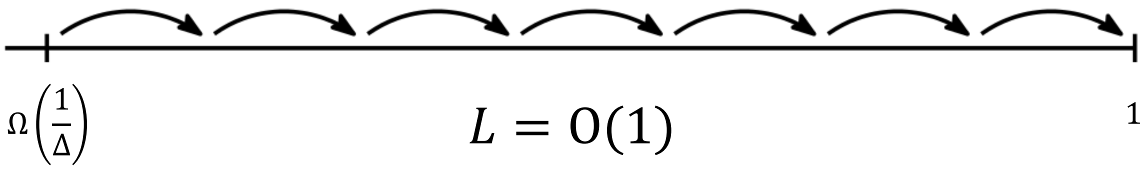


too slow!

Iterative Basic Rounding

$r^{\Theta(\log \Delta)}$ loss

$O\big(\log \Delta \cdot (\log \Delta + r^2)\big)$ rounds

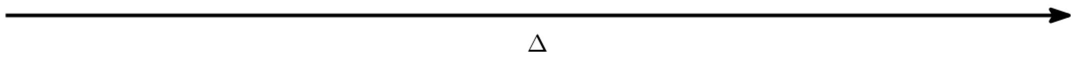


too lossy!

Recursive Rounding

&

Iterative Refilling



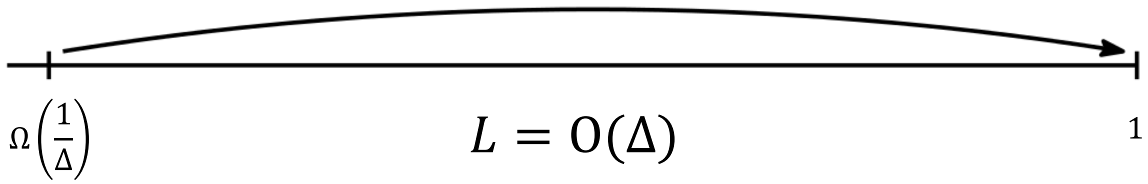
Rounding

Basic Rounding:
Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

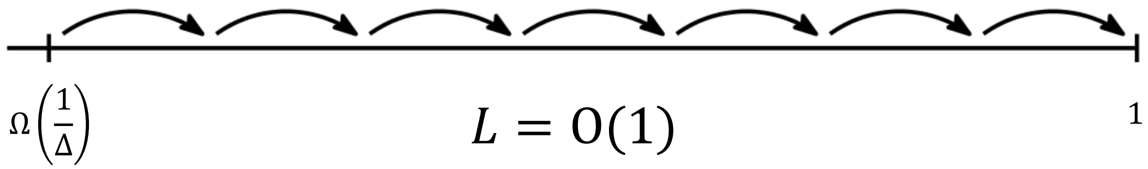


too slow!

Iterative Basic Rounding

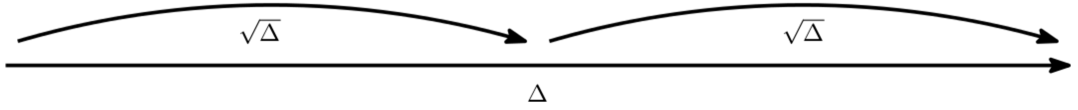
$r^{\Theta(\log \Delta)}$ loss

$O\big(\log \Delta \cdot (\log \Delta + r^2)\big)$ rounds



too lossy!

Recursive Rounding
&
Iterative Refilling



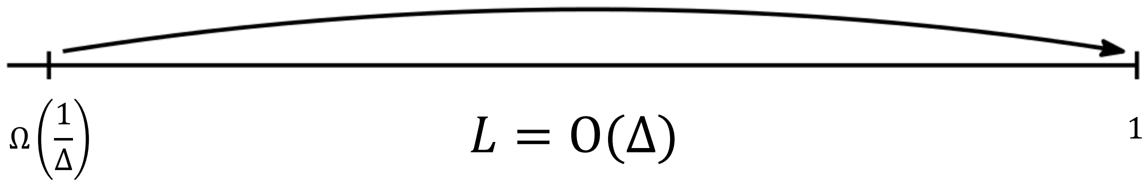
Rounding

Basic Rounding:
Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

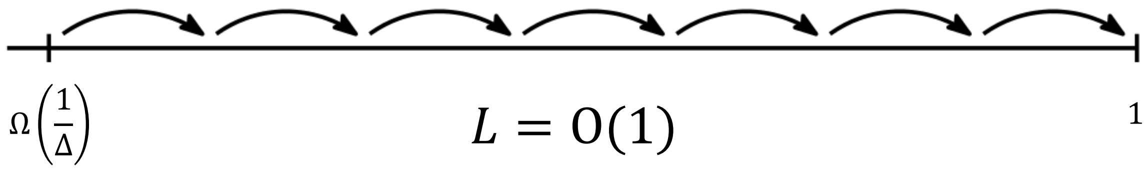


too slow!

Iterative Basic Rounding

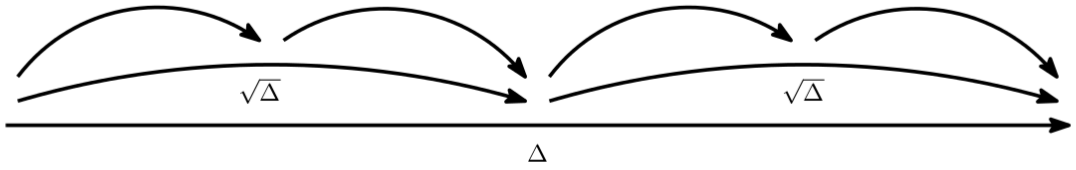
$r^{\Theta(\log \Delta)}$ loss

$O\left(\log \Delta \cdot (\log \Delta + r^2)\right)$ rounds



too lossy!

Recursive Rounding
&
Iterative Refilling



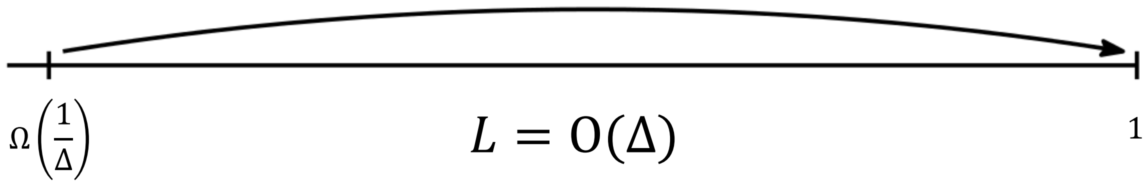
Rounding

Basic Rounding:
Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

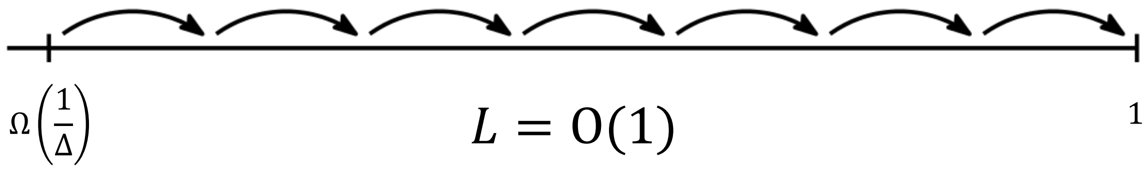


too slow!

Iterative Basic Rounding

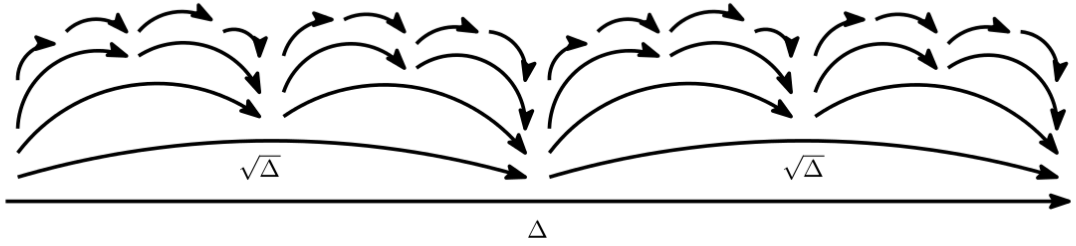
$r^{\Theta(\log \Delta)}$ loss

$O\big(\log \Delta \cdot (\log \Delta + r^2)\big)$ rounds



too lossy!

Recursive Rounding
&
Iterative Refilling



Rounding

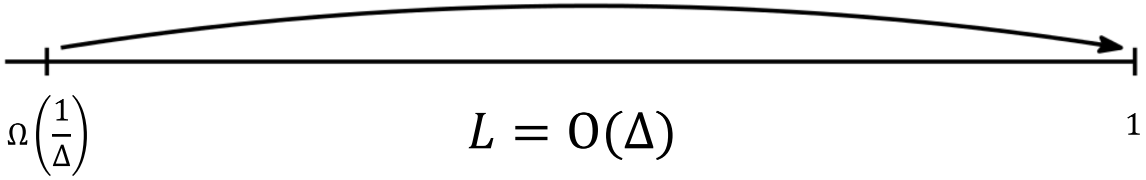
Basic Rounding:

Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

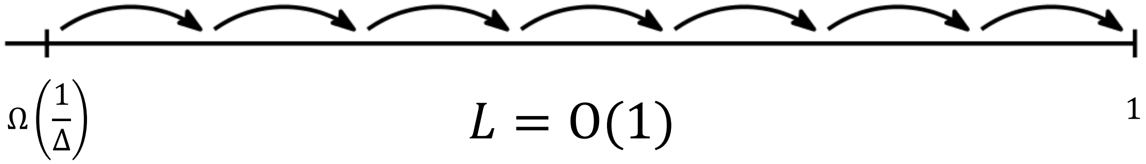


too slow!

Iterative Basic Rounding

$r^{\Theta(\log \Delta)}$ loss

$O\big(\log \Delta \cdot (\log \Delta + r^2)\big)$ rounds

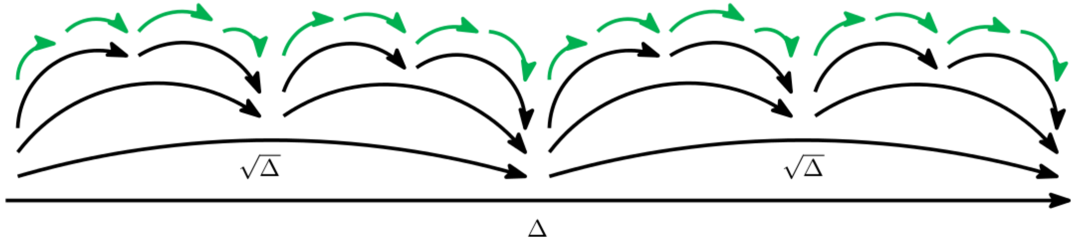


too lossy!

Recursive Rounding

&

Iterative Refilling



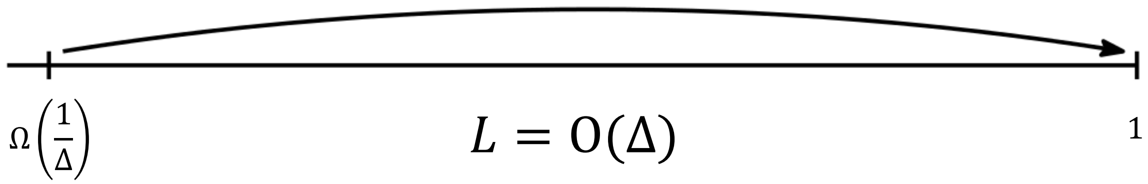
Rounding

Basic Rounding:
Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

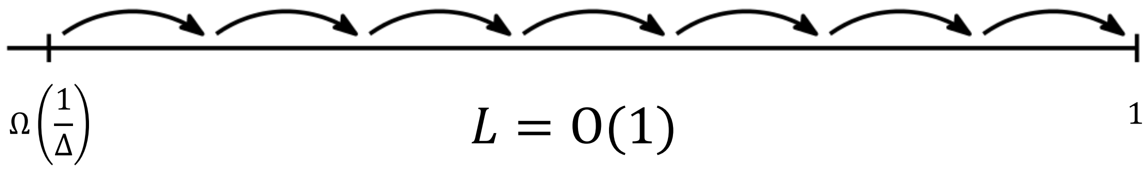


too slow!

Iterative Basic Rounding

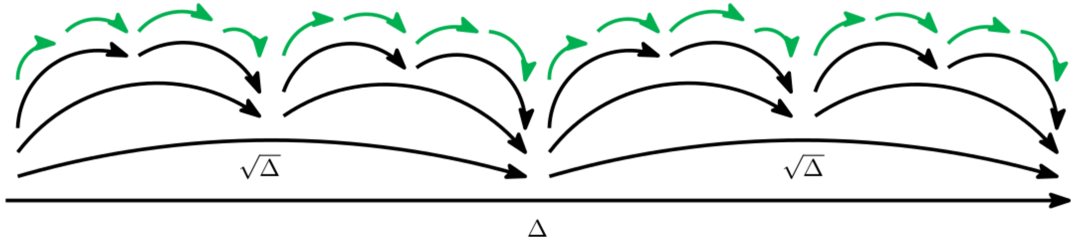
$r^{\Theta(\log \Delta)}$ loss

$O\left(\log \Delta \cdot (\log \Delta + r^2)\right)$ rounds



too lossy!

Recursive Rounding
&
Iterative Refilling



Basic Rounding

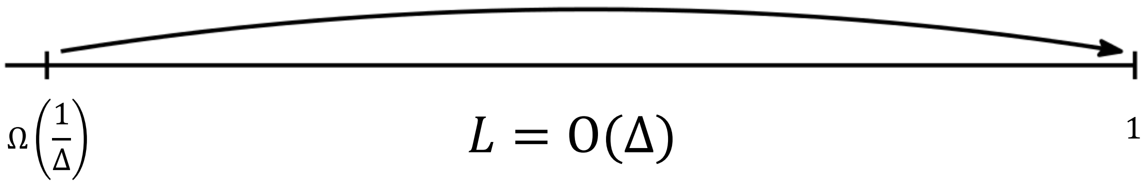
Rounding

Basic Rounding:
Factor-L-Rounding in $O(\log \Delta + (L \cdot r)^2)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O(\log \Delta + (\Delta \cdot r)^2)$ rounds

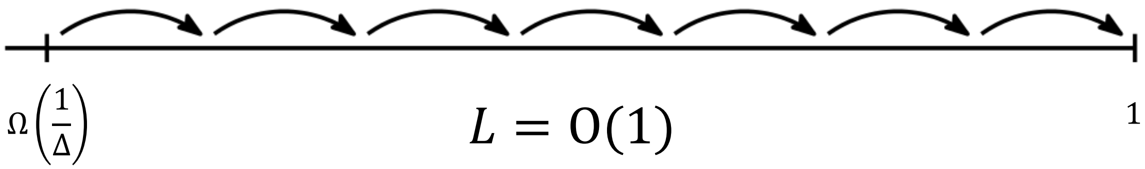


too slow!

Iterative Basic Rounding

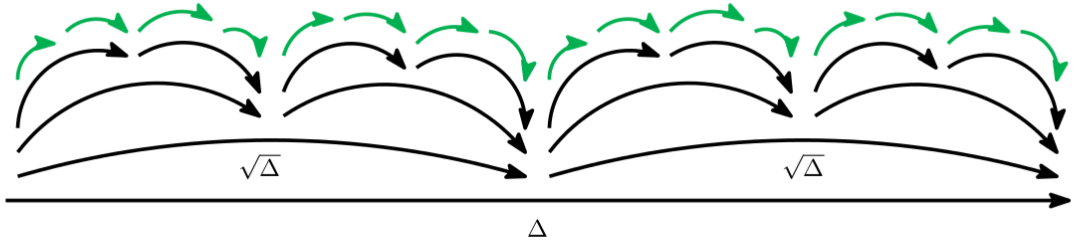
$r^{\Theta(\log \Delta)}$ loss

$O\left(\log \Delta \cdot (\log \Delta + r^2)\right)$ rounds



too lossy!

Recursive Rounding
&
Iterative Refilling

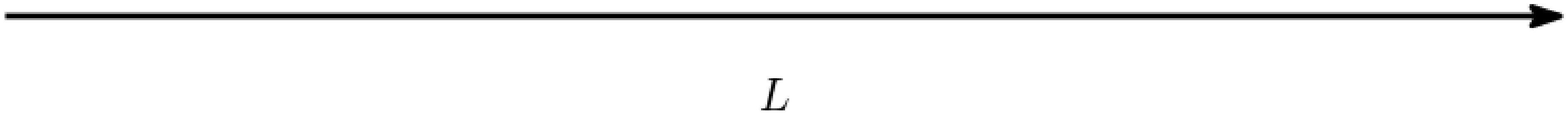


Basic Rounding

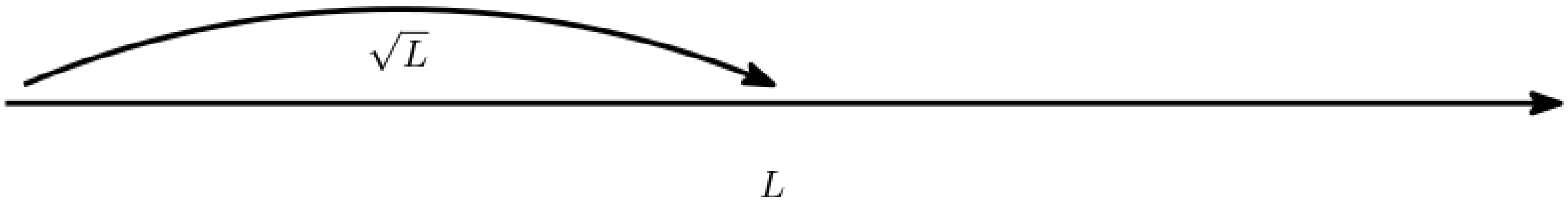
Invariant: $O(r)$ loss, kept using refilling iterations

Recursive Round

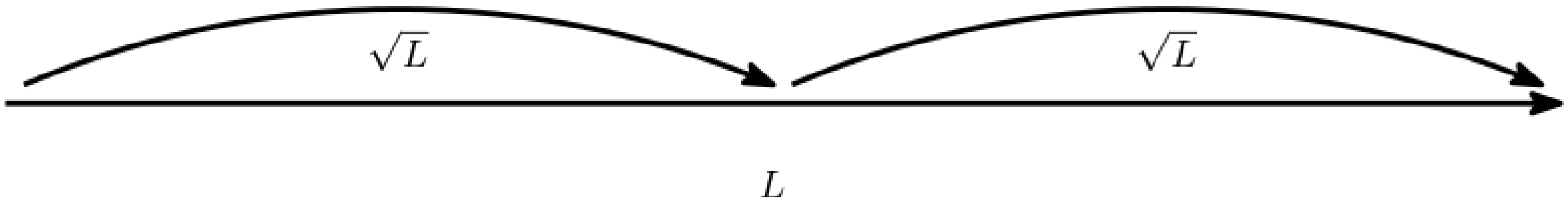
Recursive Round



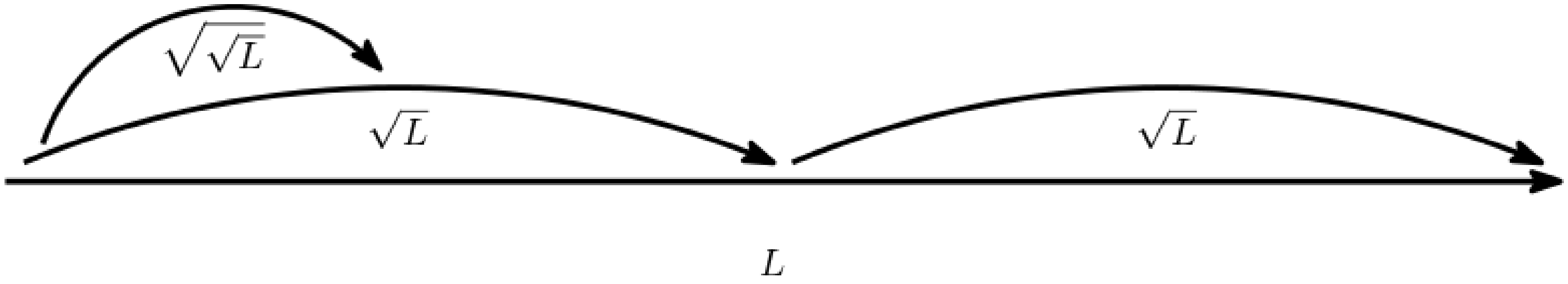
Recursive Round



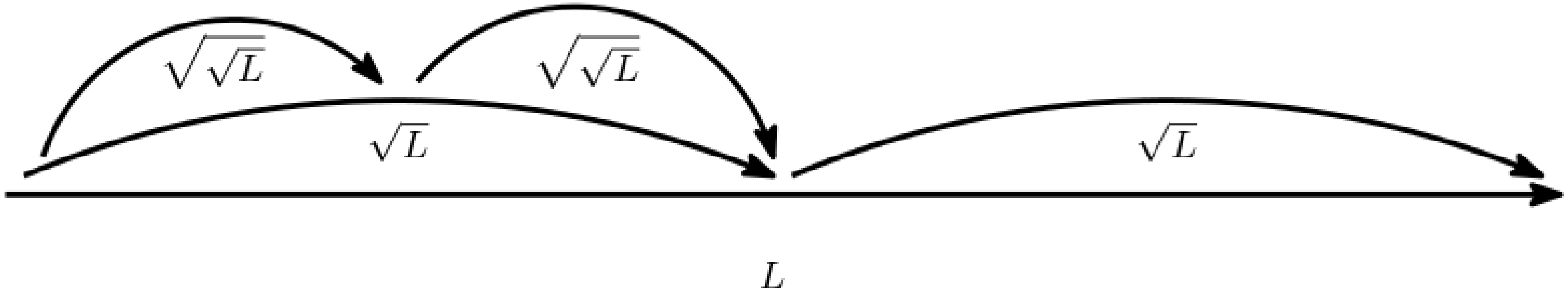
Recursive Round



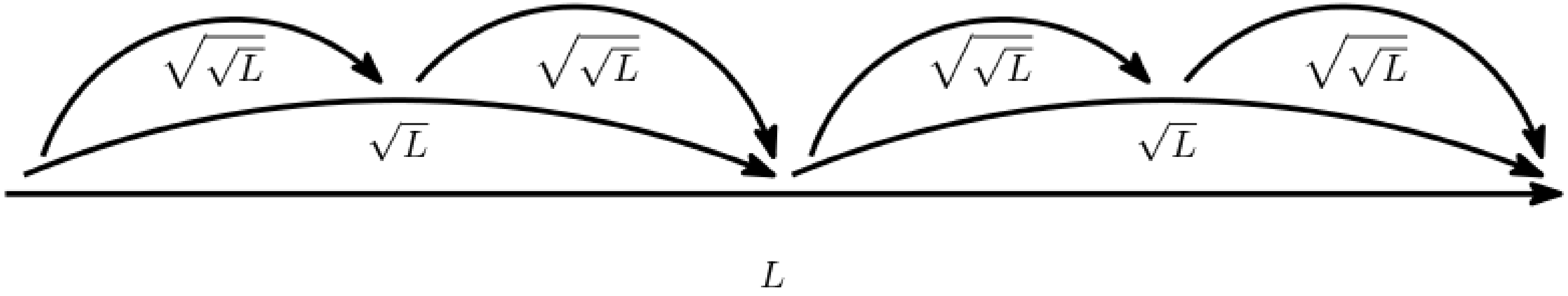
Recursive Round



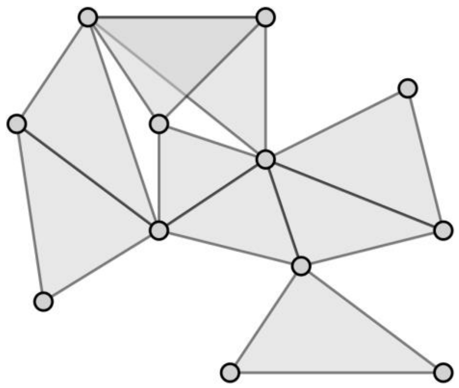
Recursive Round



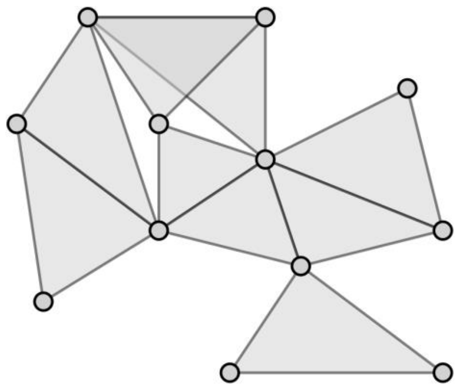
Recursive Round



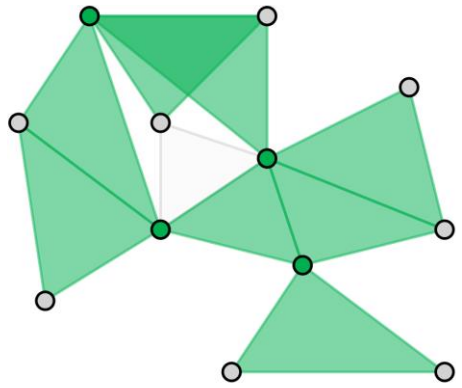
Recursive Round



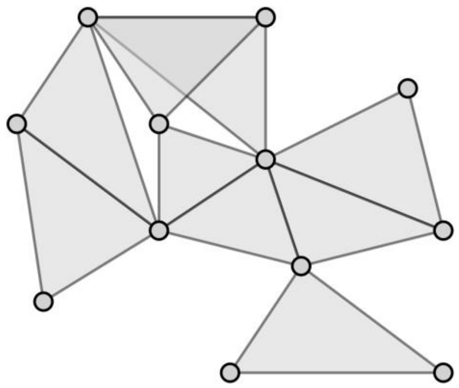
Recursive Round



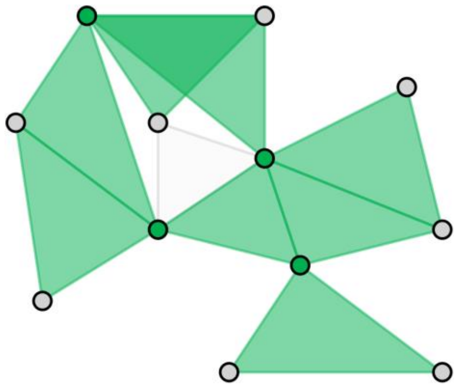
$\mathbf{round}\left(\sqrt{L}\right)$
O(r) loss



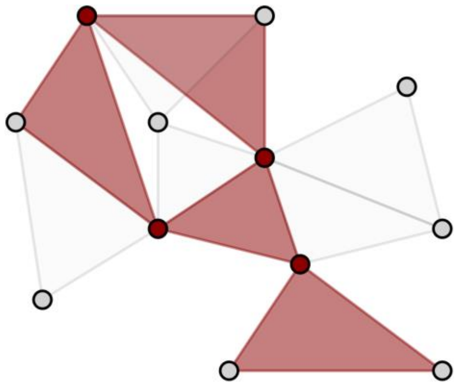
Recursive Round



round$\left(\sqrt{L}\right)$
O(r) loss

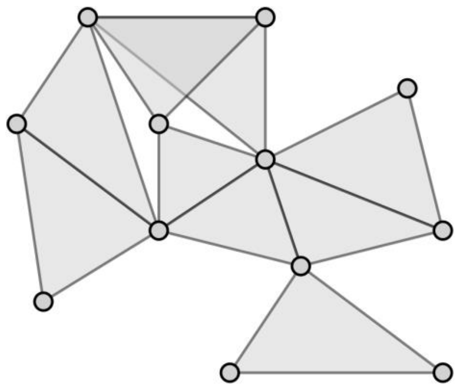


round$\left(\sqrt{L}\right)$
O(r) loss

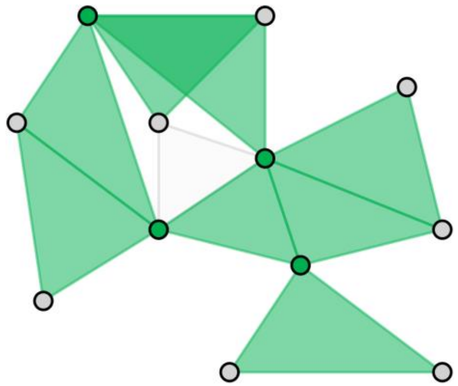


Recursive Round

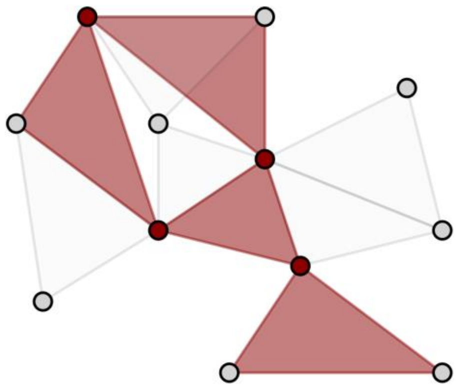
round'(L) with O(r^2) loss



round$(\sqrt{L})$
O(r) loss

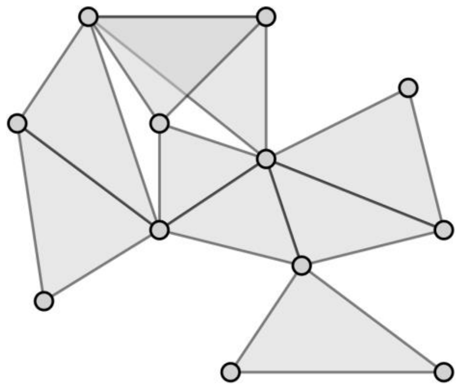


round$(\sqrt{L})$
O(r) loss

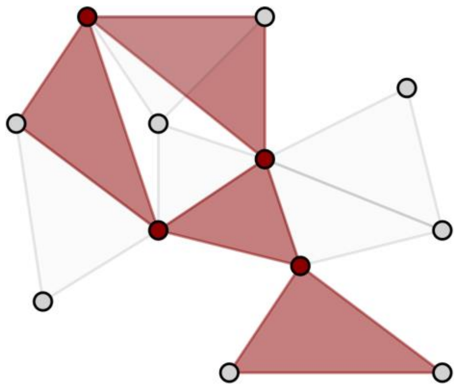


Iterative Refill

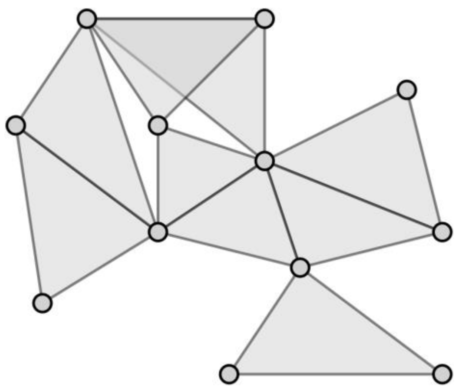
Iterative Refill



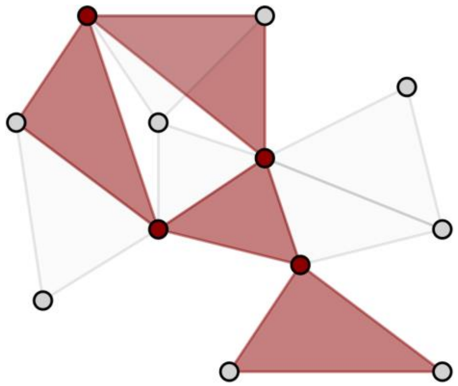
round'(L)
$O(r^2)$ loss



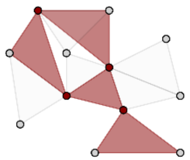
Iterative Refill

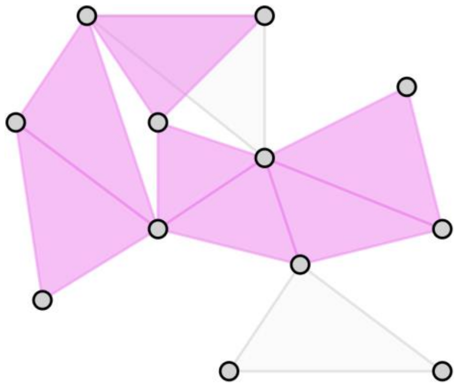


round’(L)
O(r^2) loss

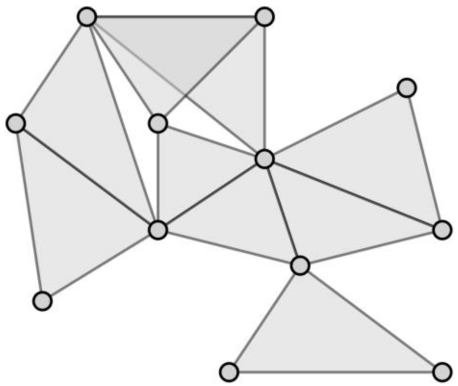


subtract

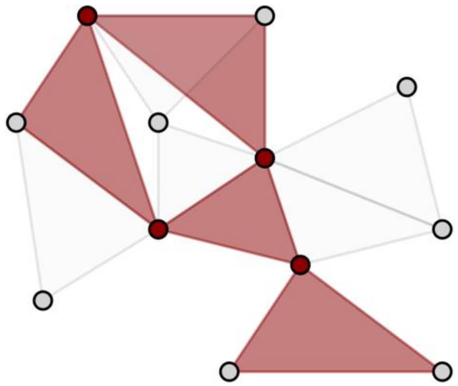




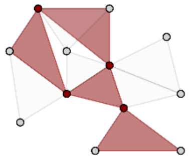
Iterative Refill

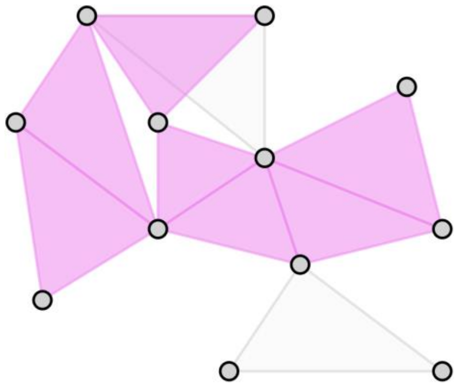


round’(L)
O(r^2) loss

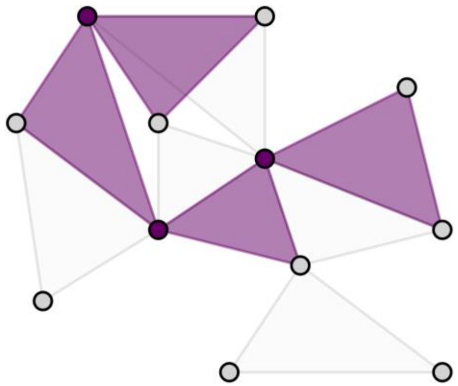


subtract

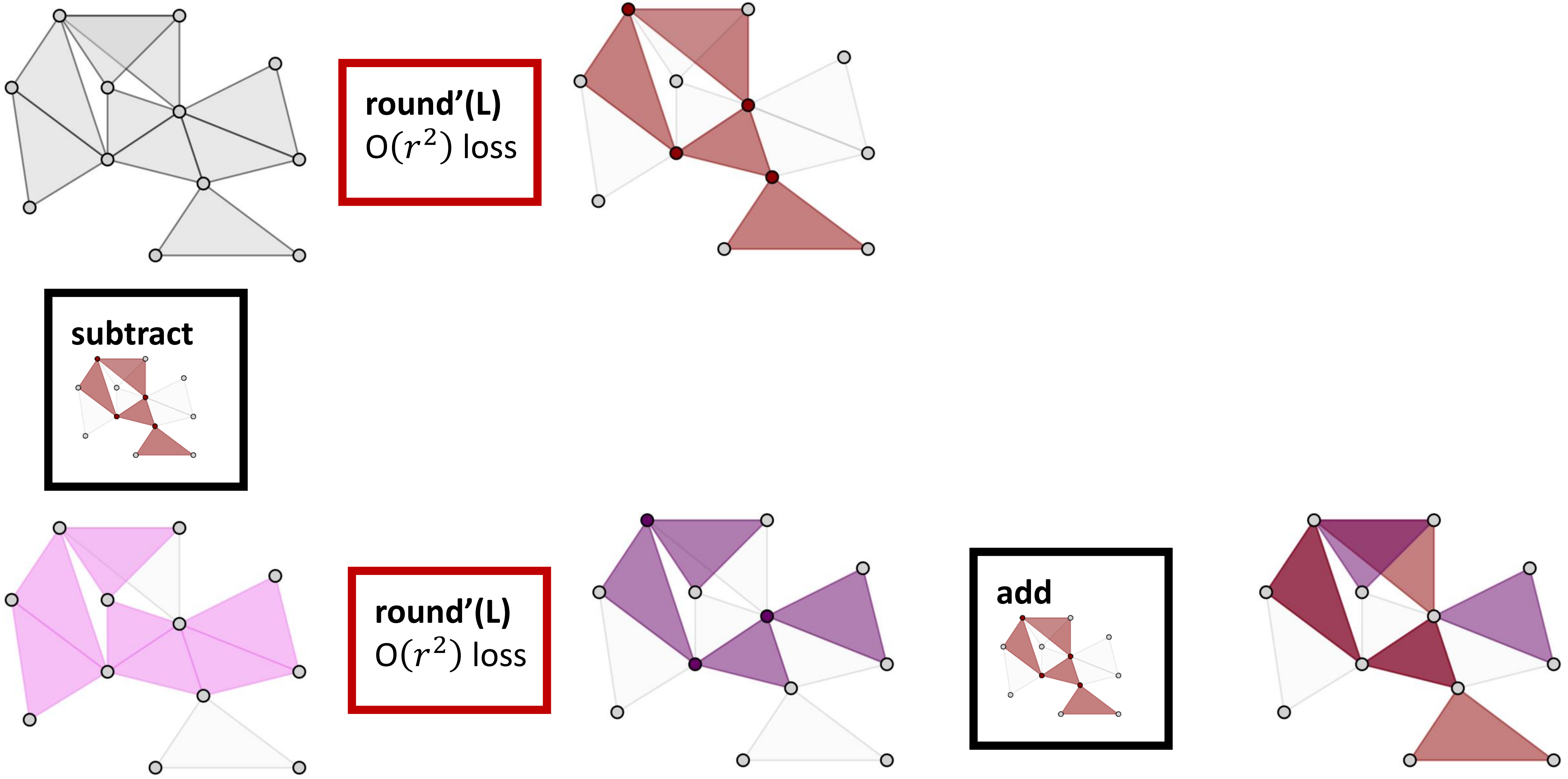




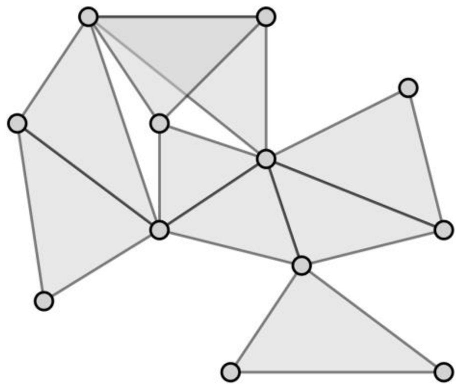
round’(L)
O(r^2) loss



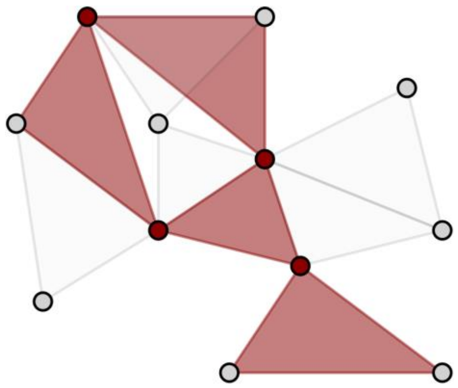
Iterative Refill



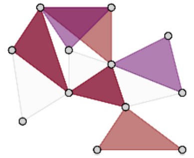
Iterative Refill

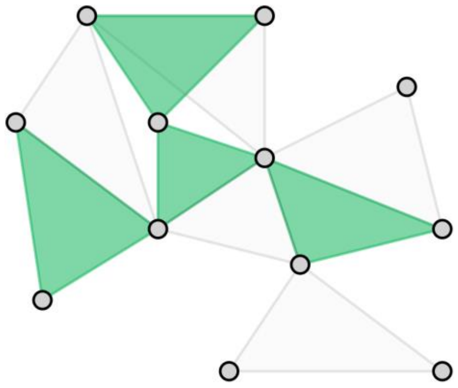


round'(L)
O(r^2) loss

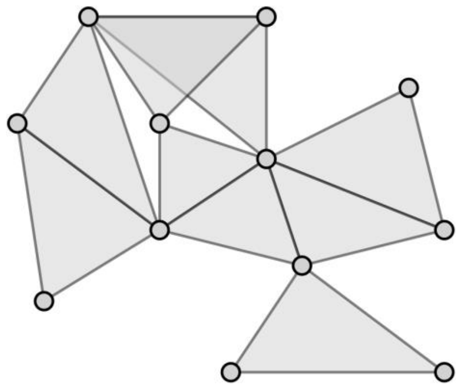


subtract

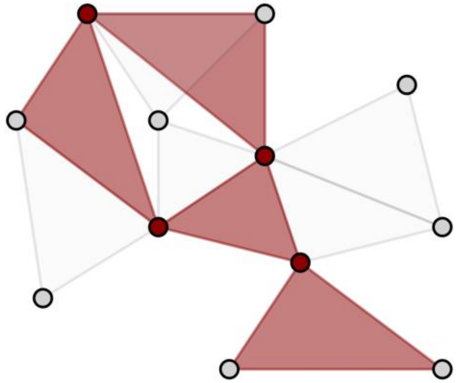




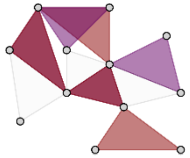
Iterative Refill

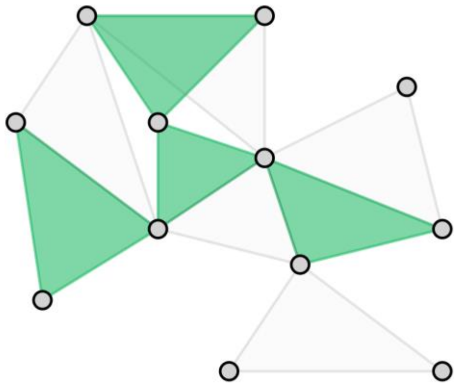


round’(L)
O(r^2) loss

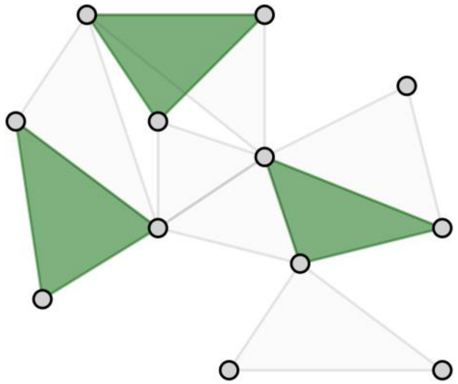


subtract

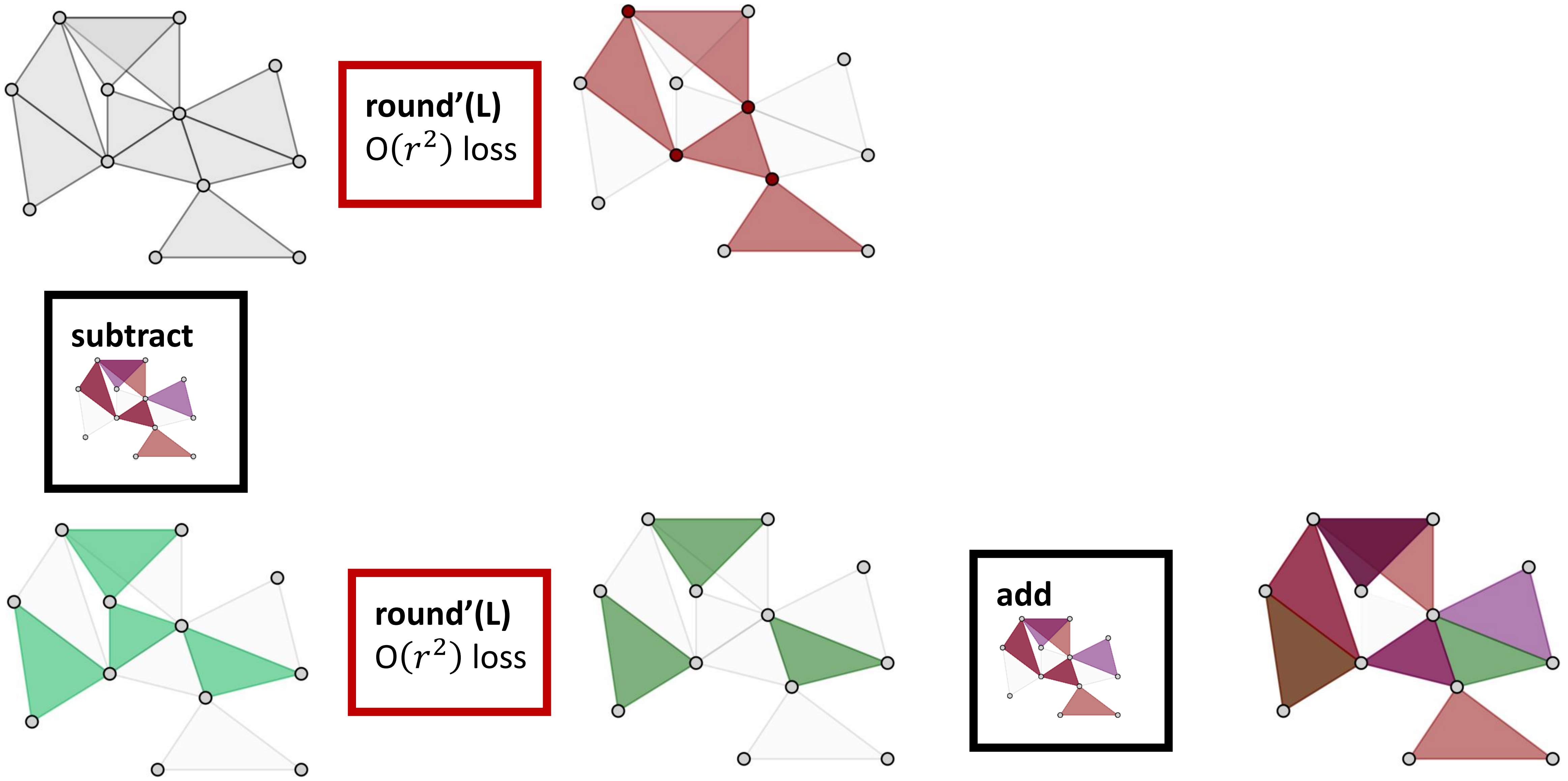




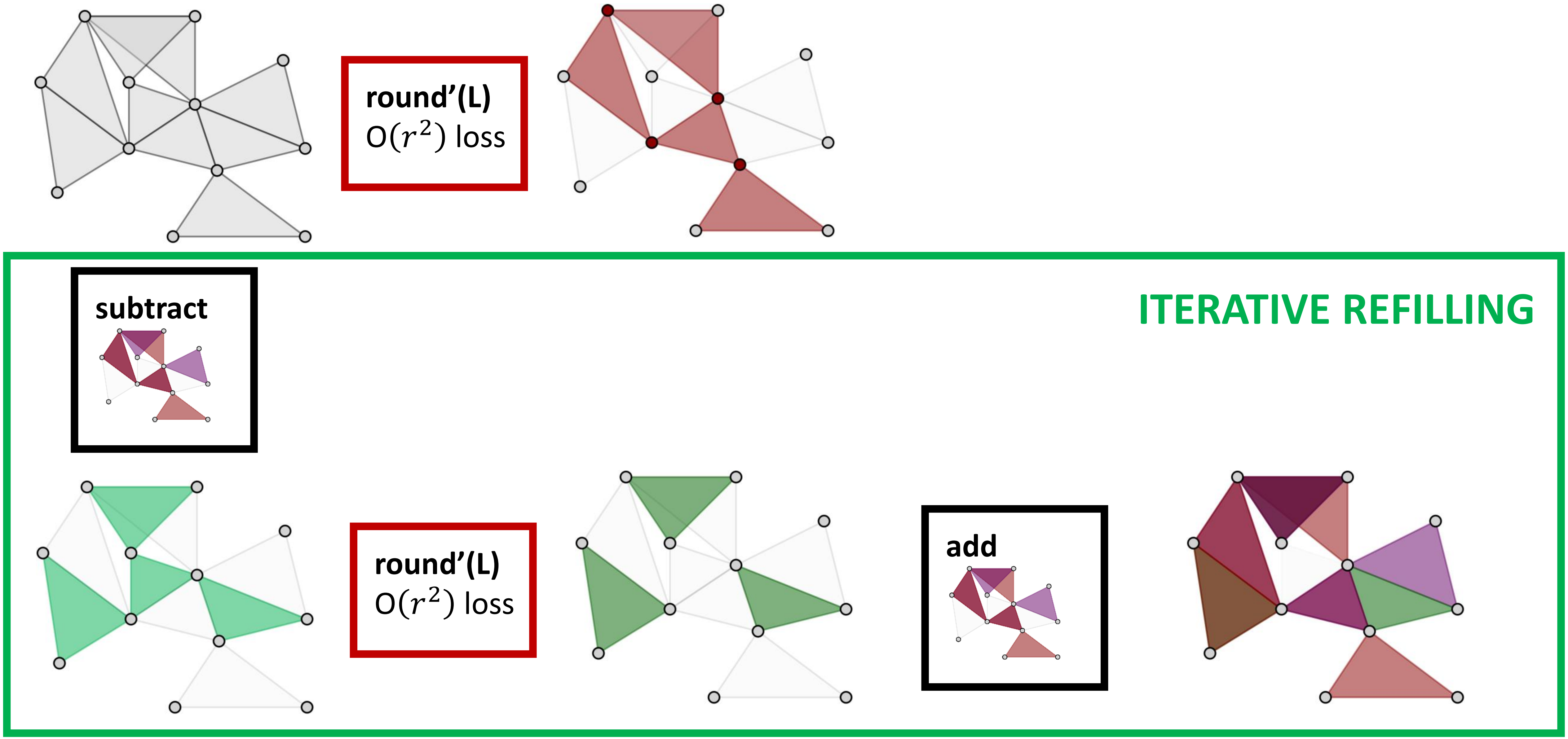
round’(L)
O(r^2) loss



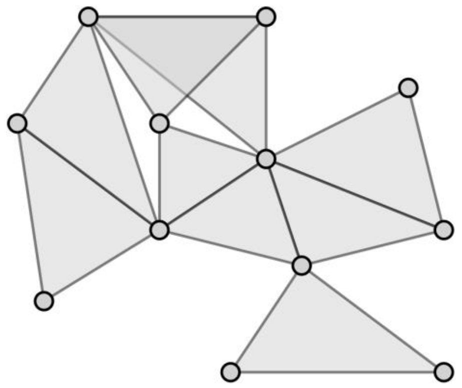
Iterative Refill



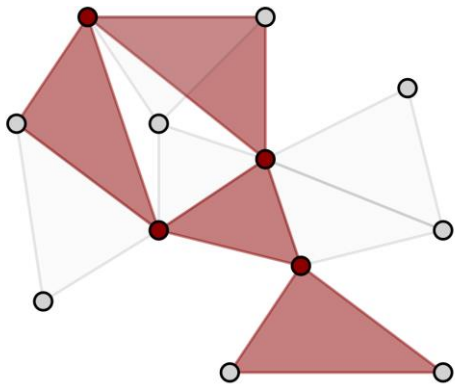
Iterative Refill



Iterative Refill



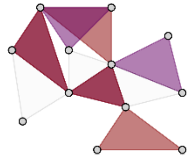
round’(L)
O(r^2) loss

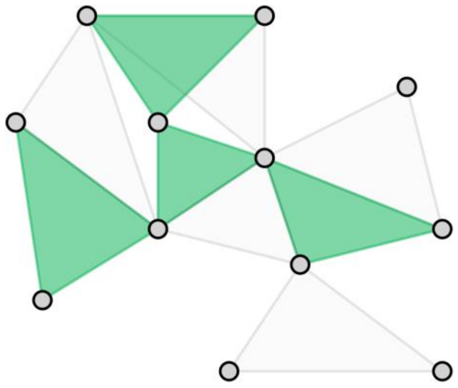


After O(r) refilling iterations:
factor-L-rounding
O(r) loss

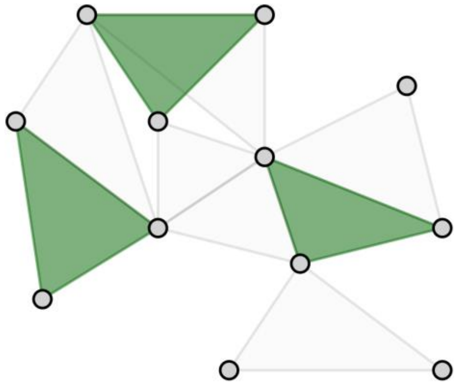
ITERATIVE REFILLING

subtract

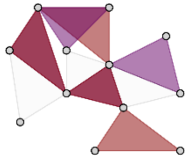


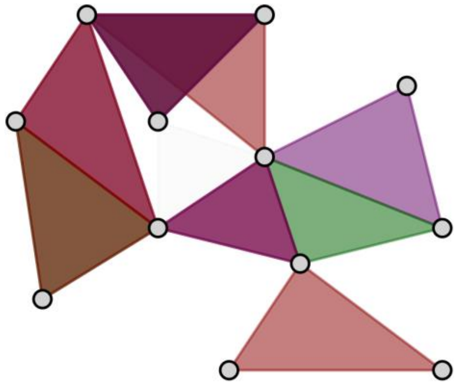


round’(L)
O(r^2) loss

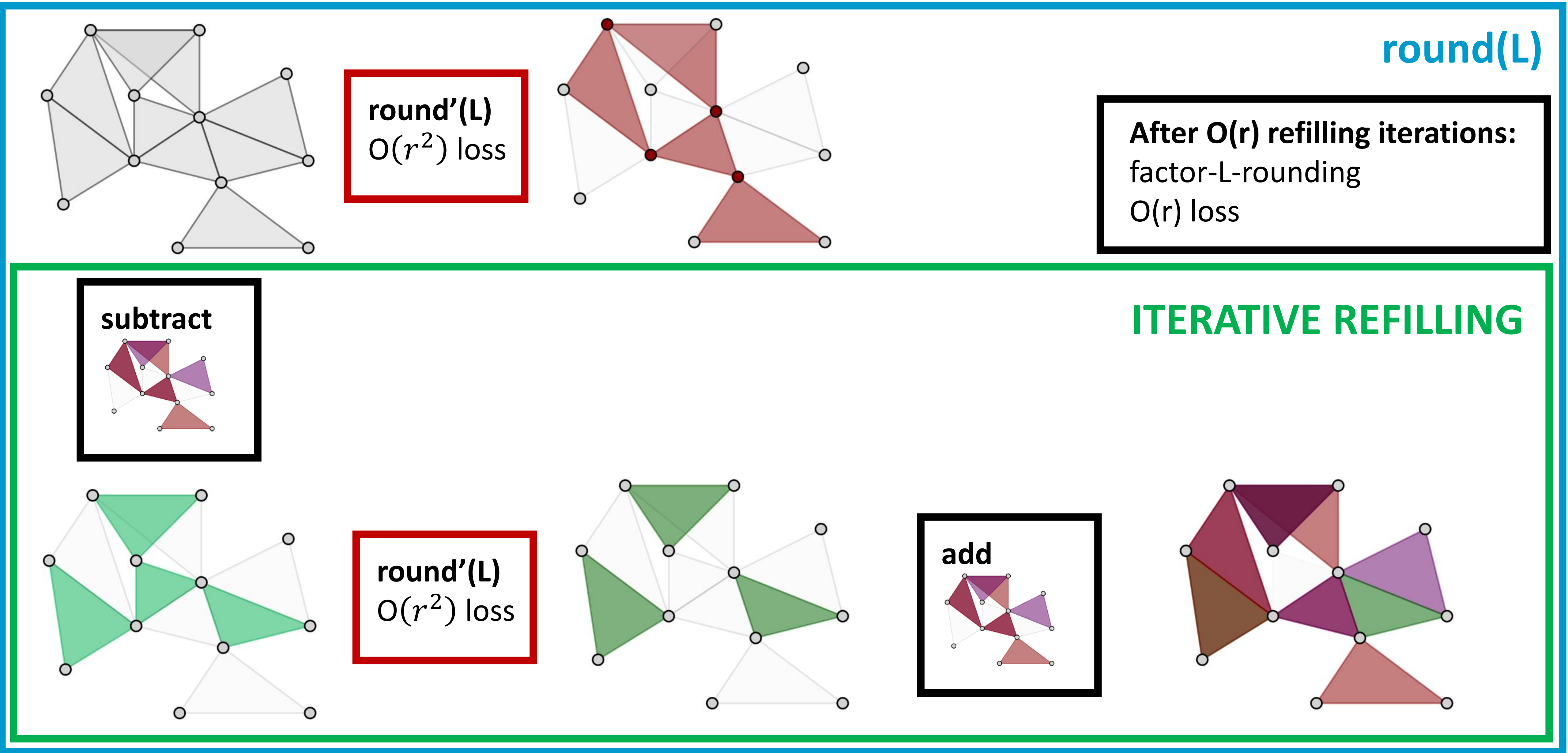


add





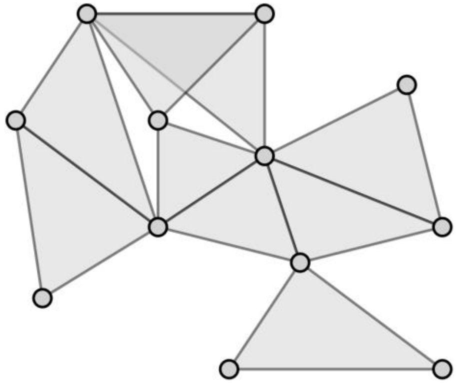
Iterative Refill



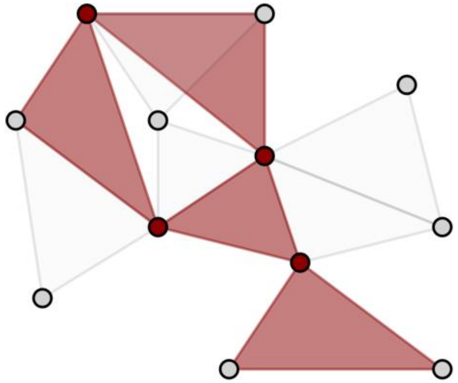
Iterative Refill

$$T[L] = O(r) \cdot T'[L] = O\left(r \cdot T\left[\sqrt{L}\right]\right)$$

round(L)



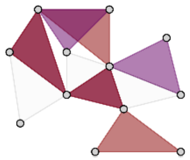
round’(L)
$O(r^2)$ loss

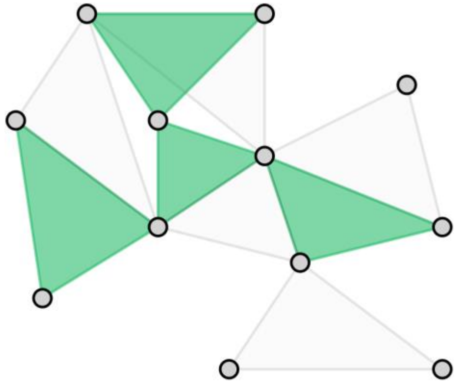


After O(r) refilling iterations:
factor-L-rounding
O(r) loss

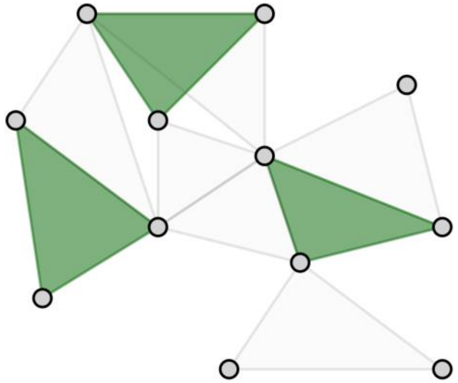
ITERATIVE REFILLING

subtract

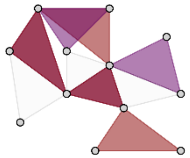


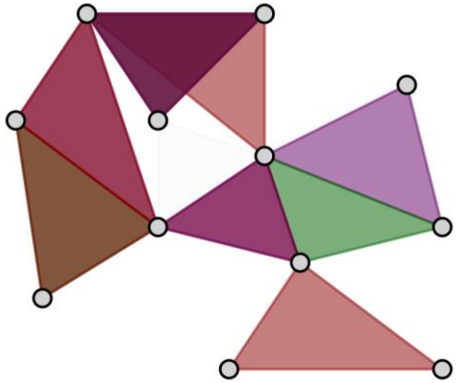


round’(L)
$O(r^2)$ loss



add

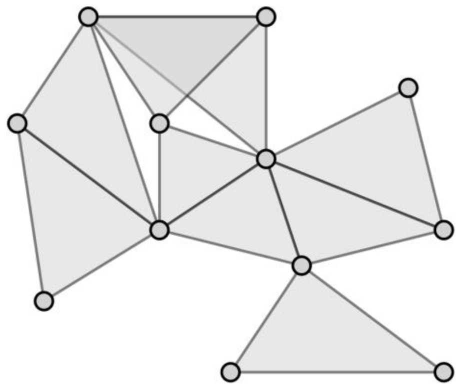




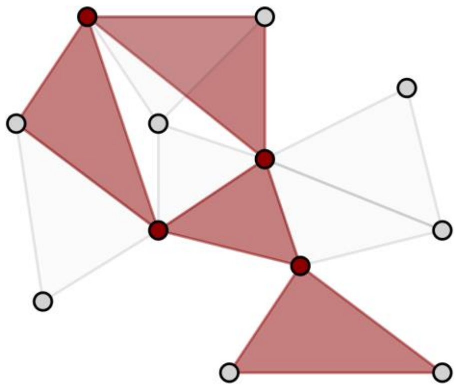
Iterative Refill

$$T[L] = O(r) \cdot T'[L] = O\left(r \cdot T\left[\sqrt{L}\right]\right)$$

$$T[O(1)] = \log \Delta + O\left(r^2\right)$$



round'(L)
$O(r^2)$ loss

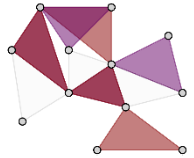


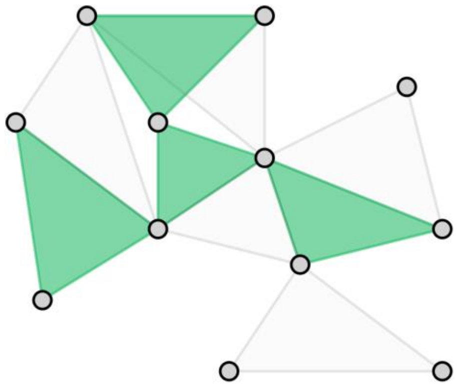
round(L)

After O(r) refilling iterations:
factor-L-rounding
O(r) loss

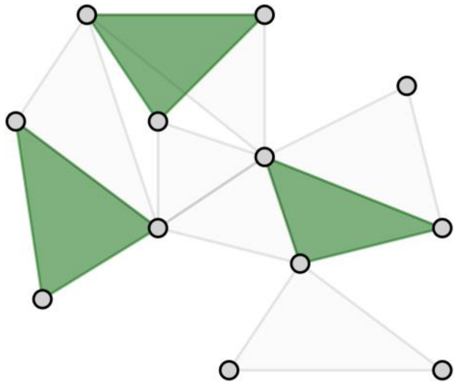
ITERATIVE REFILLING

subtract

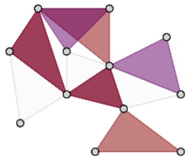


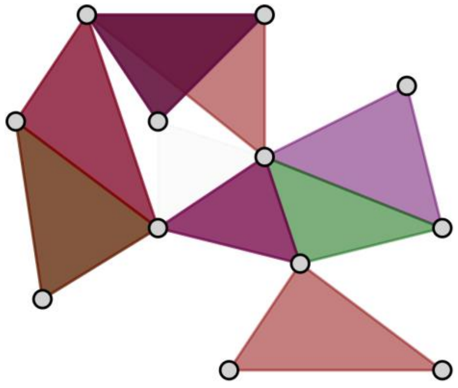


round'(L)
$O(r^2)$ loss



add

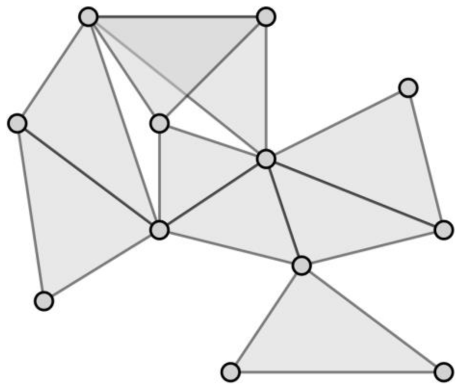




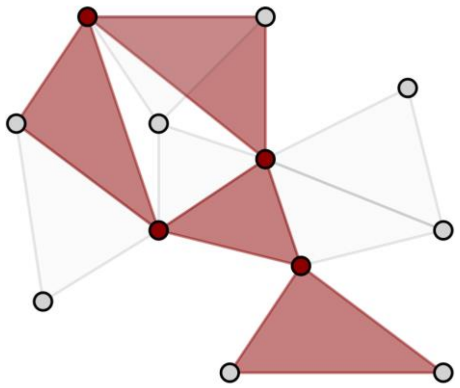
Iterative Refill

$$\left.\begin{aligned} T[L] &= O(r)\cdot T'[L] = O\left(r\cdot T[\sqrt{L}]\right) \\ T[O(1)] &= \log\Delta + O(r^2) \end{aligned}\right\} \quad T[O(\Delta)] = r^{O(\log\log\Delta)}\cdot\left(\log\Delta + O(r^2)\right)$$

round(L)



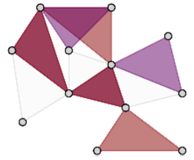
round’(L)
$O(r^2)$ loss

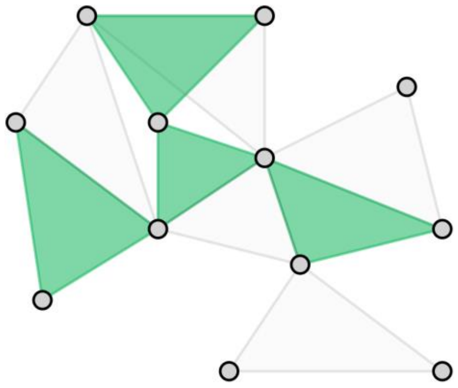


After O(r) refilling iterations:
factor-L-rounding
O(r) loss

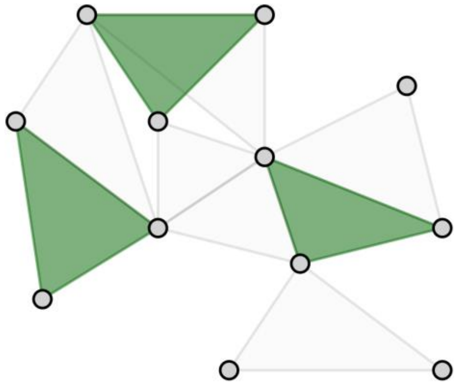
ITERATIVE REFILLING

subtract

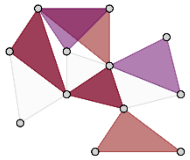


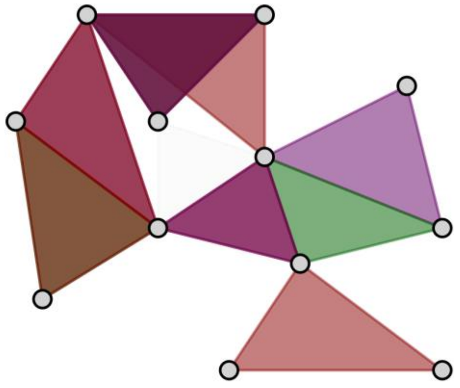


round’(L)
$O(r^2)$ loss



add





Conclusion & Open Problems

- **$(2\Delta - 1)$-Edge-Coloring is efficient: $O(\log^8 n)$**
- **Linial's Question from the 1980s:**

  Is there an efficient deterministic algorithm for Maximal Independent Set?
- **Generality of Hypergraph Maximal Matchings** $\mathbf{poly\, r \cdot \log^{O(\log r)} \Delta \cdot \log n}$ **rounds**
  - Devise or rule out a poly $(r \cdot \log n)$-round deterministic algorithm for rank-r-hypergraph maximal matching.
- **Key Problem: Efficient Deterministic Rounding**
  - **Completeness of Rounding** (Ghaffari, Kuhn, Maus [STOC'17])

    Rounding as the only obstacle for efficient deterministic LOCAL graph algorithms
  - Devise a more general deterministic rounding method.