Distributed Algorithms for Local Potential Problems

GRAN SASSO SCIENCE INSTITUTE

Francesco d'Amore

Joint work with A. Balliu, T. Boudier, D. Olivetti, G. Schmid, and J. Suomela

Helsinki Algorithms & Theory Days
29 August 2025

The result

Theorem

For any local potential problem Π*, there exists a randomized LOCAL algorithm that solves* Π *with high probability in time* $O(\log^6 n)$.

The result

Theorem

For any local potential problem Π*, there exists a randomized LOCAL algorithm that solves* Π *with high probability in time* $O(\log^6 n)$.

derandomization [Ghaffari, Harris, and Kuhn, FOCS '18]

+

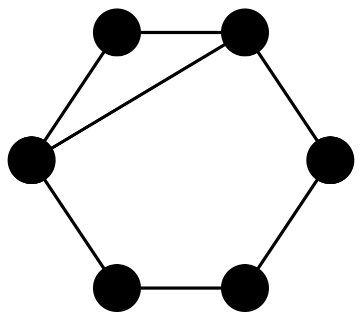
network decomposition [Ghaffari and Grunau, FOCS '24]

Corollary:

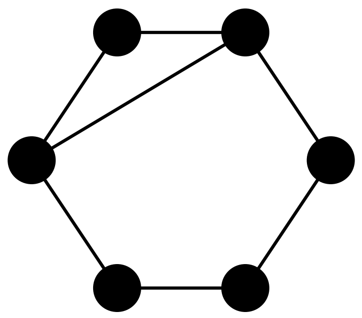
For any local potential problem Π*, there exists a deterministic LOCAL algorithm that solves* Π *in time* $O(\log^8 n \, \mathit{poly}(\log\log n))$.

Locally Optimal Cut (LOC)

Input: - graph $G = (V, E)$
- two colors red and green



Locally Optimal Cut (LOC)

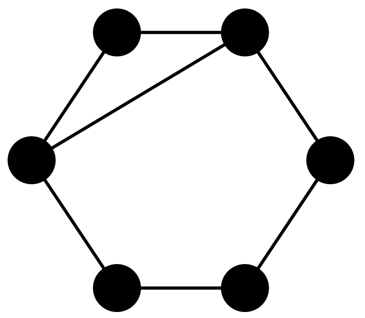


Input: - graph $G = (V, E)$
- two colors red and green

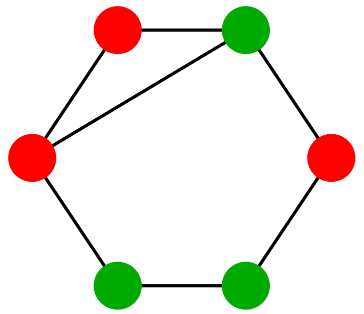
Output: - a (not necessarily proper) 2-coloring $c: V \to \{\text{red}, \text{green}\}$ of G
- for each $v \in V$, at least $\geq \deg(v)/2$ *neighbors of different color* (w.r.t. v)

Locally Optimal Cut (LOC)

Input: - graph $G = (V, E)$
- two colors red and green



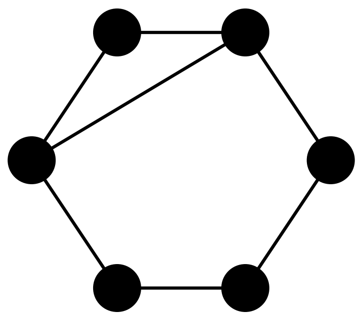
Output: - a (not necessarily proper) 2-coloring $c: V \to \{\text{red}, \text{green}\}$ of G
- for each $v \in V$, at least $\geq \deg(v)/2$ *neighbors of different color* (w.r.t. v) → *locally checkable*



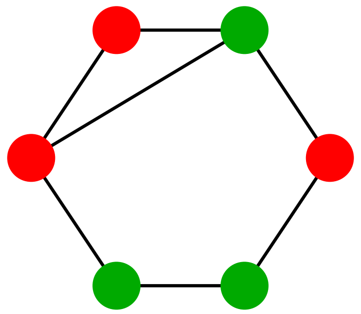
valid solution

Locally Optimal Cut (LOC)

Input:
- graph $G = (V, E)$
- two colors red and green



Output:
- a (not necessarily proper) 2-coloring $c: V \to \{\text{red}, \text{green}\}$ of G
- for each $v \in V$, at least $\geq \deg(v)/2$ *neighbors of different color* (w.r.t. v) → *locally checkable*

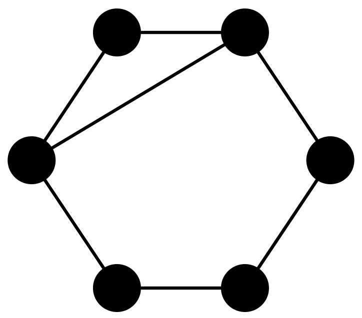


valid solution

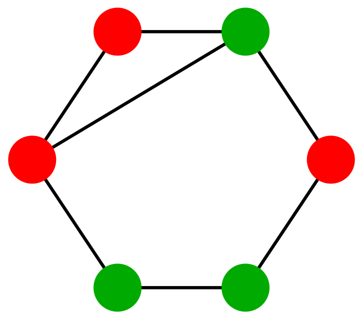
2-*apx* of MAX-CUT (locally optimal)

Locally Optimal Cut (LOC)

Input:
- graph $G = (V, E)$
- two colors red and green

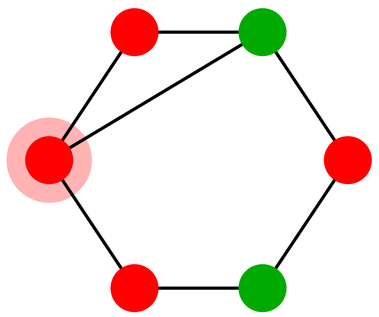


Output:
- a (not necessarily proper) 2-coloring $c: V \to \{\text{red}, \text{green}\}$ of G
- for each $v \in V$, at least $\geq \deg(v)/2$ *neighbors of different color* (w.r.t. v) → *locally checkable*



valid solution

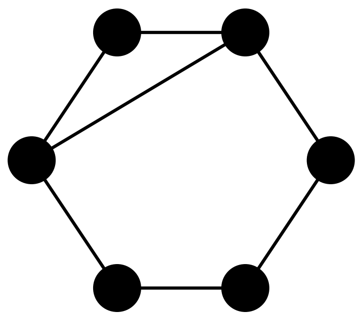
2-*apx* of MAX-CUT (locally optimal)



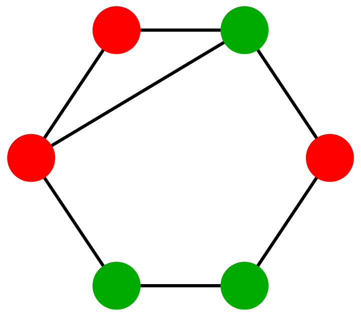
invalid solution

Locally Optimal Cut (LOC)

Input: - graph $G = (V, E)$
- two colors red and green

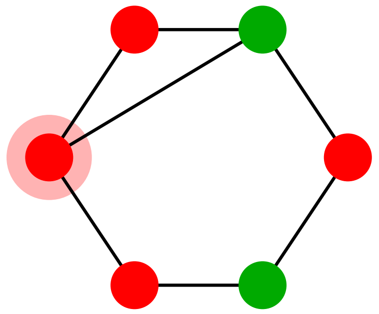


Output: - a (not necessarily proper) 2-coloring $c: V \to \{\text{red}, \text{green}\}$ of G
- for each $v \in V$, at least $\geq \deg(v)/2$ *neighbors of different color* (w.r.t. v) → *locally checkable*



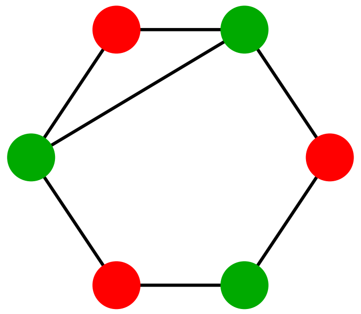
valid solution

2-*apx* of MAX-CUT (locally optimal)



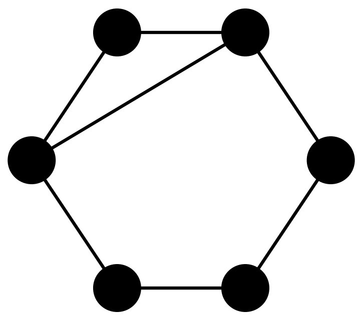
invalid solution

fixing procedure → *flip color of invalid node*

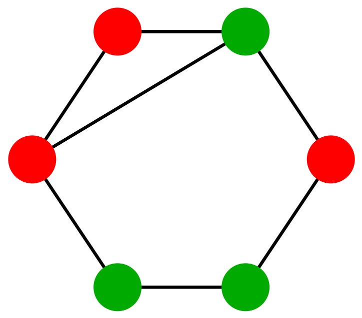


Locally Optimal Cut (LOC)

Input: - graph $G = (V, E)$
- two colors red and green

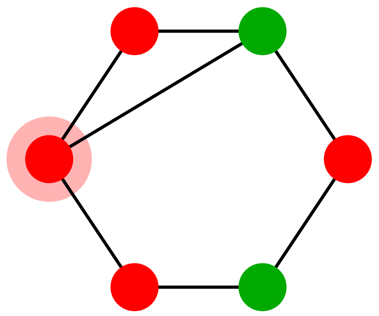


Output: - a (not necessarily proper) 2-coloring $c: V \to \{\text{red}, \text{green}\}$ of G
- for each $v \in V$, at least $\geq \mathbf{deg}(v)/2$ *neighbors of different color* (w.r.t. v) → *locally checkable*



valid solution

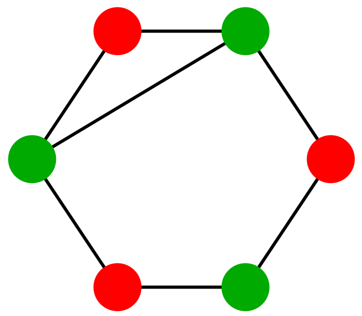
2-*apx* of MAX-CUT (locally optimal)



invalid solution

fixing procedure

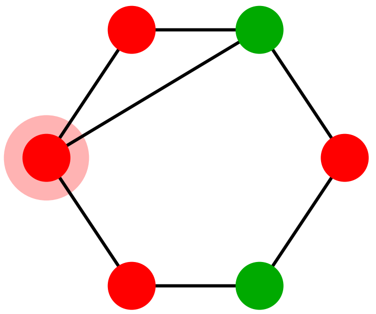
flip color of invalid node



Question: can we *always* solve the problem?

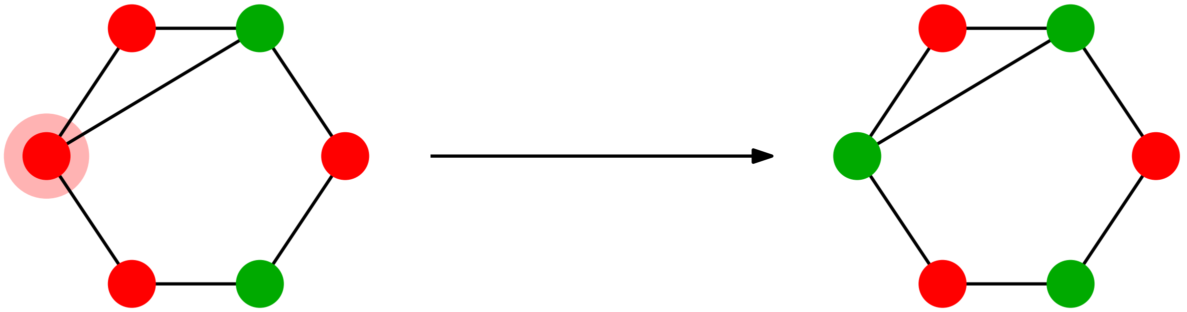
LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*



LOC: fixing procedure

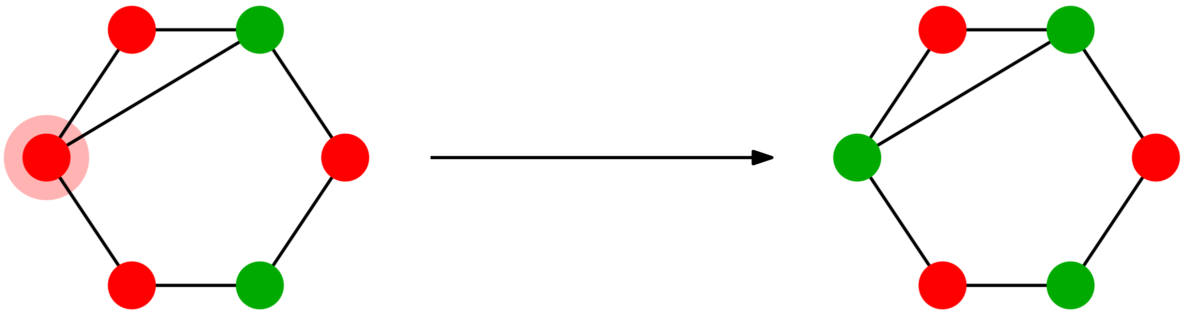
Fixing procedure: *pick an arbitrary invalid node and flip its color*



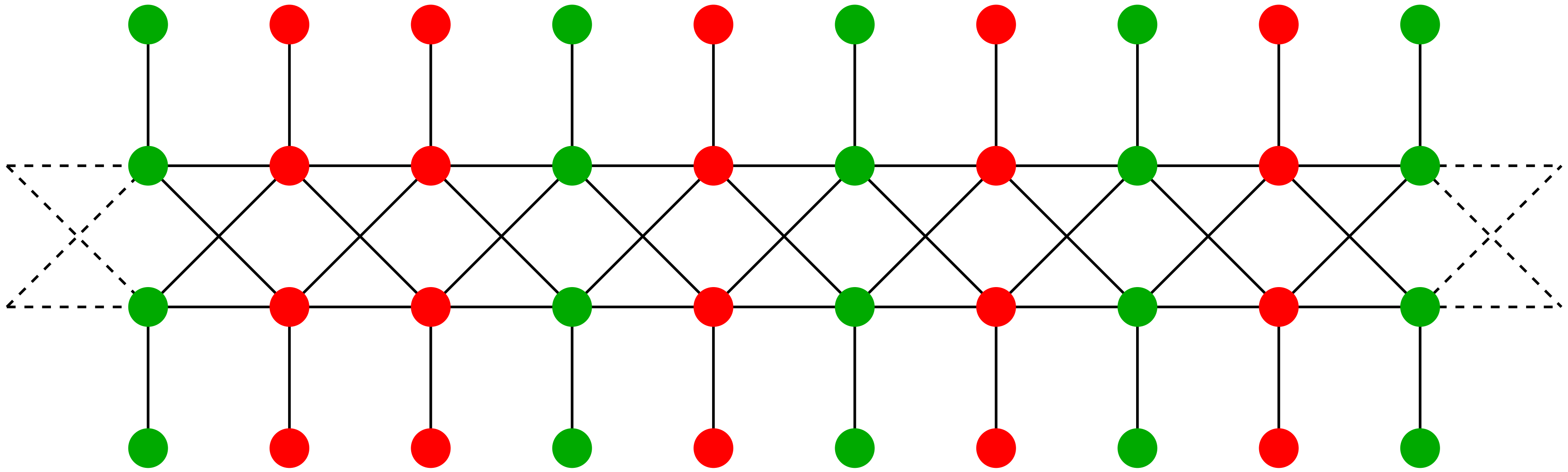
valid solution

LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

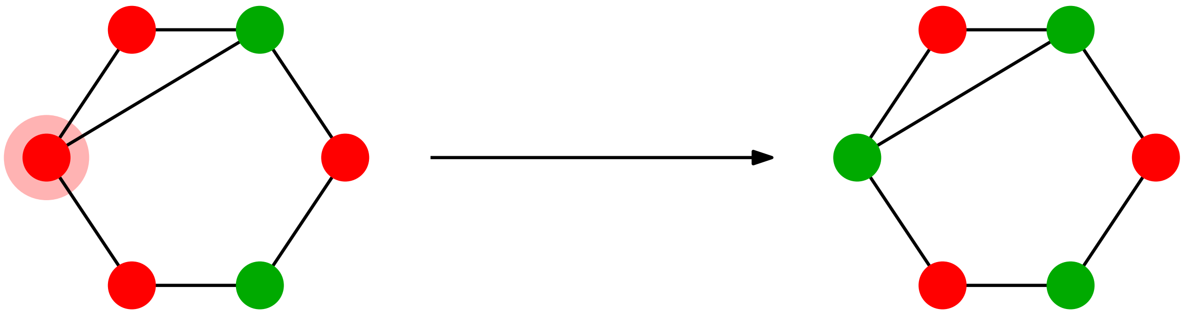


valid solution

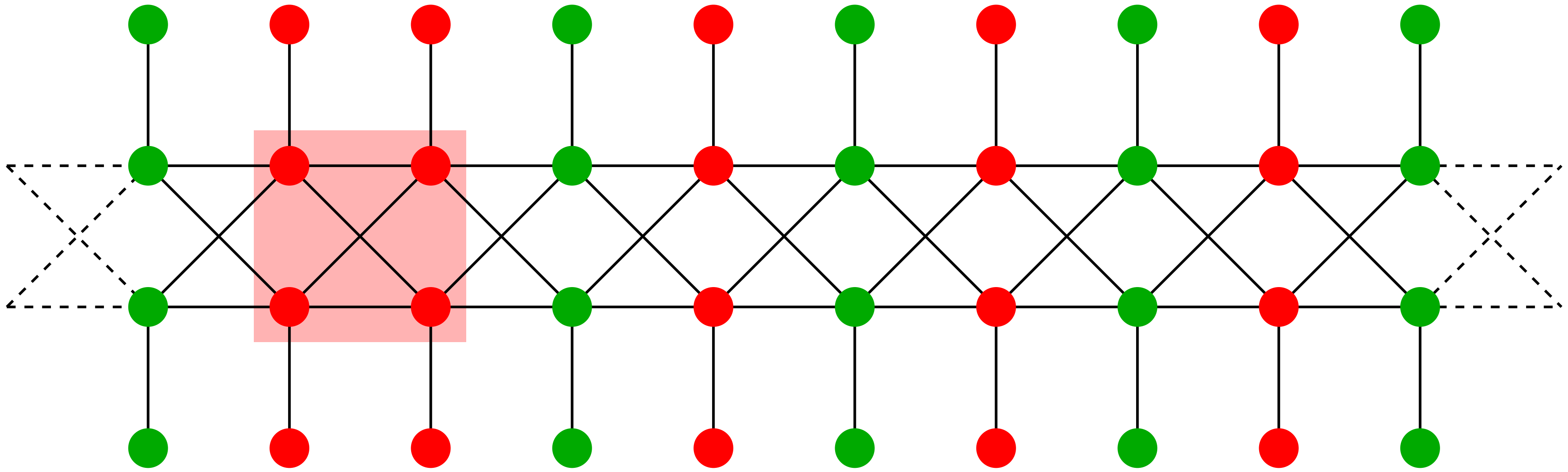


LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

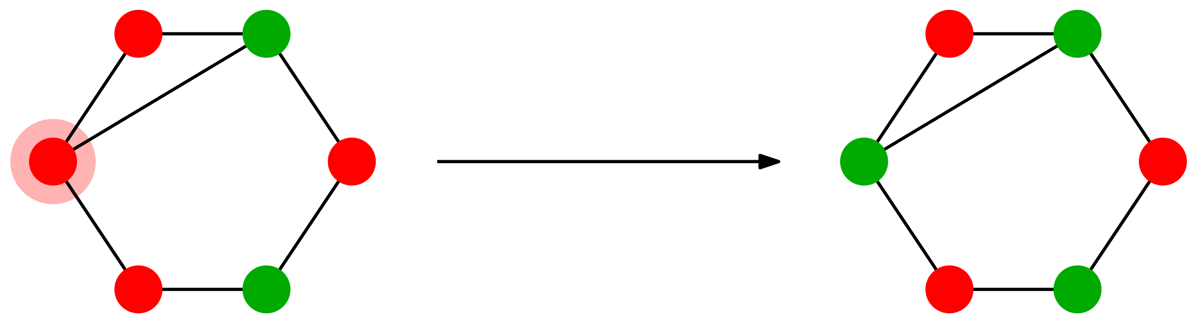


valid solution

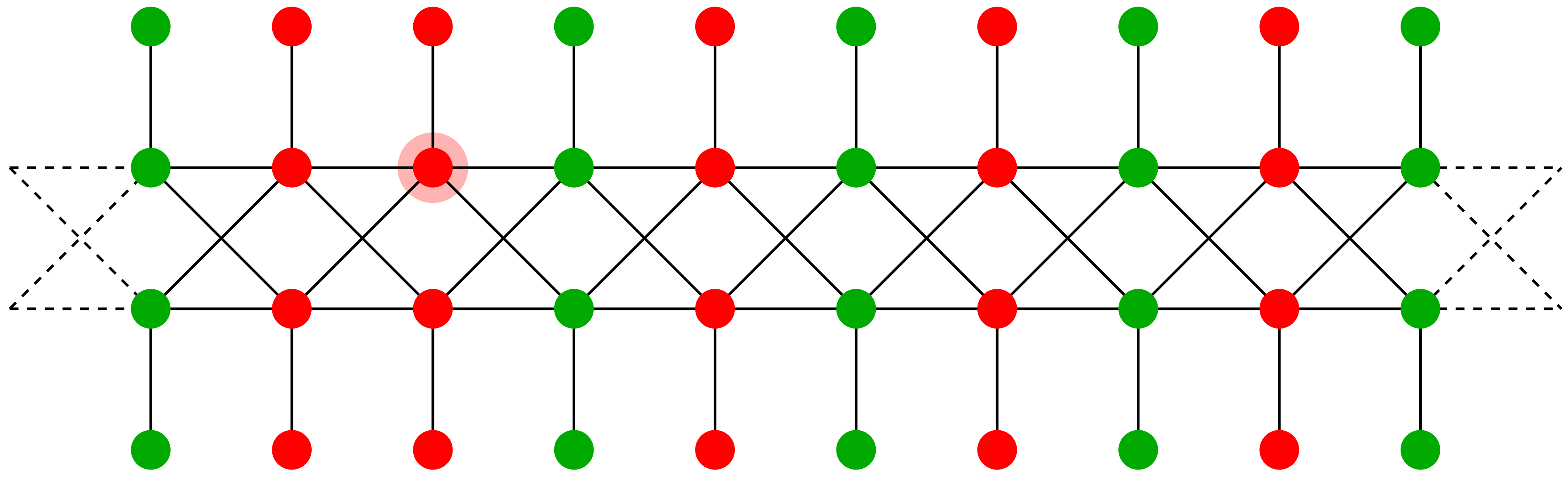


LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

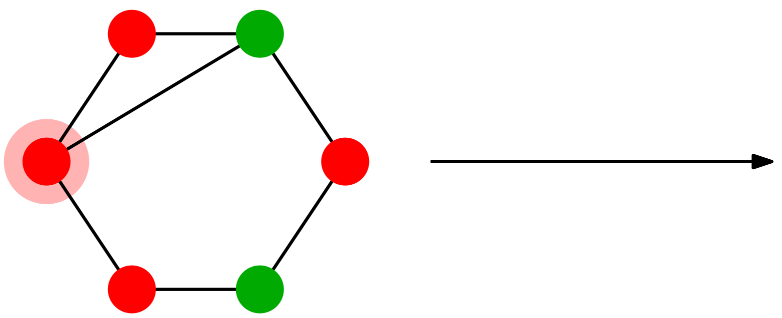


valid solution

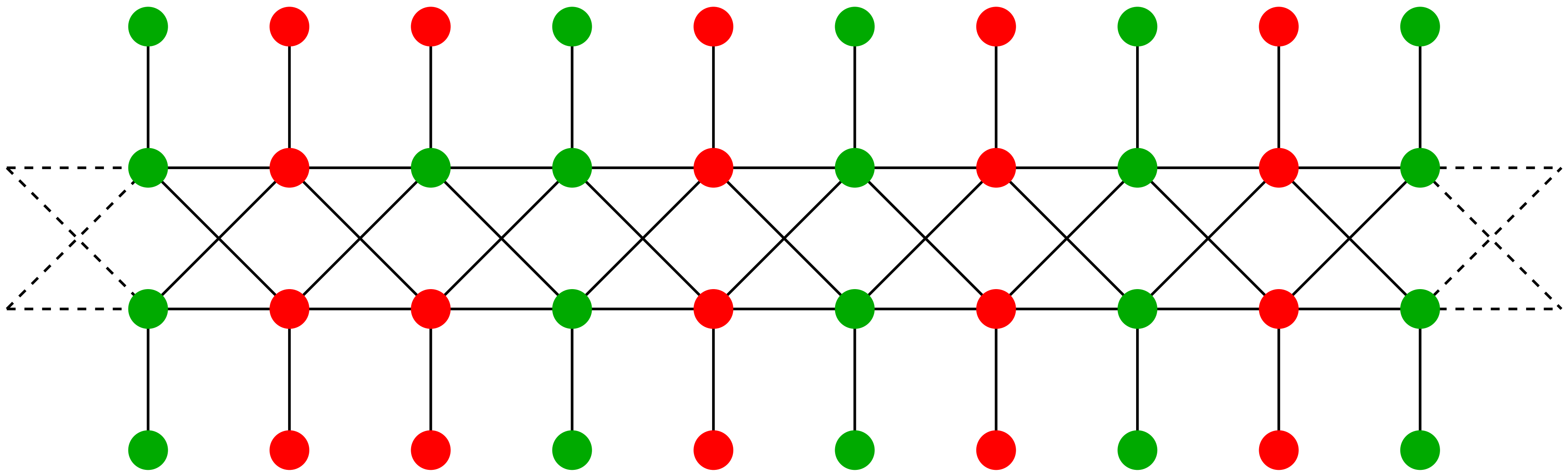


LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

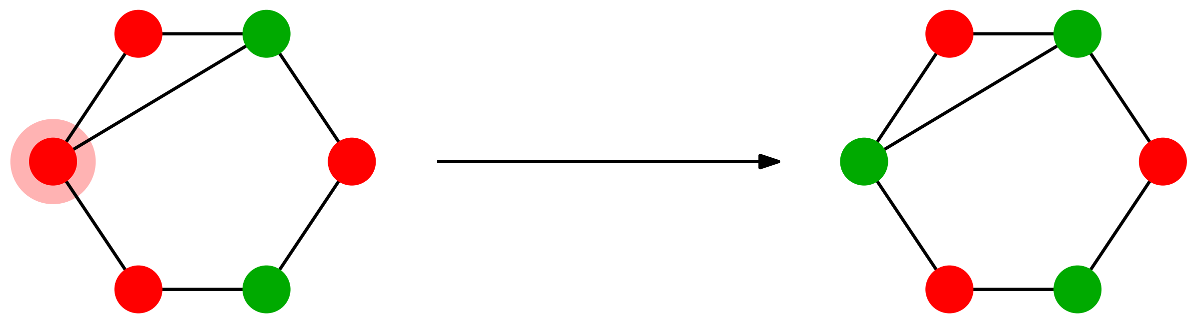


valid solution

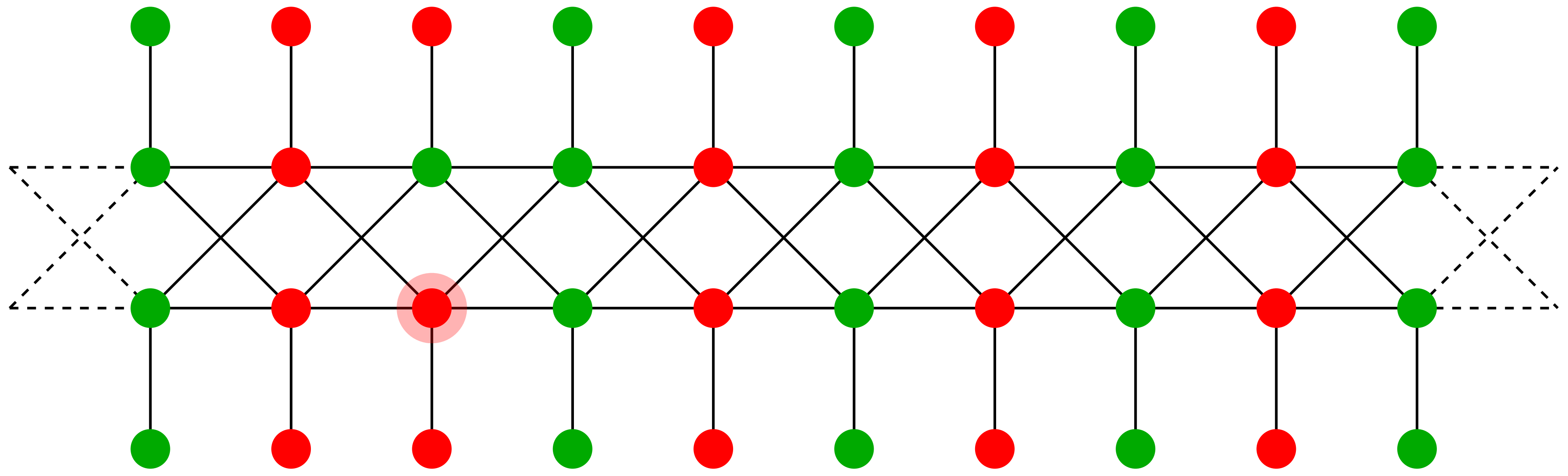


LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

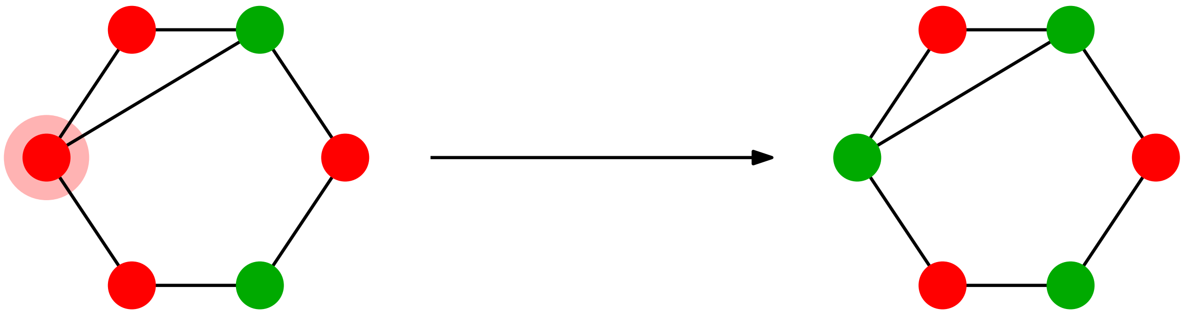


valid solution

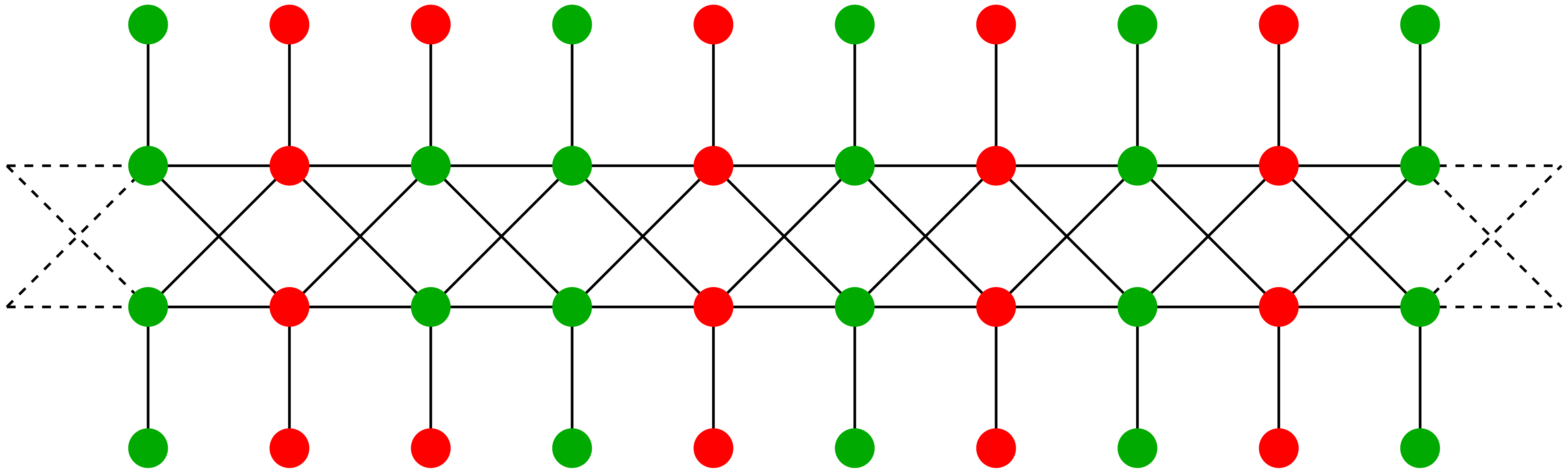


LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

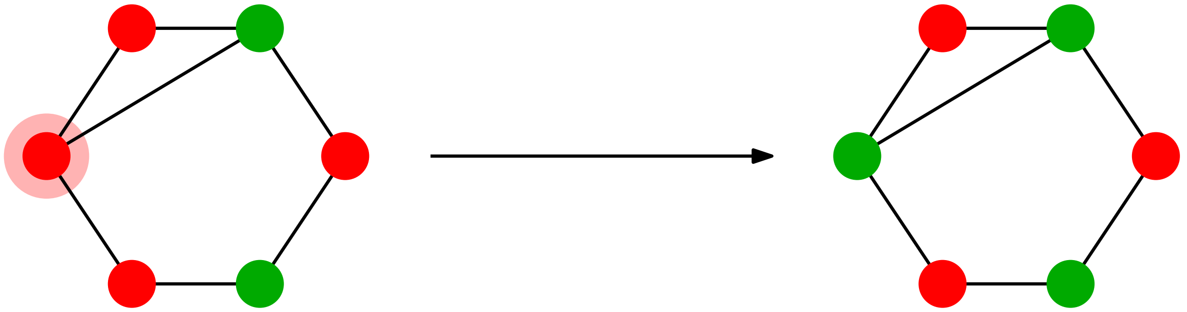


valid solution

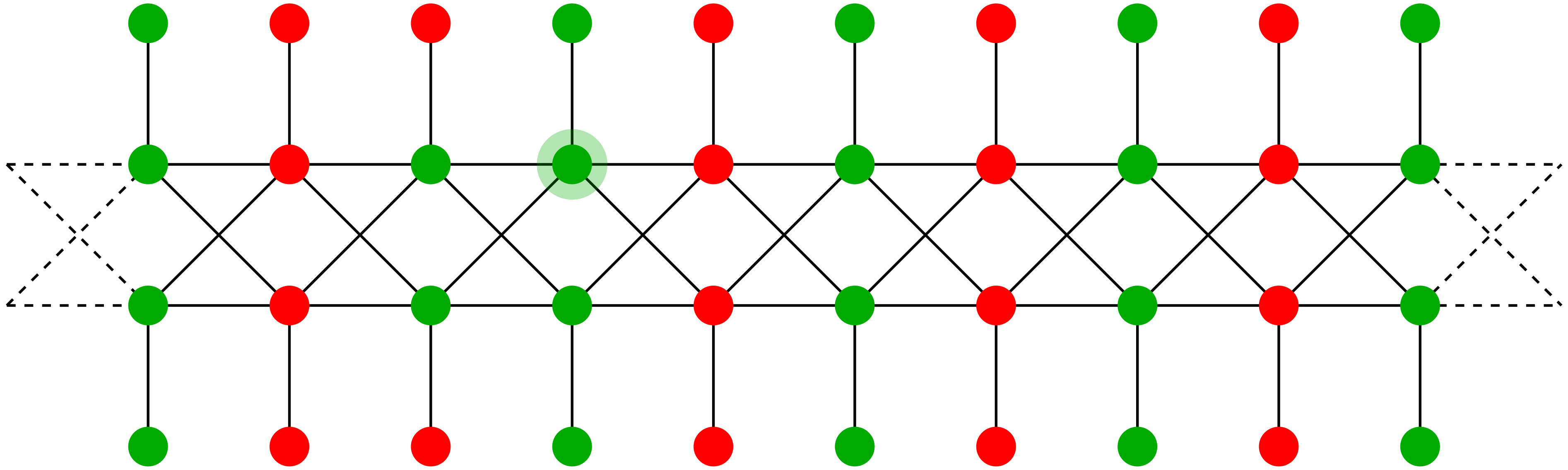


LOC: fixing procedure

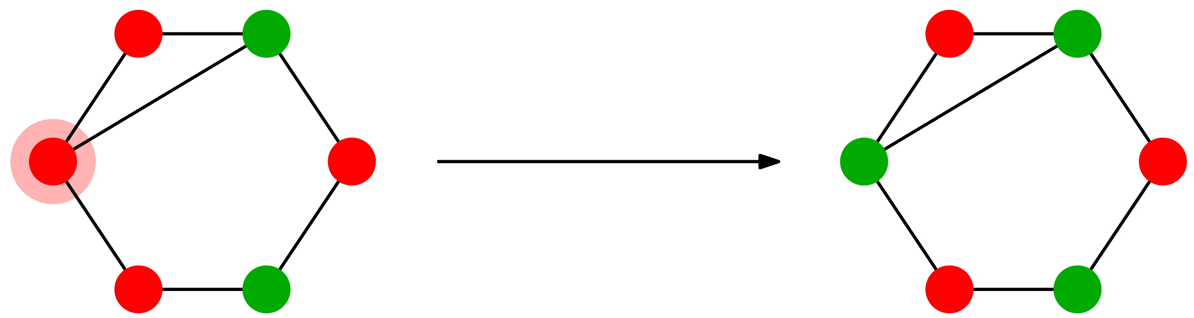
Fixing procedure: *pick an arbitrary invalid node and flip its color*



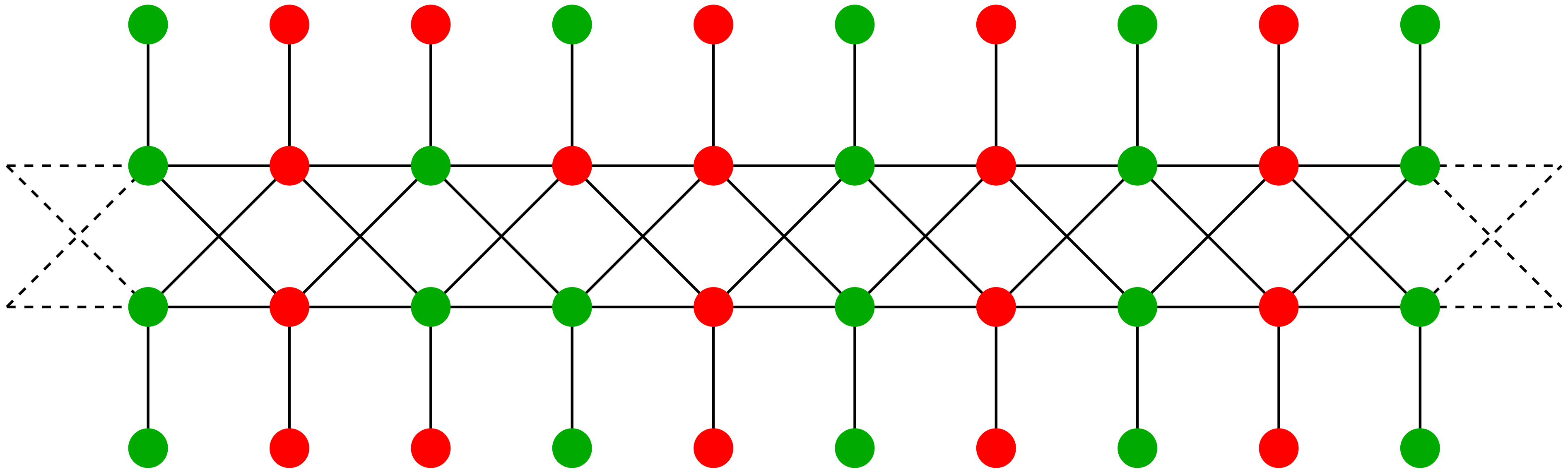
valid solution



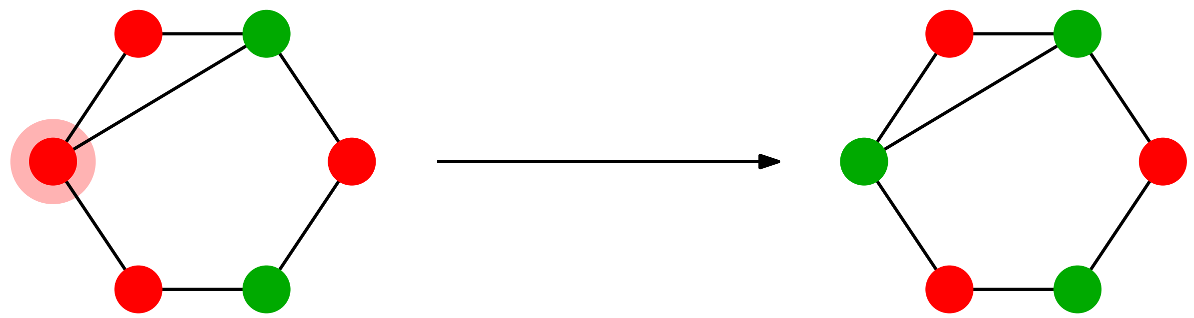
Fixing procedure: *pick an arbitrary invalid node and flip its color*



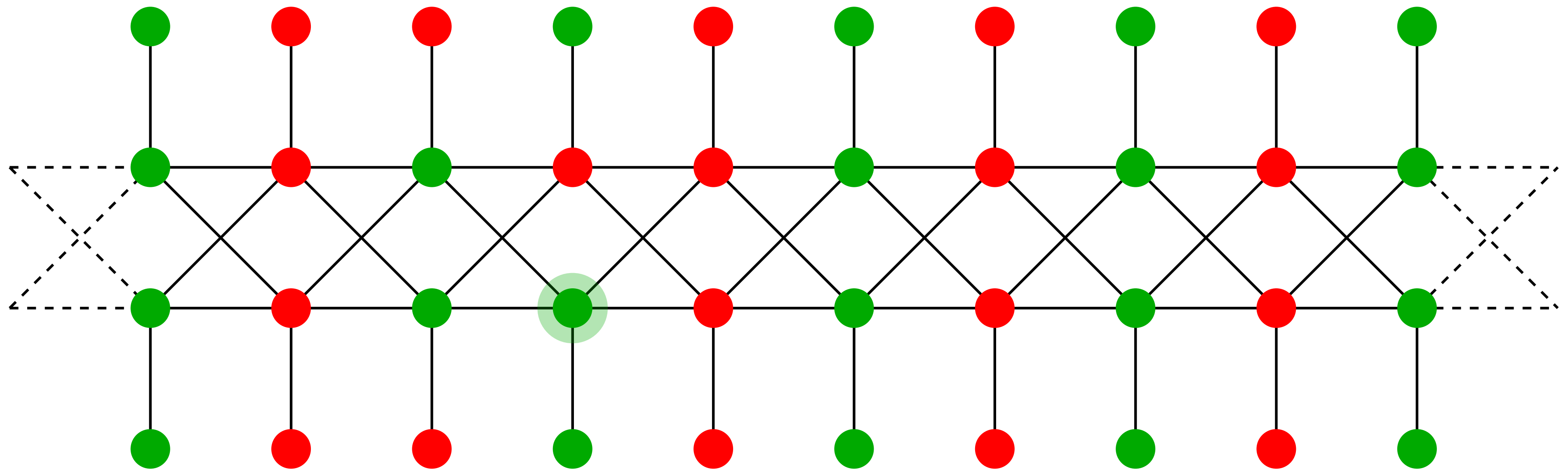
valid solution



Fixing procedure: *pick an arbitrary invalid node and flip its color*

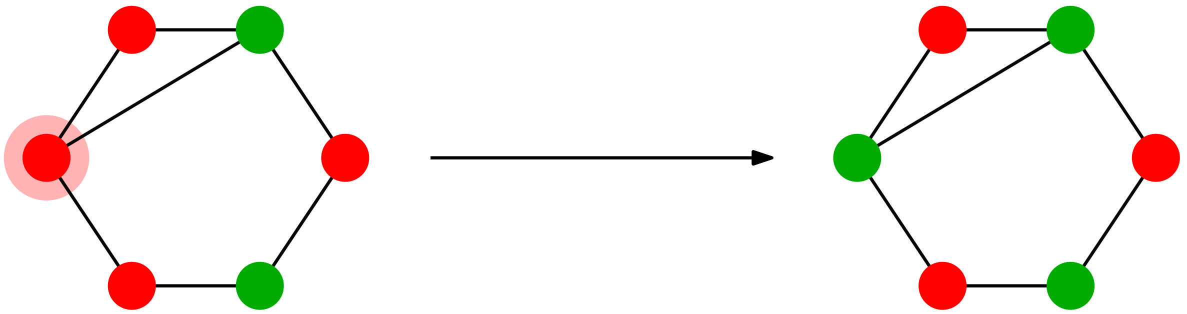


valid solution

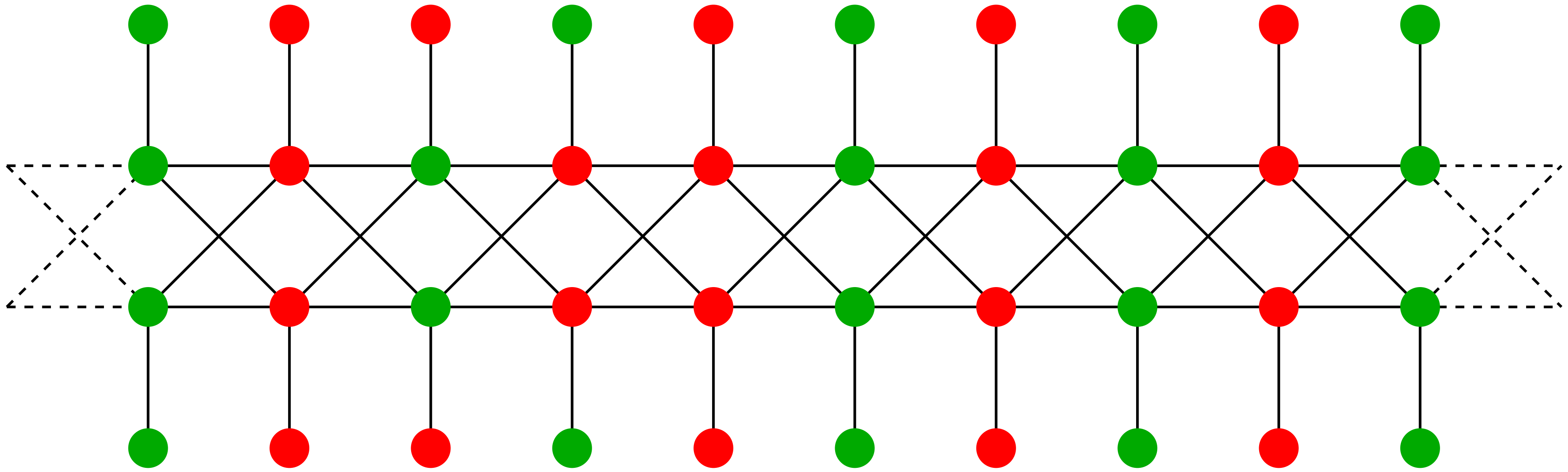


LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

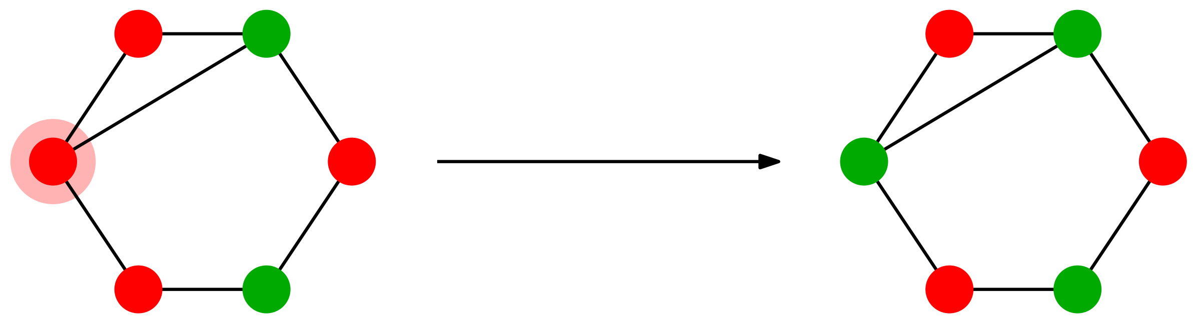


valid solution

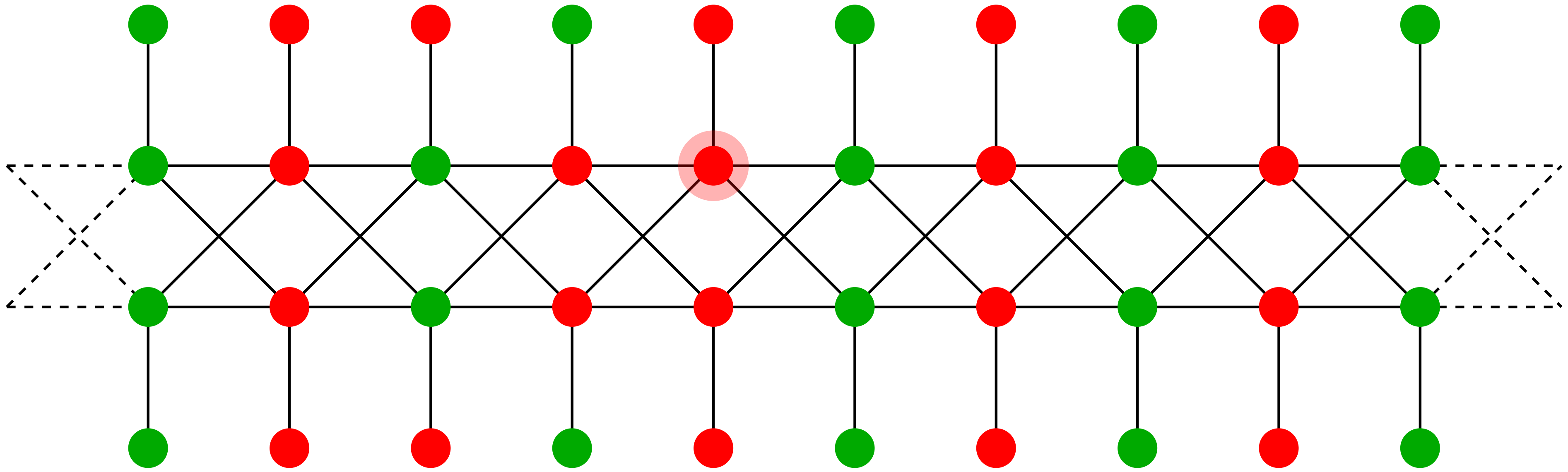


LOC: fixing procedure

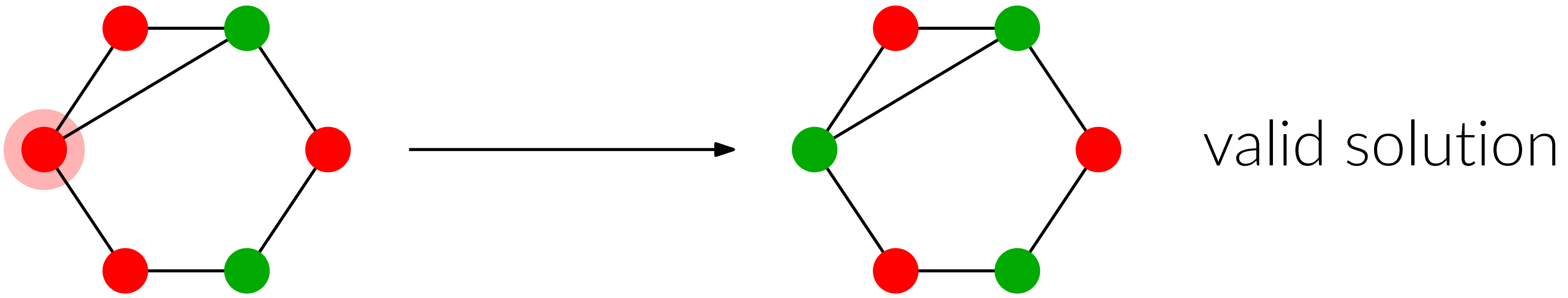
Fixing procedure: *pick an arbitrary invalid node and flip its color*

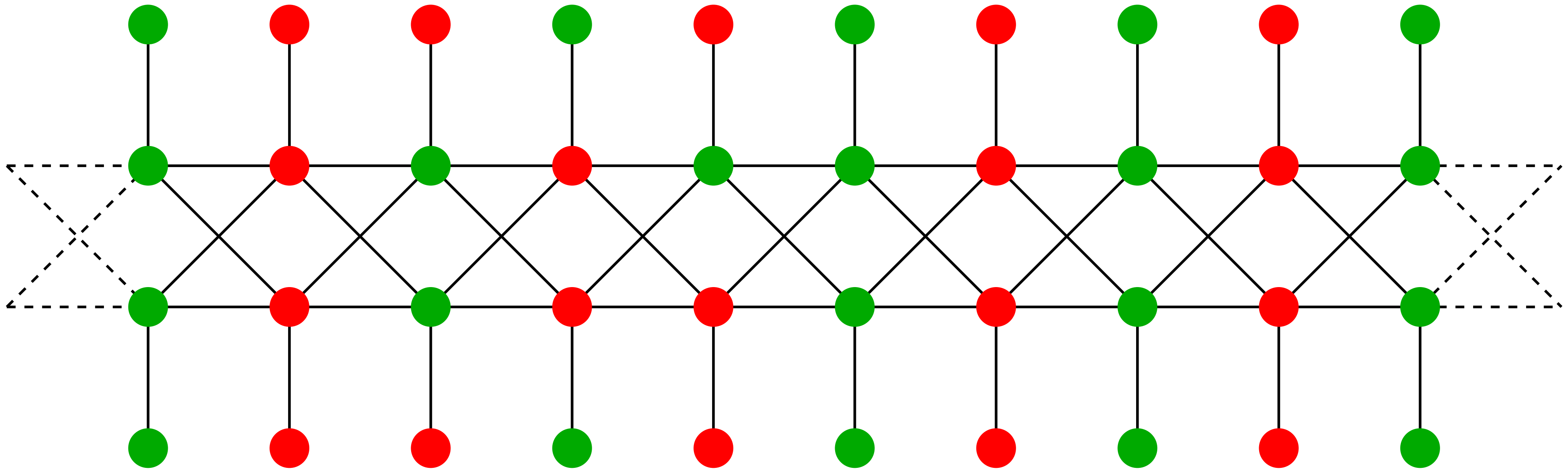


valid solution



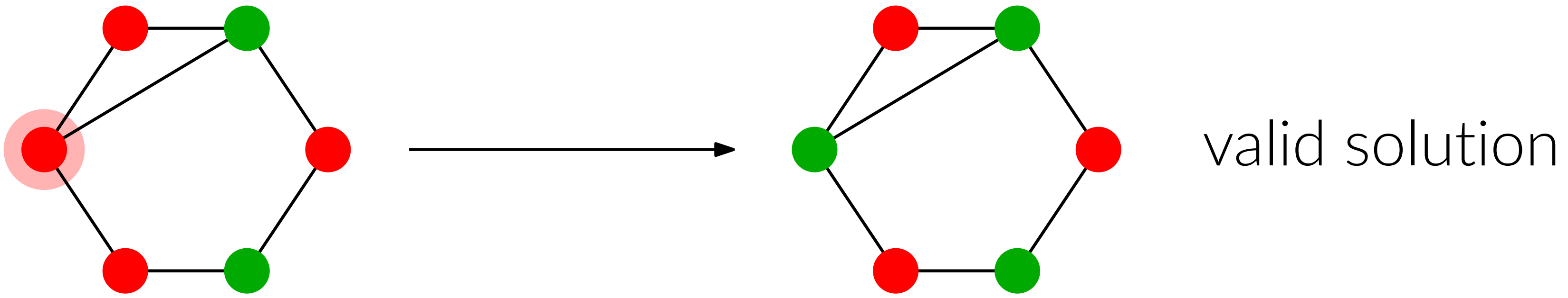
Fixing procedure: *pick an arbitrary invalid node and flip its color*

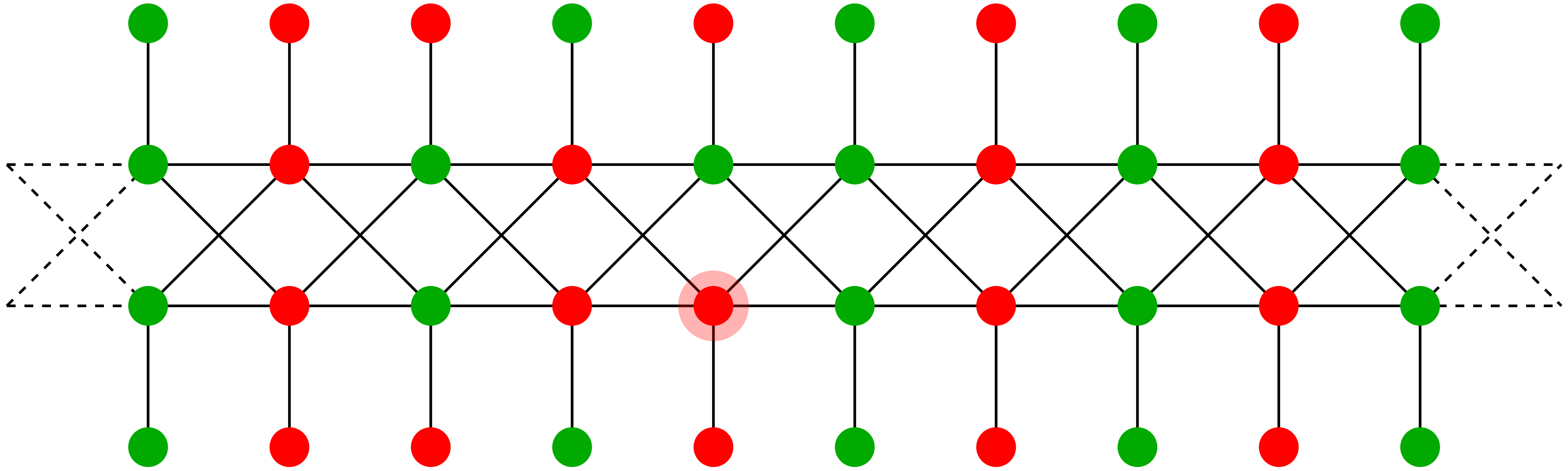




LOC: fixing procedure

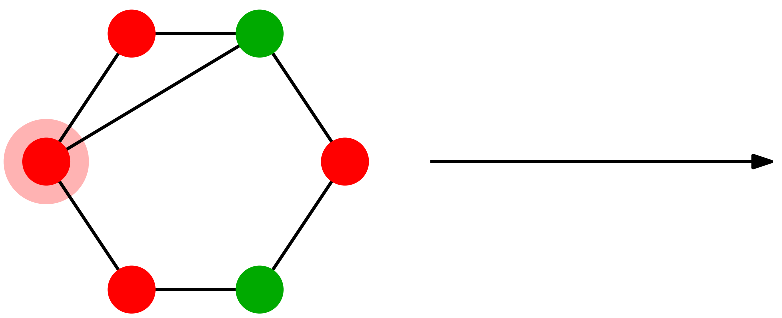
Fixing procedure: *pick an arbitrary invalid node and flip its color*



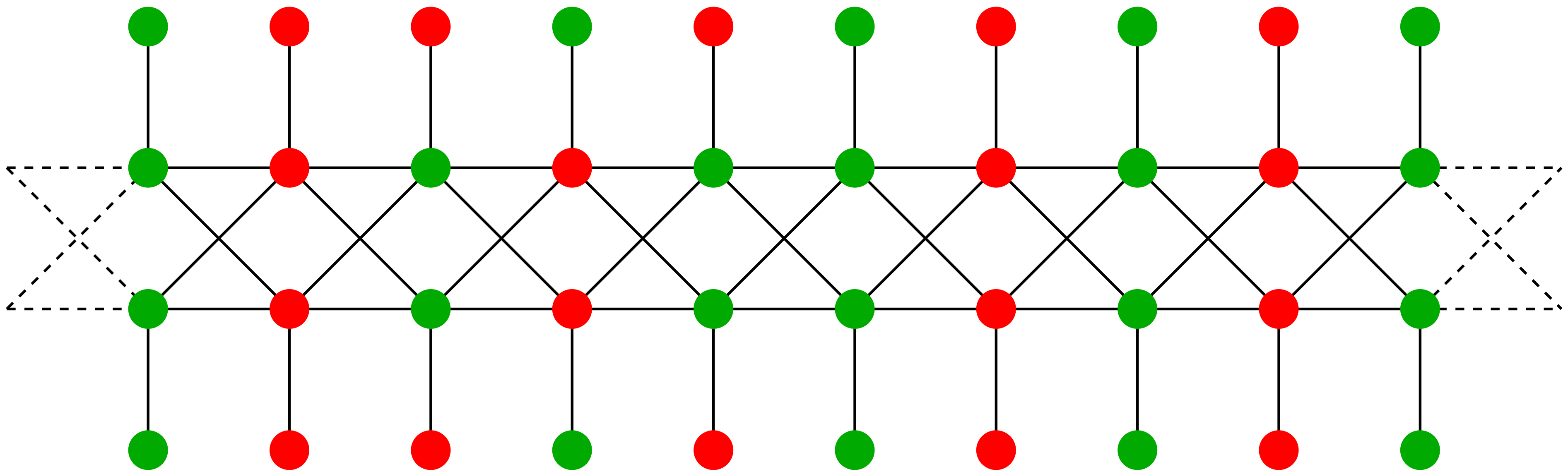


LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

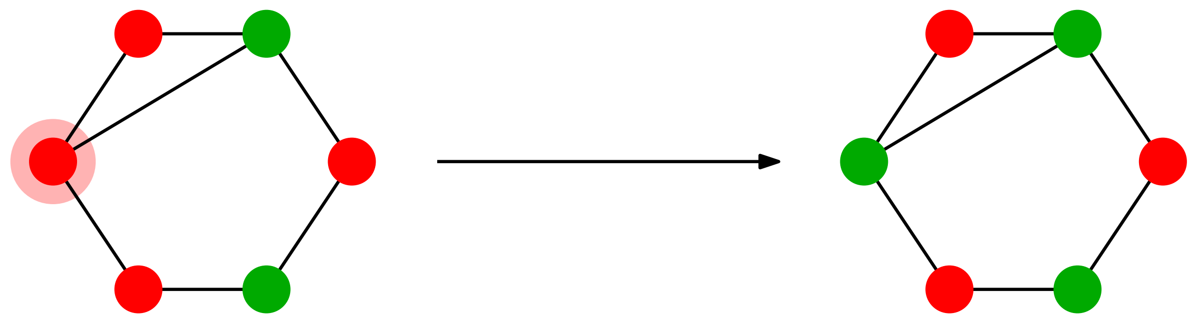


valid solution

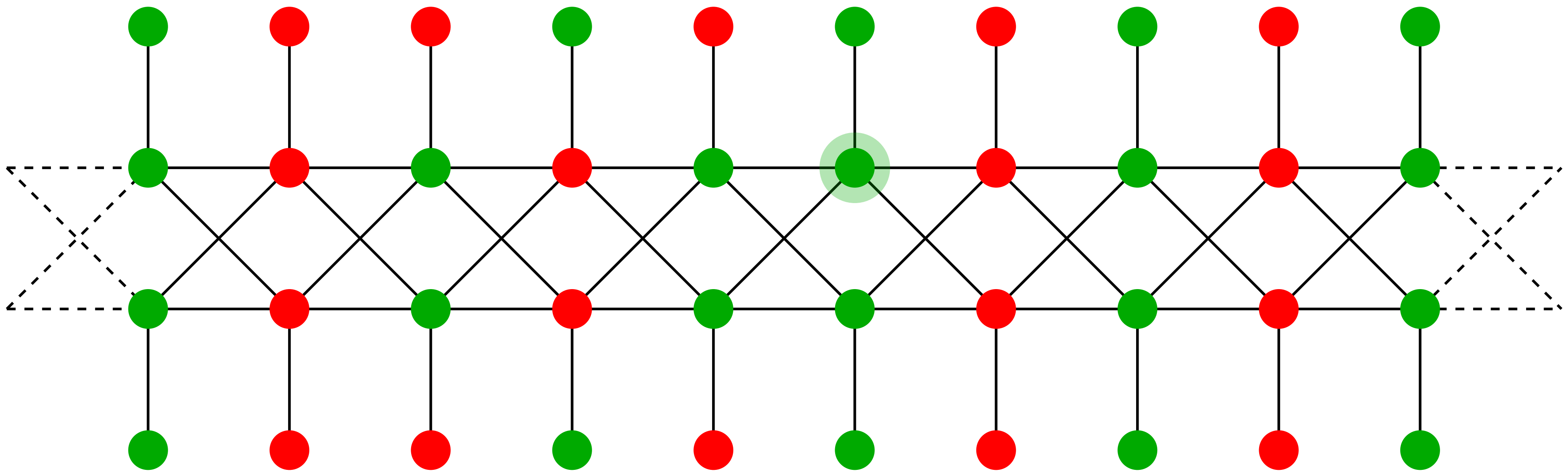


LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

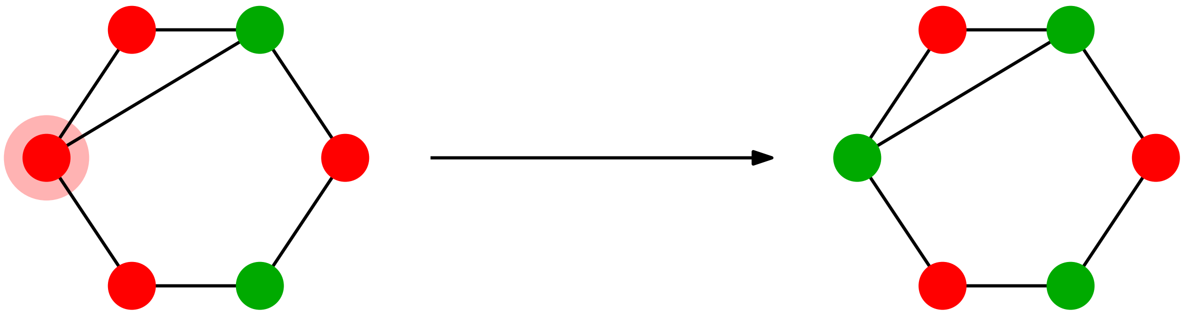


valid solution

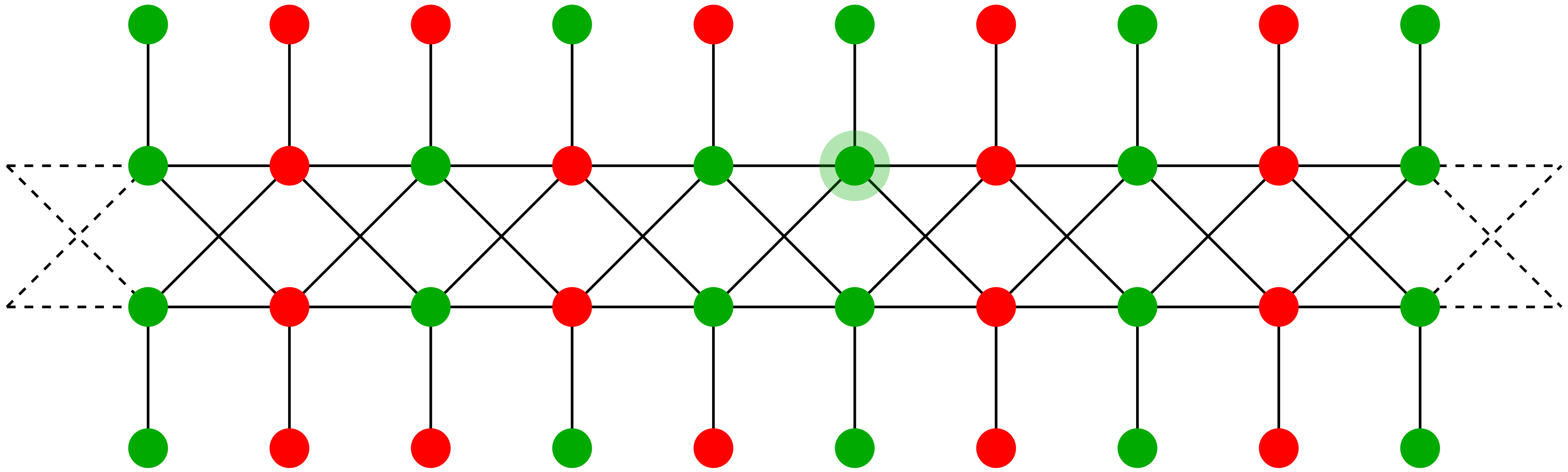


LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*

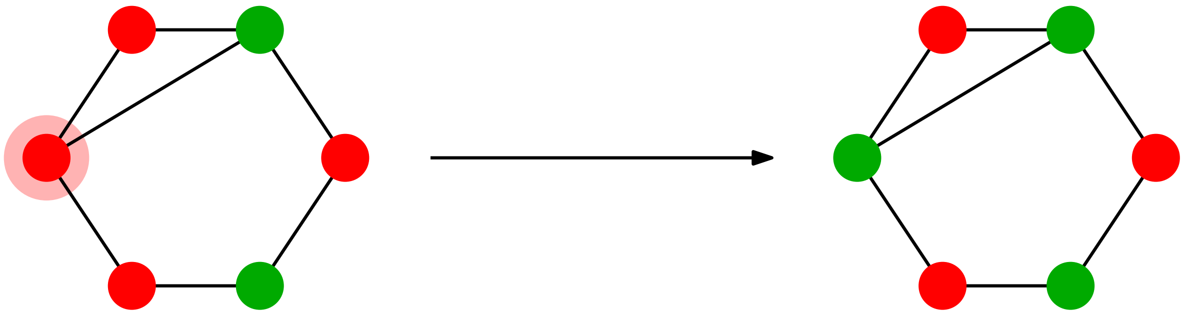


valid solution

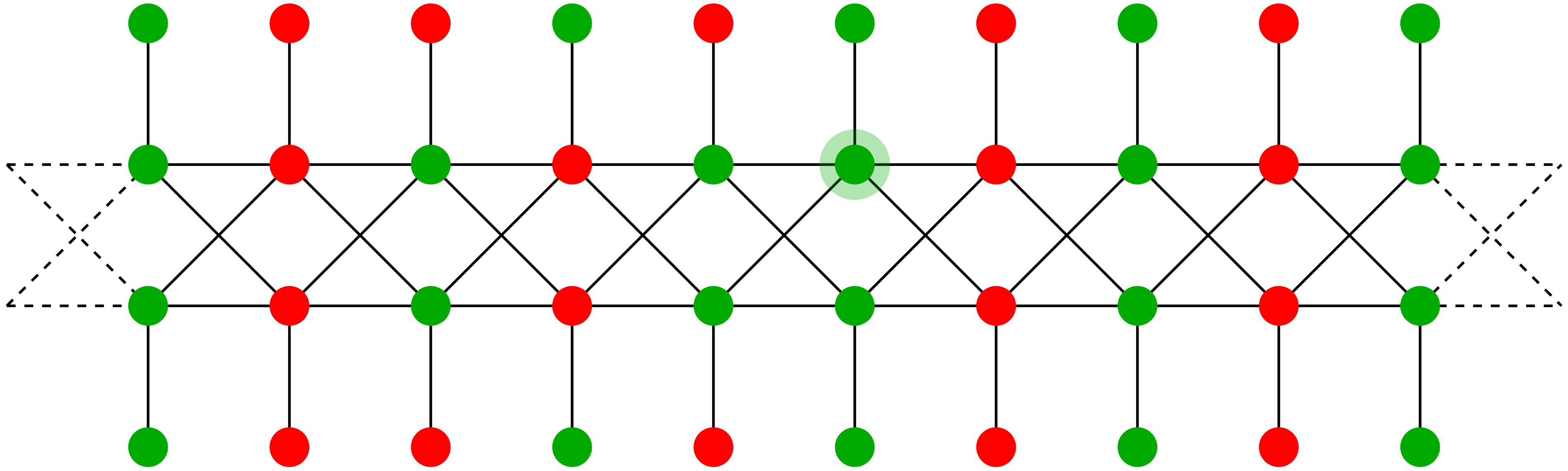


- **Convergence?**

Fixing procedure: *pick an arbitrary invalid node and flip its color*



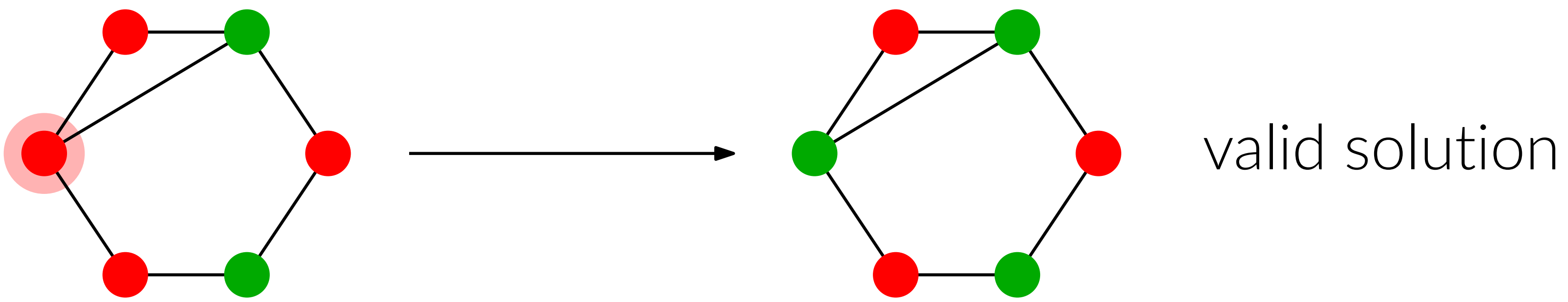
valid solution

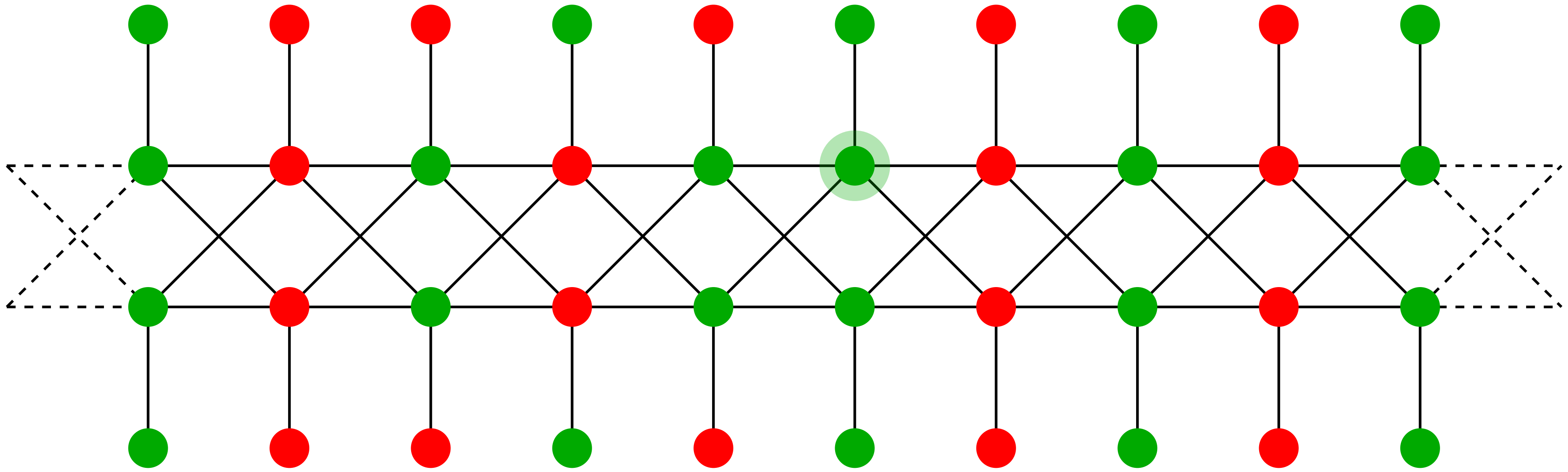


- **Convergence?** Potential function!

LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*



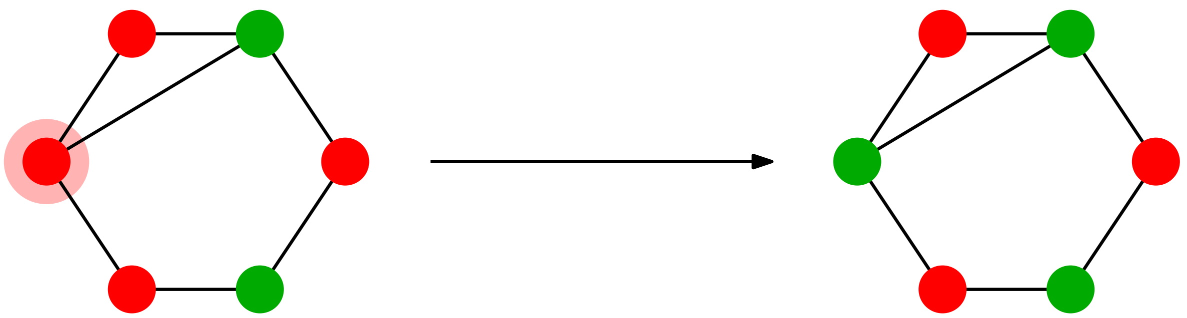


- **Convergence?** Potential function!

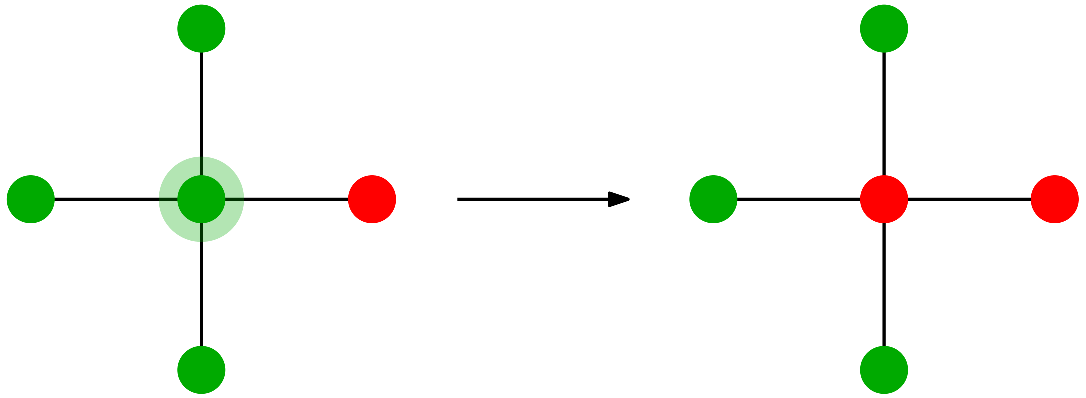
  - # of monochromatic edges decreases

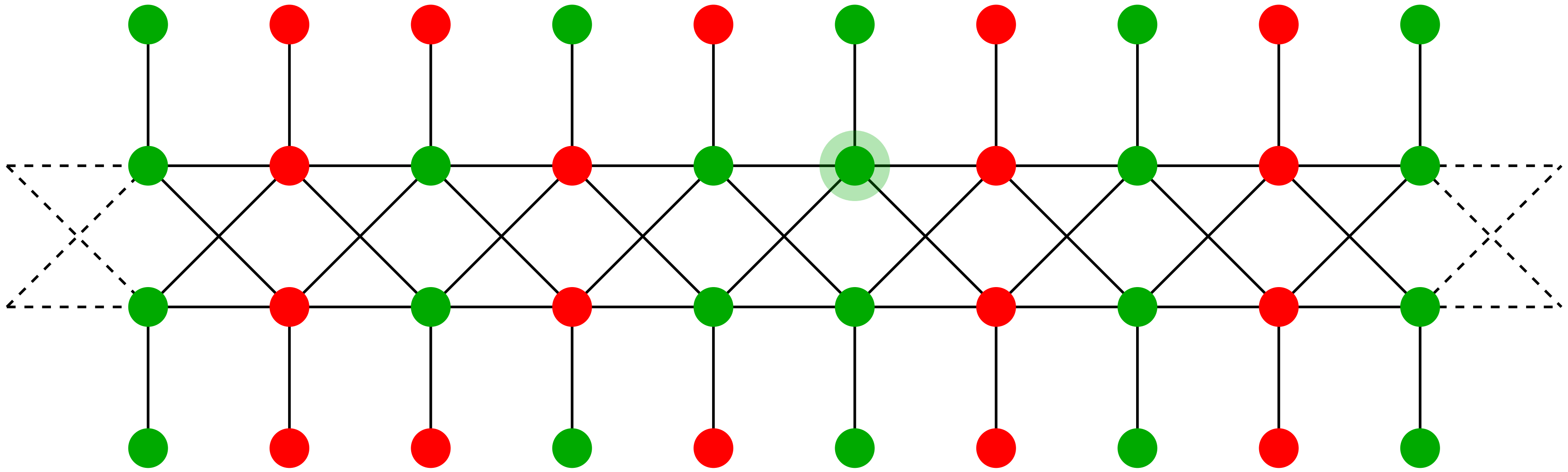
LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*



valid solution



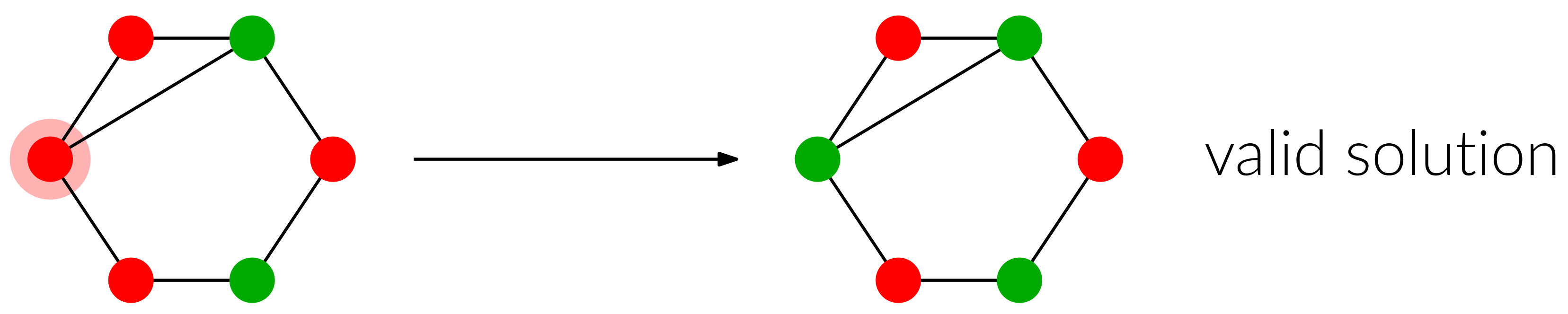


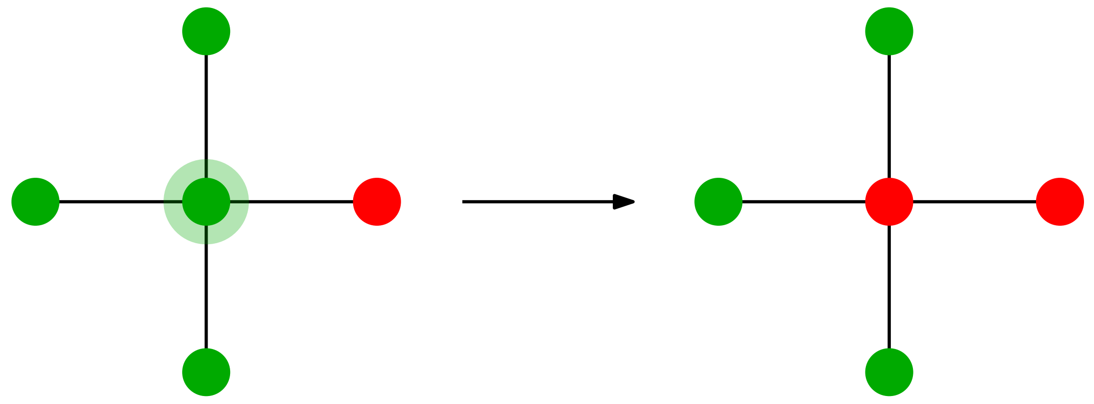
- **Convergence?** Potential function!

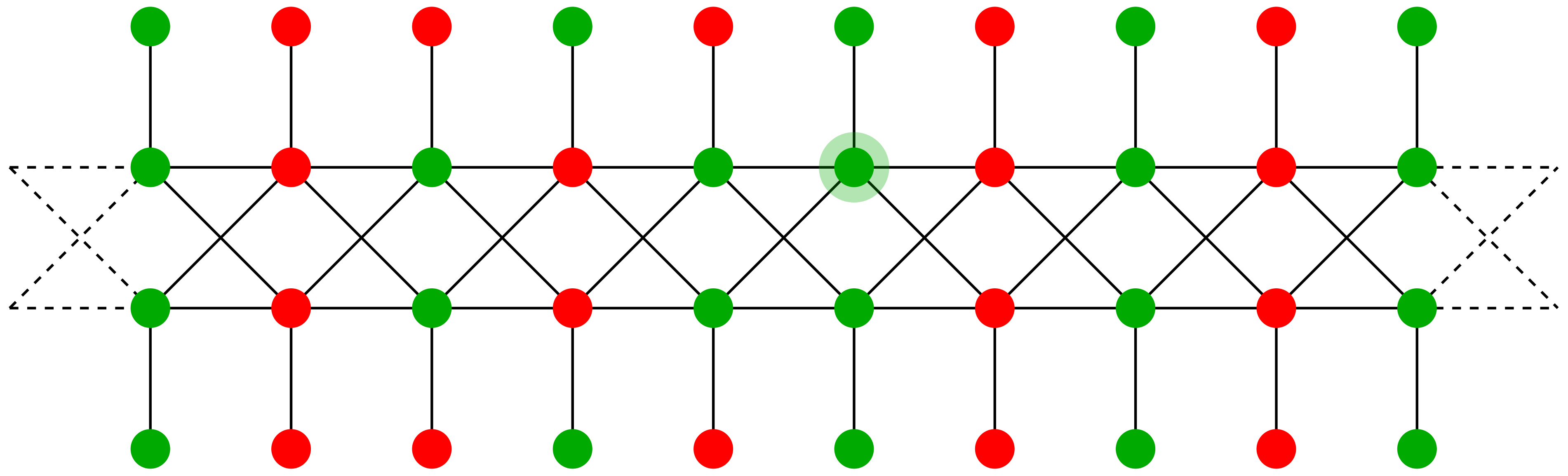
  - # of monochromatic edges decreases

LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*





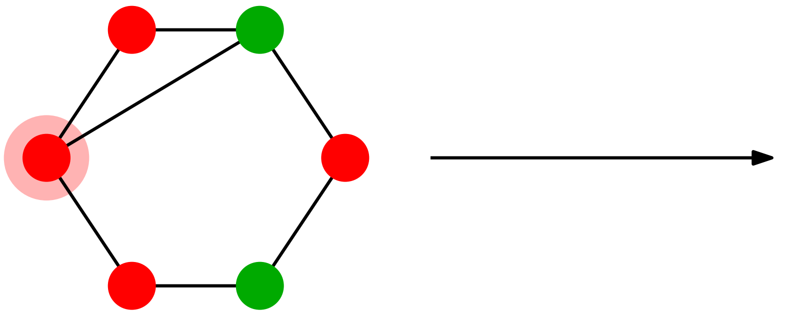


- **Convergence?** Potential function!

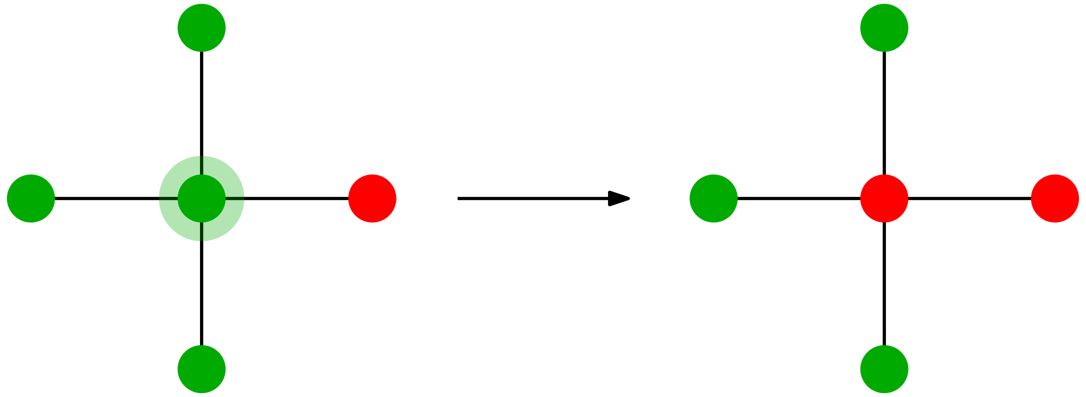
  \- # of monochromatic edges decreases

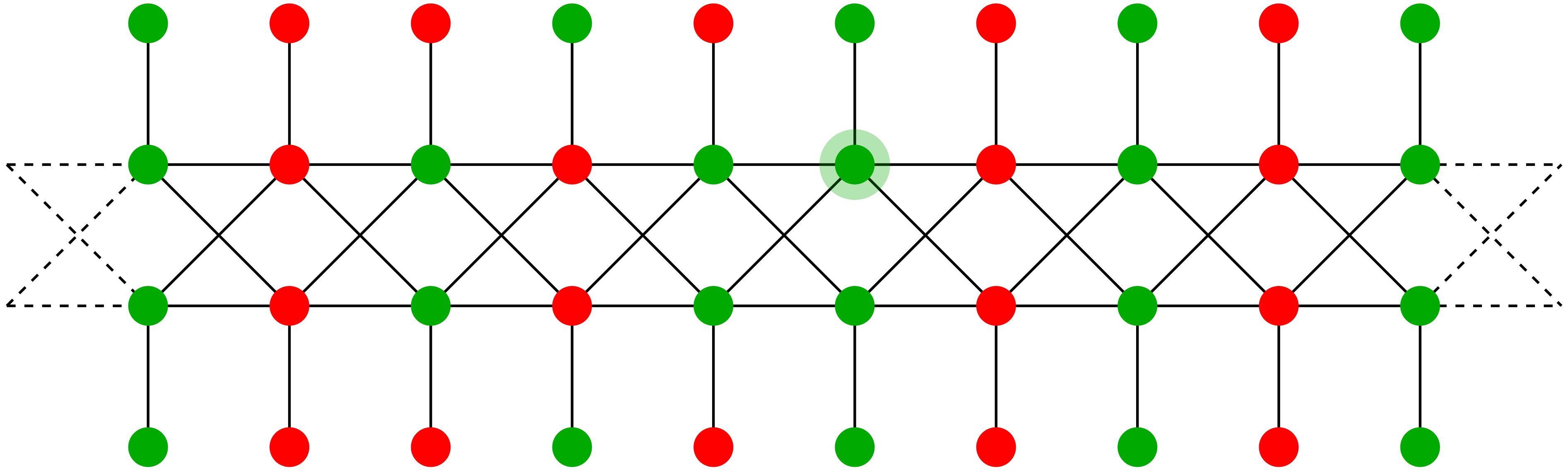
- Possibly, very long chains…

Fixing procedure: *pick an arbitrary invalid node and flip its color*



valid solution





- **Convergence?** Potential function!

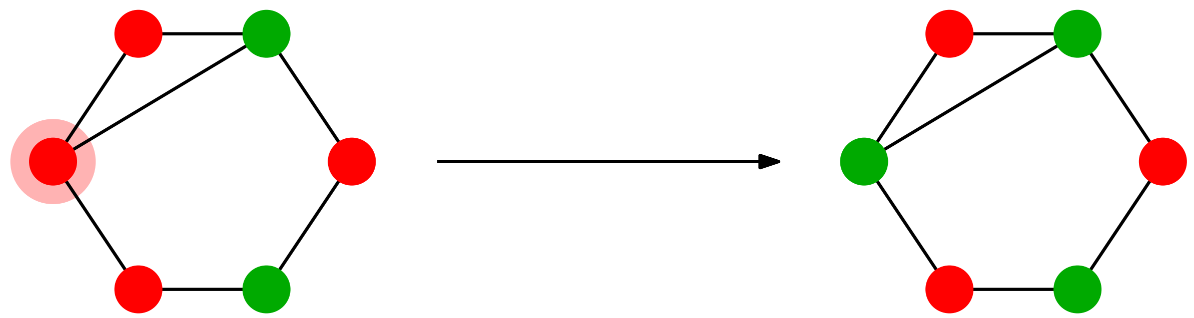
  - # of monochromatic edges decreases

- Possibly, very long chains…

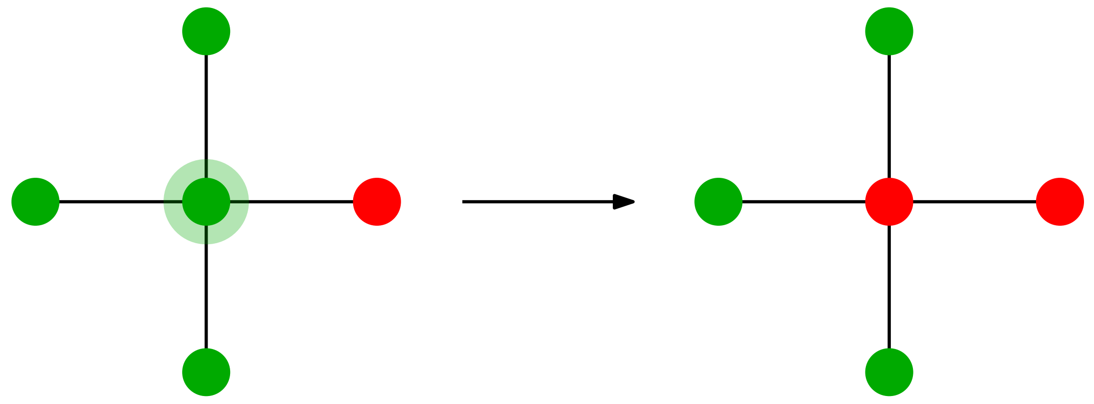
  - $O(|E|)$-time sequential algorithm

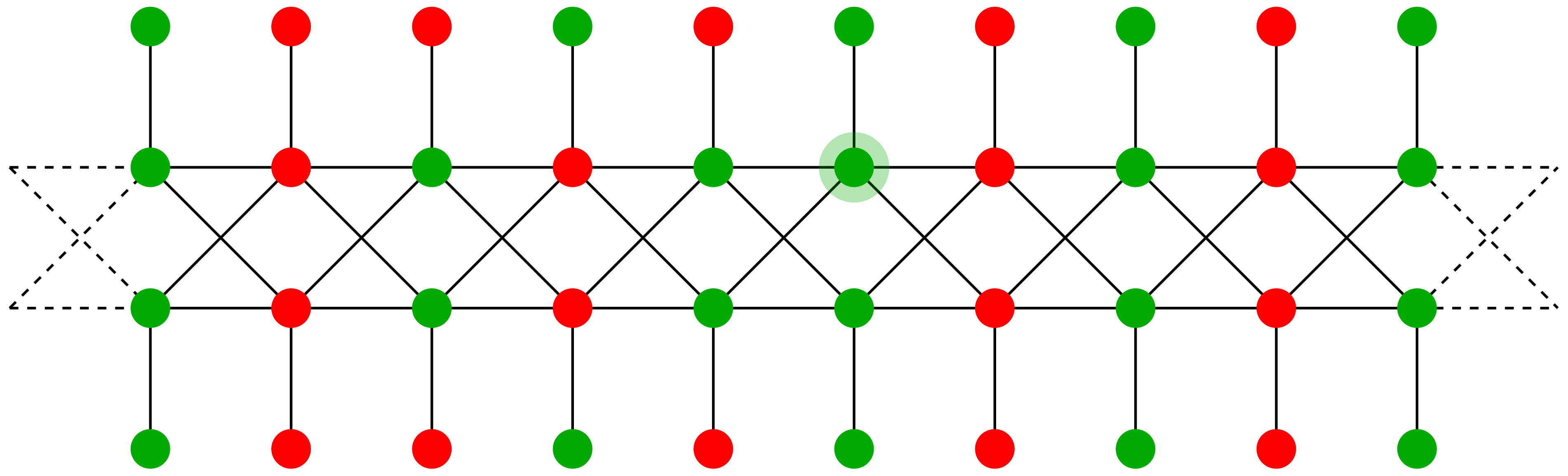
LOC: fixing procedure

Fixing procedure: *pick an arbitrary invalid node and flip its color*



valid solution

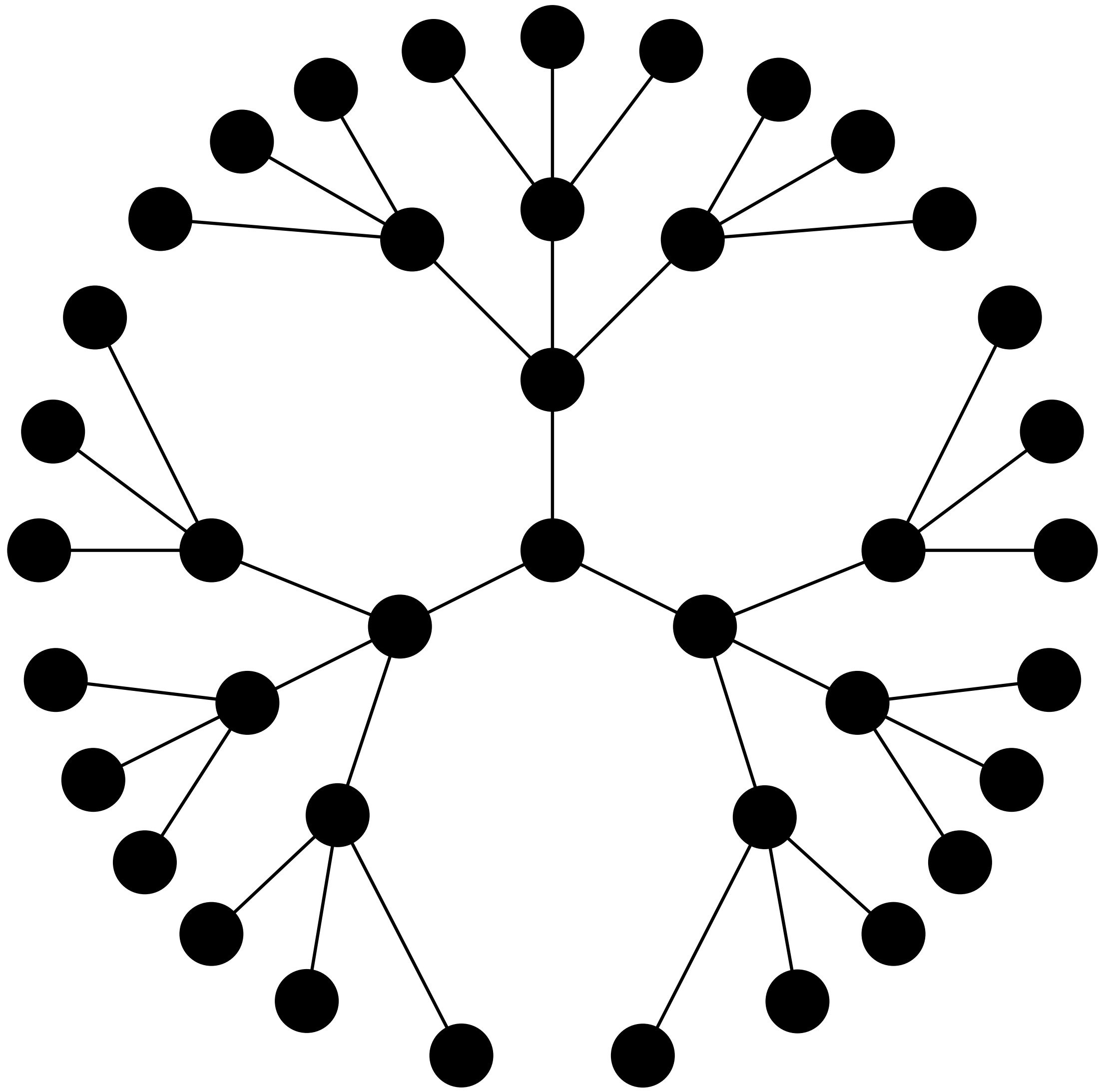




- **Convergence?** Potential function!
  - # of monochromatic edges decreases
- Possibly, very long chains…
  - $O(|E|)$-time sequential algorithm
- With distributed algorithms?

[Linial FOCS '87 & SICOMP '92]

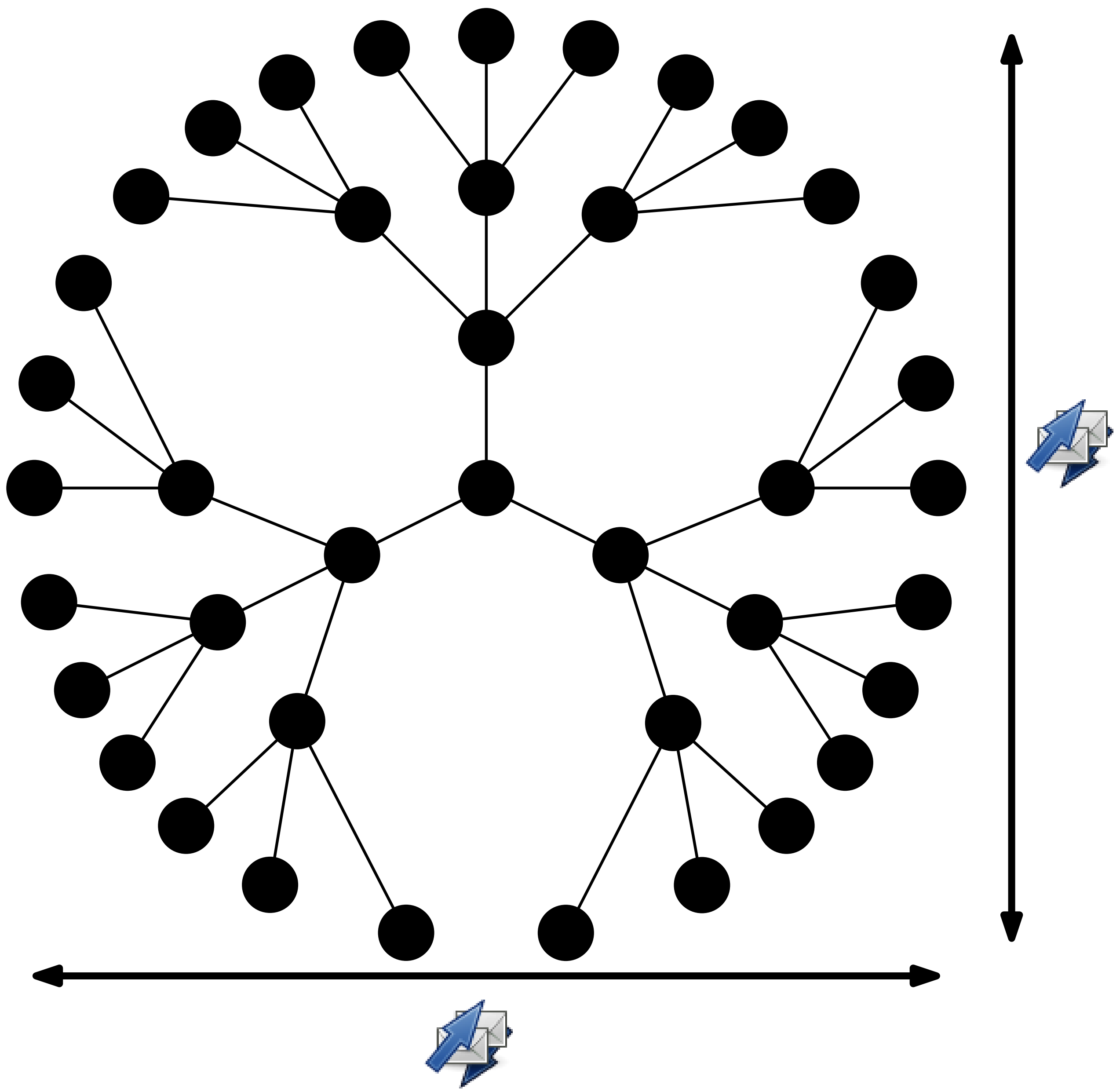
- **Distributed network** of n processors/nodes
  - graph $G = (V, E)$ with $|V| = n$
  - E: communication links
  - each node in V runs the same algorithm



The LOCAL model

[Linial FOCS '87 & SICOMP '92]

- **Distributed network** of n processors/nodes
  - graph $G = (V, E)$ with $|V| = n$
  - E: communication links
  - each node in V runs the same algorithm
- **Time is synchronous**: nodes alternate
  - arbitrary local computation & update of state variables
  - sending of messages to all neighbors
    - * no bandwidth constraints



[Linial FOCS '87 & SICOMP '92]

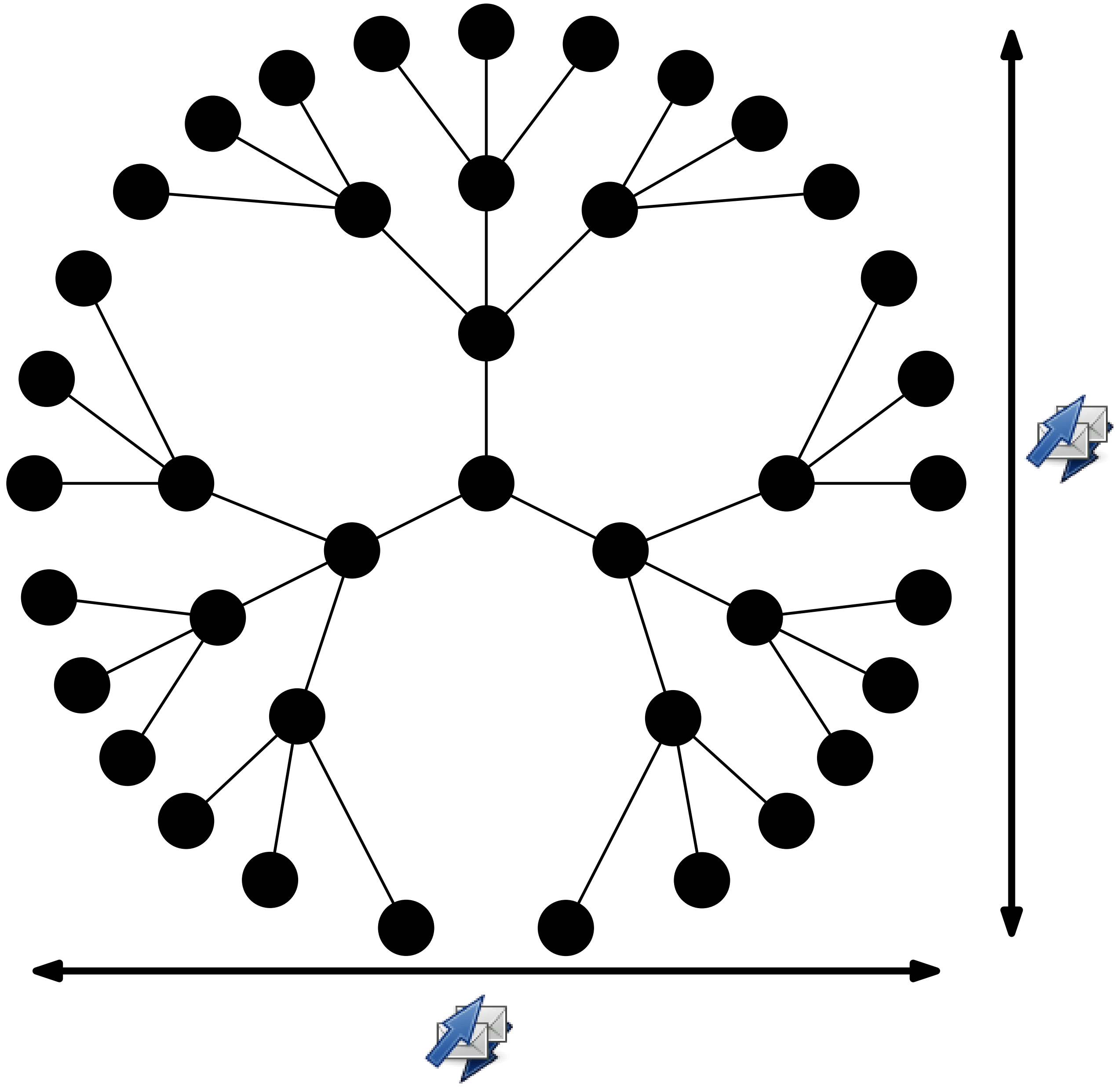
- **Distributed network** of n processors/nodes

  - graph $G = (V, E)$ with $|V| = n$
  - E: communication links
  - each node in V runs the same algorithm

- **Time is synchronous**: nodes alternate

  - arbitrary local computation & update of state variables
  - sending of messages to all neighbors
    * no bandwidth constraints

- **Unique identifiers** to nodes in the set $1, \ldots, \text{poly}(n)$
  * adversarially chosen * n is known to the nodes
  - needed to solve even basic problems (2-coloring a 2-path)



The LOCAL model

[Linial FOCS '87 & SICOMP '92]

- **Distributed network** of n processors/nodes

  - graph $G = (V, E)$ with $|V| = n$
  - E: communication links
  - each node in V runs the same algorithm

- **Time is synchronous**: nodes alternate

  - arbitrary local computation & update of state variables
  - sending of messages to all neighbors

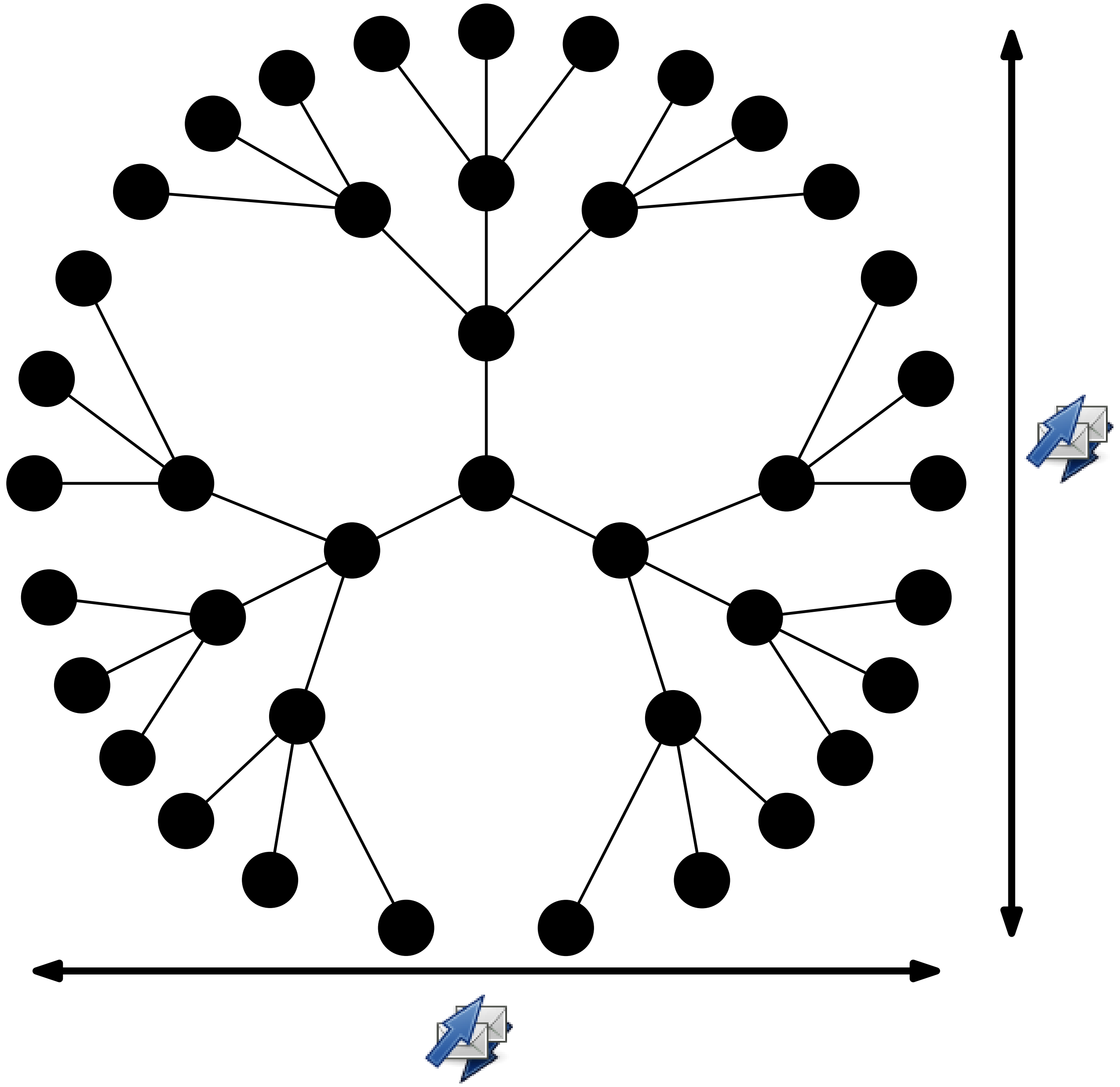
    * no bandwidth constraints

- **Unique identifiers** to nodes in the set $1, \ldots, \mathrm{poly}(n)$

  * adversarially chosen * n is known to the nodes

  - needed to solve even basic problems (2-coloring a 2-path)

- **Possible randomness**: i.i.d. infinite random bit strings to nodes



The LOCAL model

[Linial FOCS '87 & SICOMP '92]

- **Distributed network** of n processors/nodes

  - graph $G = (V, E)$ with $|V| = n$
  - E: communication links
  - each node in V runs the same algorithm

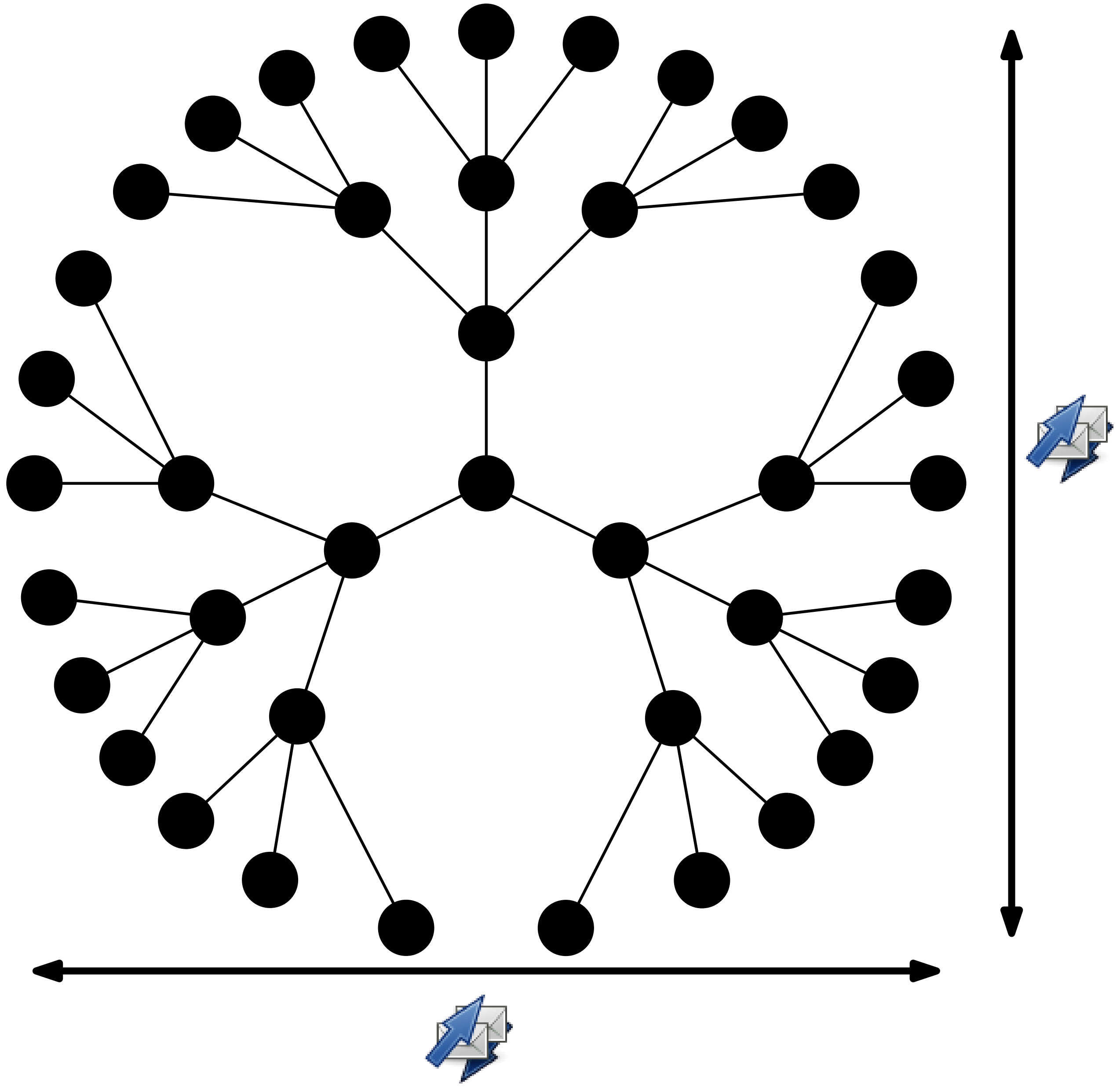
- **Time is synchronous**: nodes alternate

  - arbitrary local computation & update of state variables
  - sending of messages to all neighbors
    * no bandwidth constraints

- **Unique identifiers** to nodes in the set $1, \ldots, \text{poly}(n)$
  * adversarially chosen * n is known to the nodes
  - needed to solve even basic problems (2-coloring a 2-path)

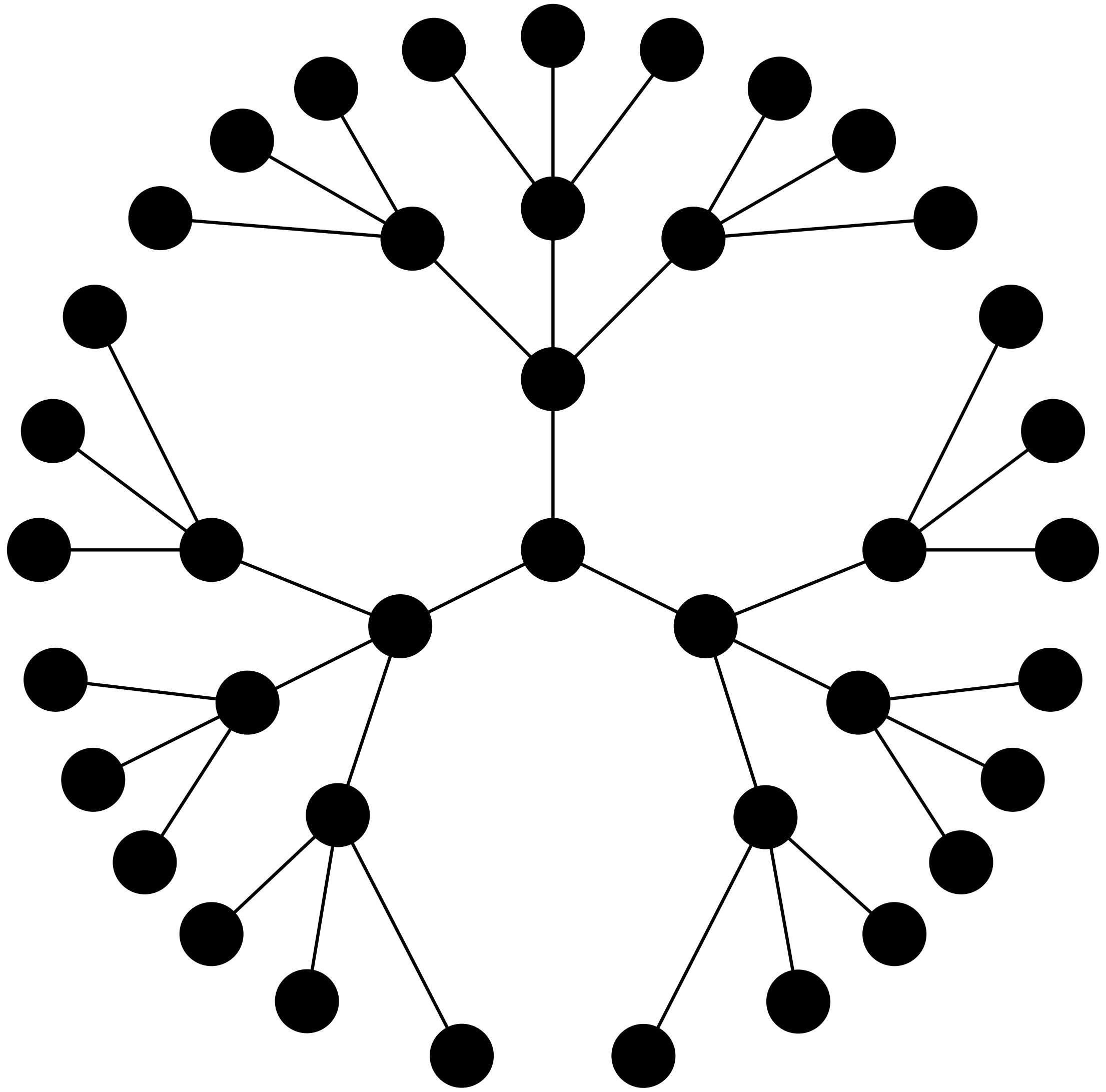
- **Possible randomness**: i.i.d. infinite random bit strings to nodes

- **Complexity measure**: number of communication rounds



Local view

Complexity measure: number of communication rounds

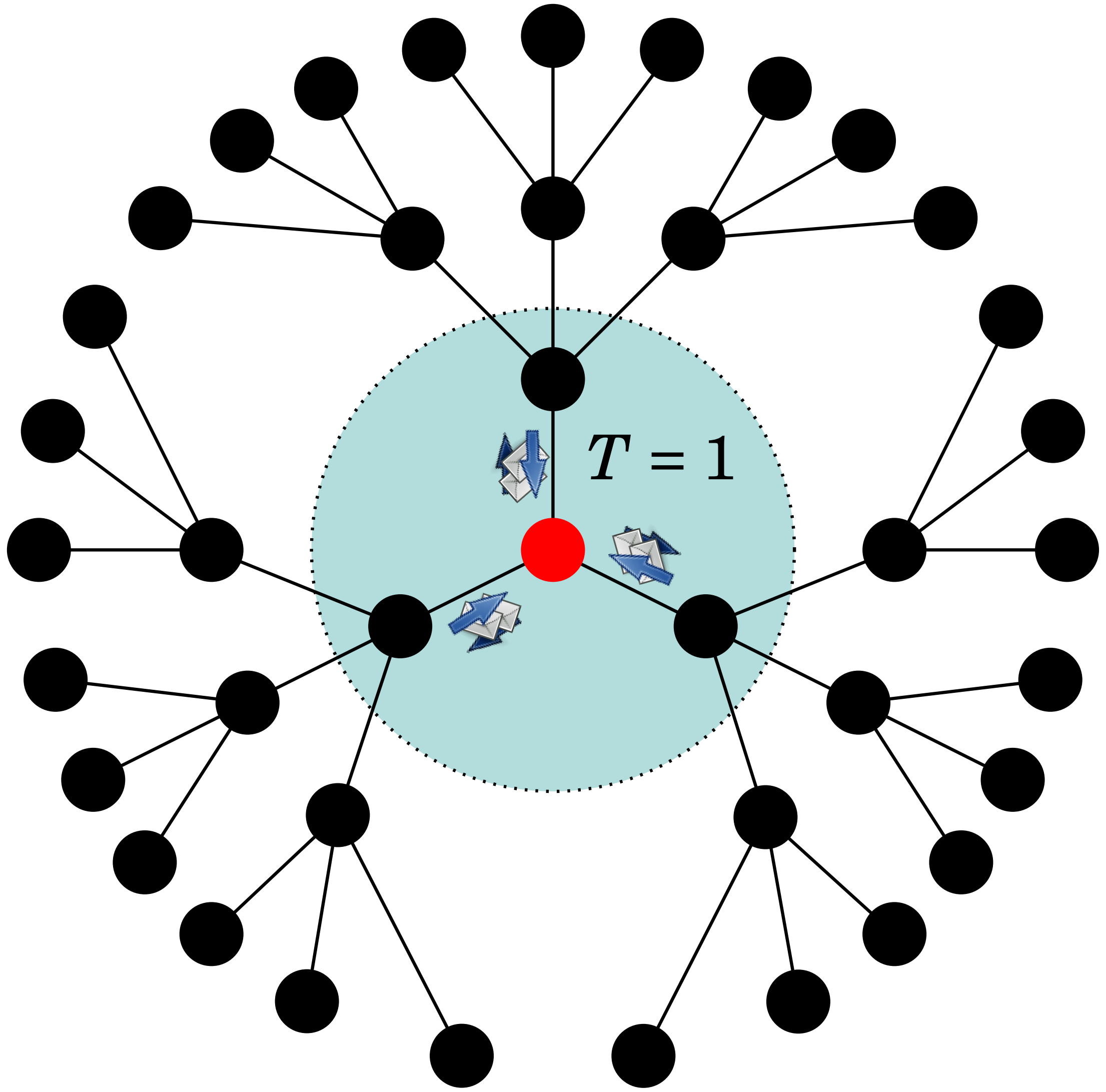


- What do we know after T rounds?

Local view

Complexity measure: number of communication rounds

- What do we know after $\boldsymbol{T}$ rounds?

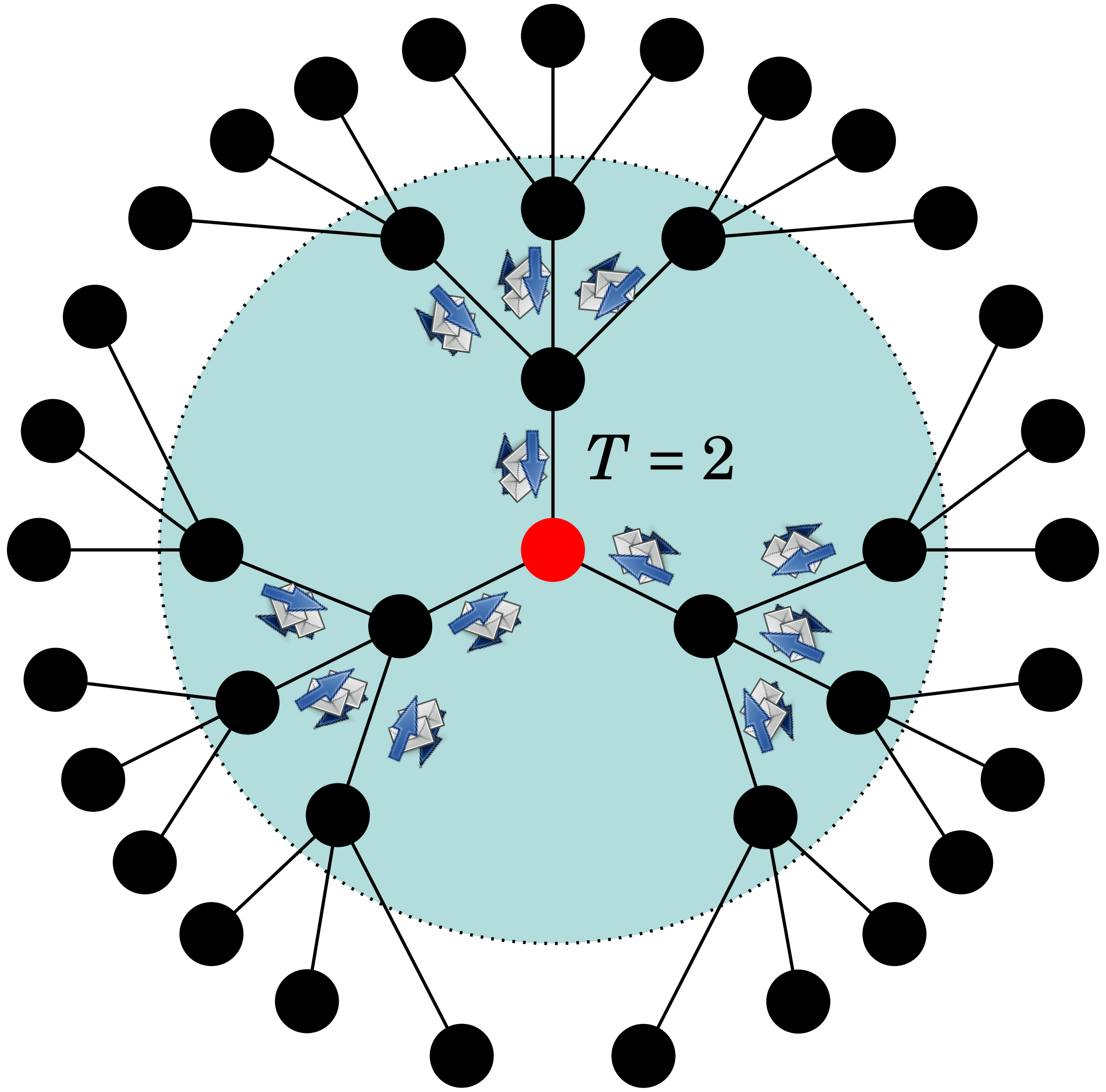


knowledge after $\boldsymbol{T}$ rounds of communication

Local view

Complexity measure: number of communication rounds

- What do we know after T rounds?

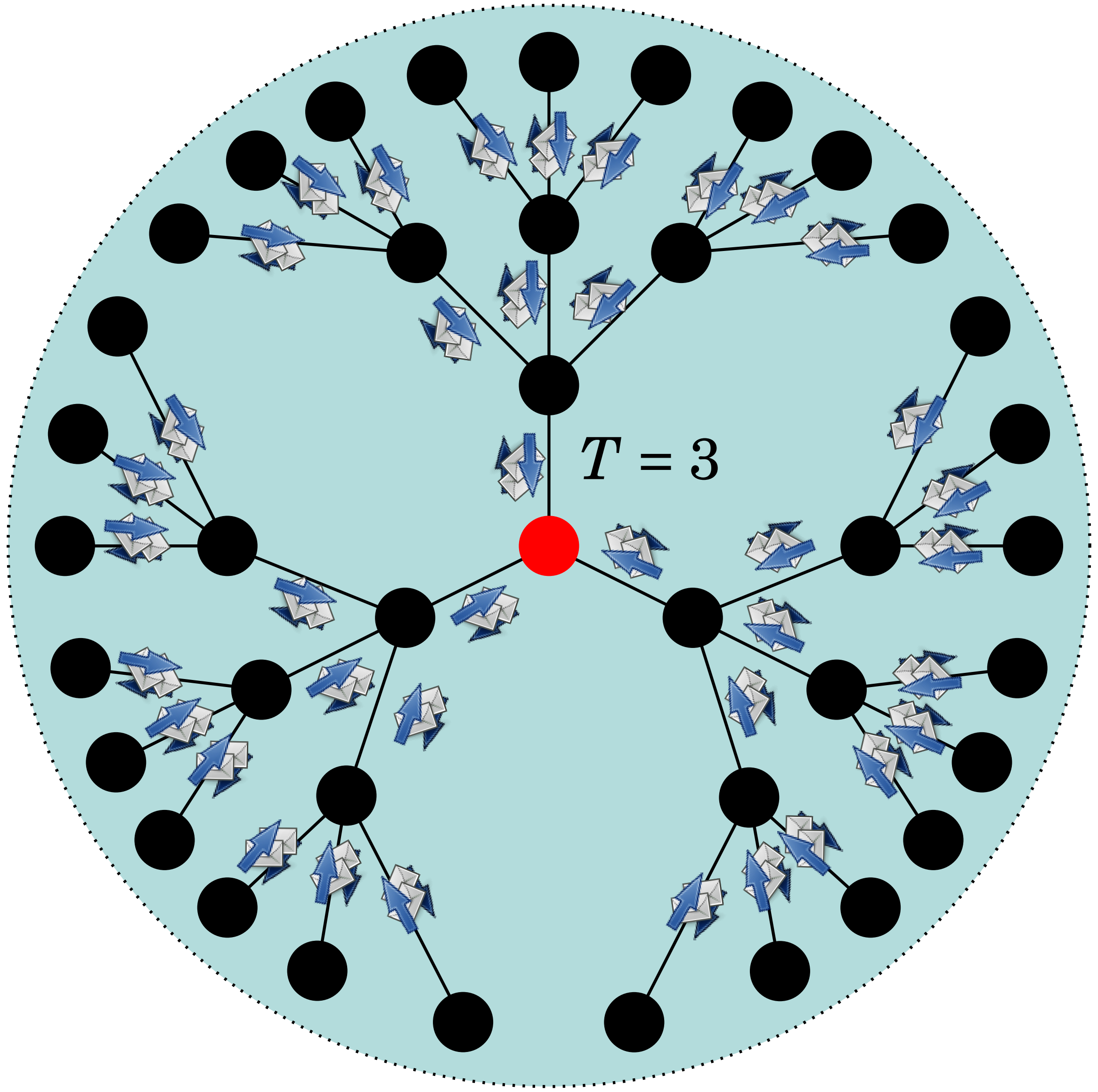


knowledge after T rounds of communication

Local view

Complexity measure: number of communication rounds

- What do we know after T rounds?

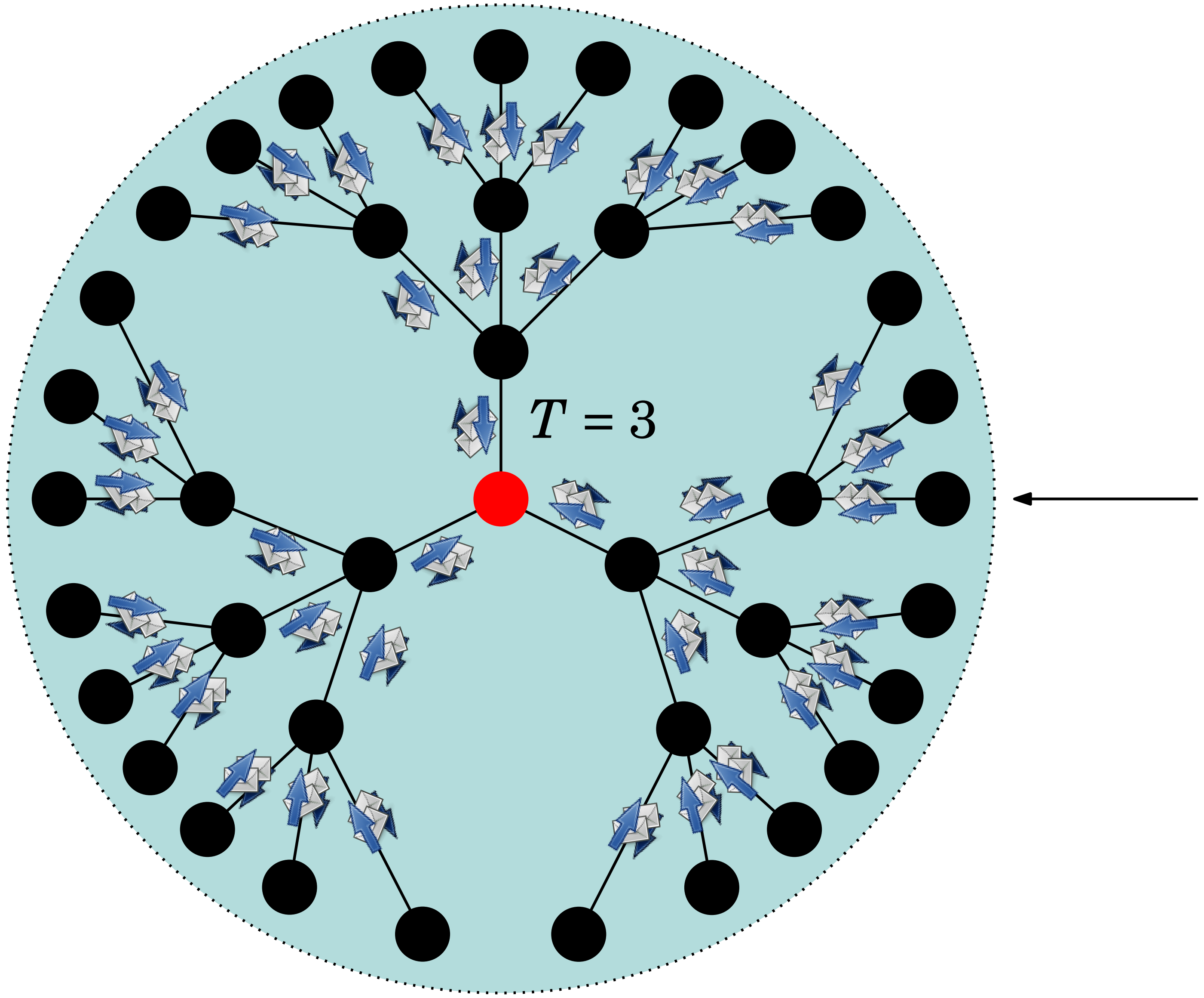


knowledge after T rounds of communication

Local view

Complexity measure: number of communication rounds

- What do we know after T rounds?



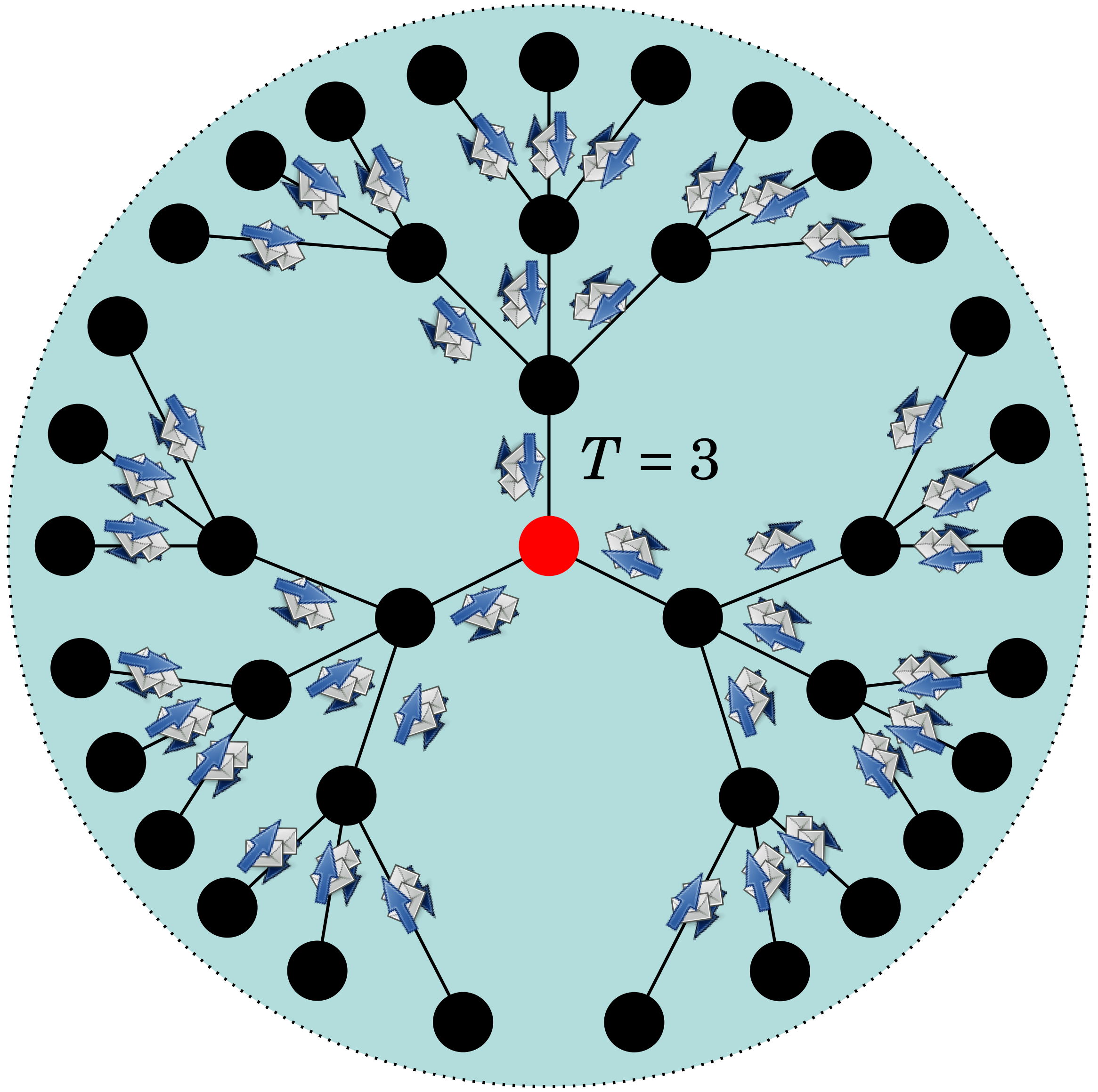
knowledge after T rounds of communication

- Locality $T = \mathrm{diam}(G) + 1$ is always sufficient to solve any problem: gathering algorithm

Local view

Complexity measure: number of communication rounds

- What do we know after T rounds?



knowledge after T rounds of communication

- Locality $T = \mathrm{diam}(G) + 1$ is always sufficient to solve any problem: gathering algorithm
- LOCAL algorithm A with locality T_A + LOCAL algorithm B with locality T_B = LOCAL algorithm C with locality $T_A + T_B$

Previous results about LOC

Lower bound: - $\Omega(\log n)$-rounds in deterministic LOCAL (in *bounded-degree* trees)

- $\Omega(\log \log n)$-rounds in randomized LOCAL (in *bounded-degree* trees)

- reduction from Sinkless Orientation [Balliu, Hirvonen, Lenzen, Olivetti, and Suomela, SIROCCO '19]
- fixed point in RE [Balliu, Brandt, Kuhn, Olivetti, and Saarhelo, DISC '25]

Previous results about LOC

Lower bound: - $\Omega(\log n)$-rounds in deterministic LOCAL (in *bounded-degree* trees)

- $\Omega(\log \log n)$-rounds in randomized LOCAL (in *bounded-degree* trees)

- reduction from Sinkless Orientation [Balliu, Hirvonen, Lenzen, Olivetti, and Suomela, SIROCCO '19]
- fixed point in RE [Balliu, Brandt, Kuhn, Olivetti, and Saarhelo, DISC '25]

Upper bound: - $O(n)$-rounds in both deterministic and randomized LOCAL (even in bounded-degree graphs)

Previous results about LOC

Lower bound: - $\Omega(\log n)$-rounds in deterministic LOCAL (in *bounded-degree* trees)

- $\Omega(\log\log n)$-rounds in randomized LOCAL (in *bounded-degree* trees)

- reduction from Sinkless Orientation [Balliu, Hirvonen, Lenzen, Olivetti, and Suomela, SIROCCO '19]
- fixed point in RE [Balliu, Brandt, Kuhn, Olivetti, and Saarhelo, DISC '25]

Upper bound: - $O(n)$-rounds in both deterministic and randomized LOCAL (even in bounded-degree graphs)

HUGE GAP!

Let's find a *better* distributed algorithm...
(for bounded-degree graphs)

MPX subroutine

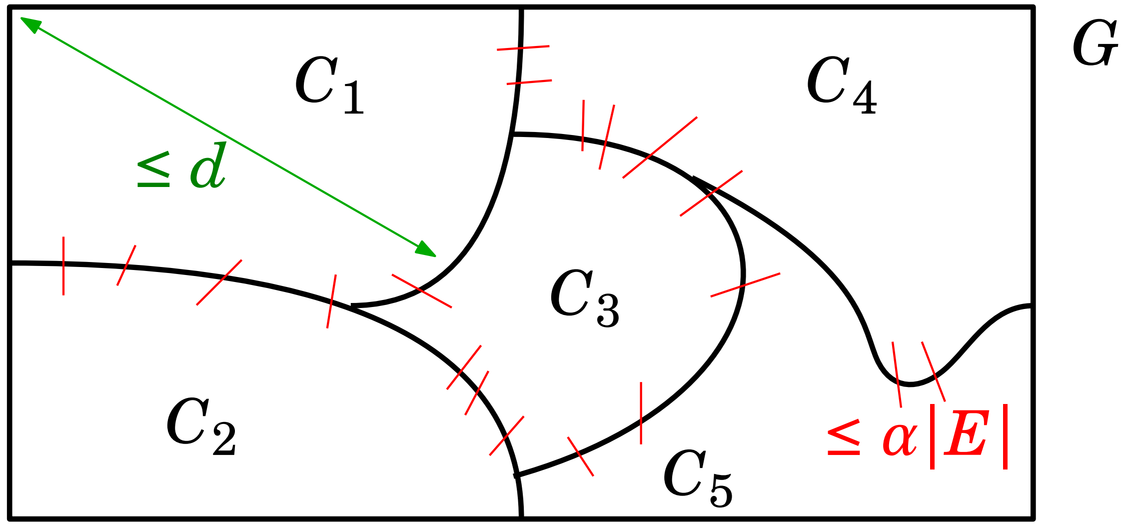
(α, d)-**decomposition** of a graph $G = (V, E)$:

- partition of V into clusters (sets) $C_1, \ldots, C_k$
- $\mathrm{diam}(C_i) \leq d$ for all i
- # inter-clusters edges $\leq \alpha |E|$

G

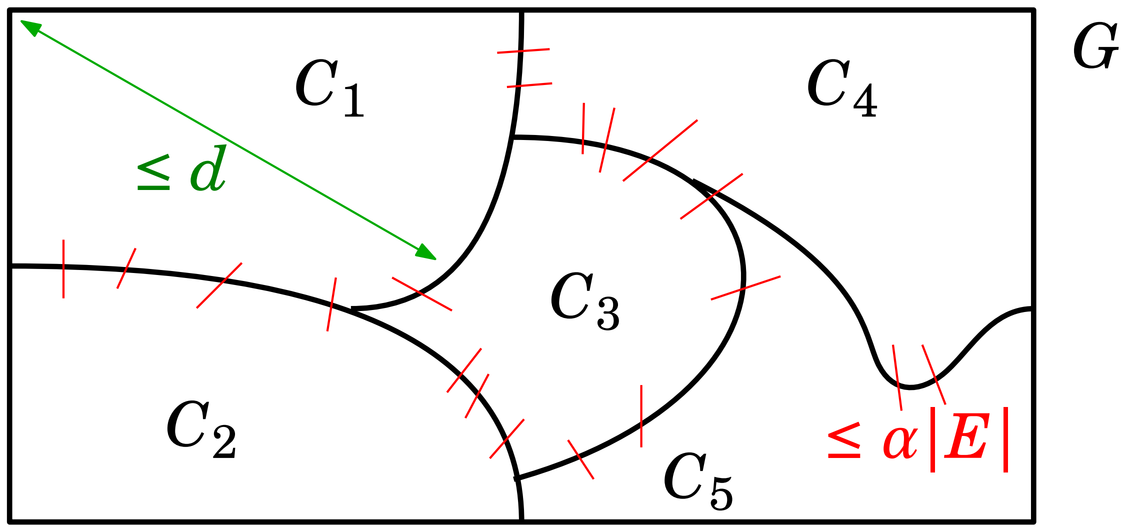
(α, d)-**decomposition** of a graph $G = (V, E)$:

- partition of V into clusters (sets) $C_1, \dots, C_k$
- $\mathrm{diam}(C_i) \le d$ for all i
- # inter-clusters edges $\le \alpha|E|$



(α, d)-**decomposition** of a graph $G = (V, E)$:

- partition of V into clusters (sets) $C_1, \ldots, C_k$
- $\mathrm{diam}(C_i) \leq d$ for all i
- # inter-clusters edges $\leq \alpha|E|$

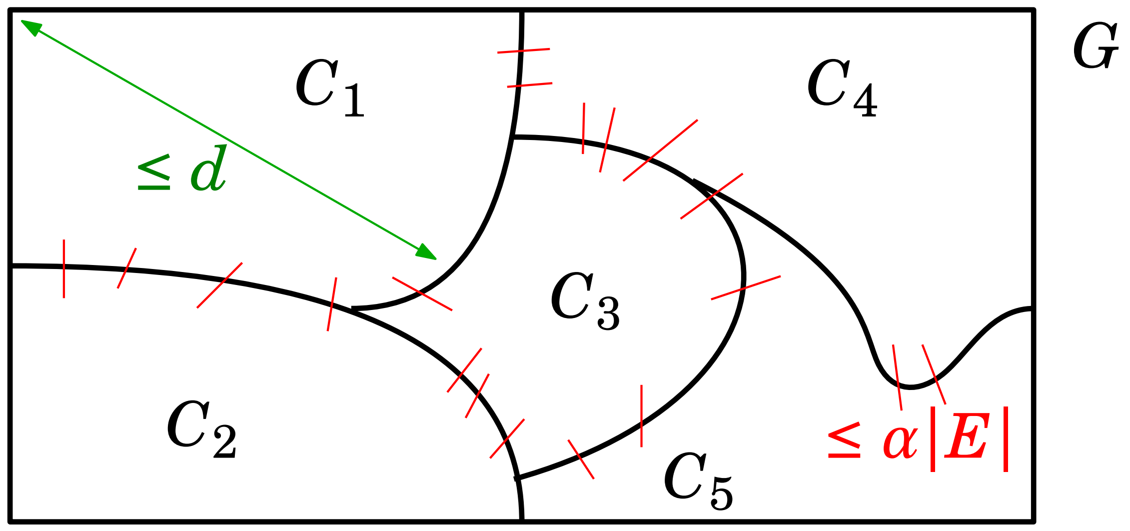


Theorem (adaptation of [Miller, Peng, and Xu, SPAA '13]):

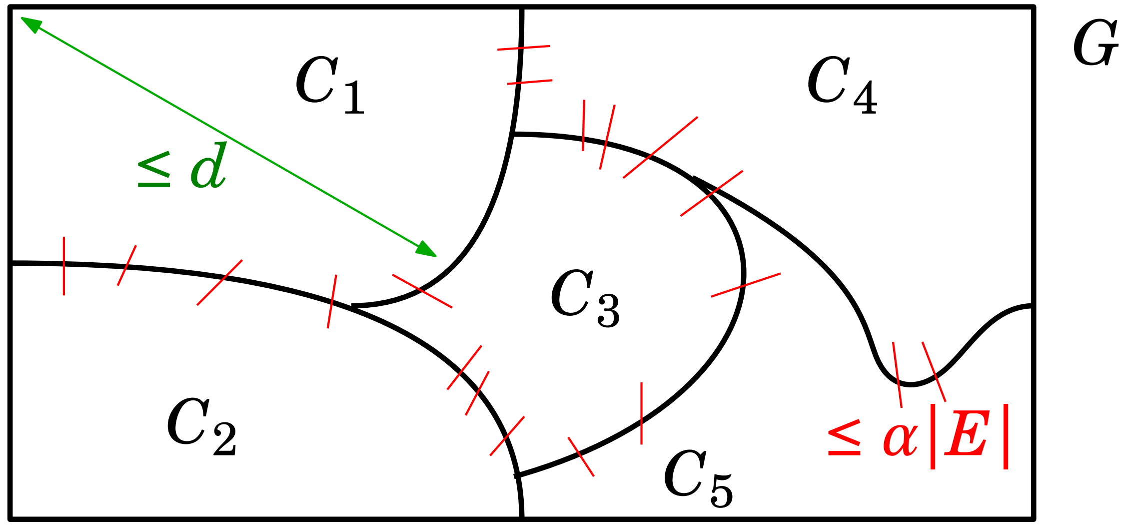
There exists a randomized LOCAL algorithm $\mathcal{MPX}$ that computes an (α, d)-decomposition of a graph $G = (V, E)$ with the following properties:

- *Running time $O(\log n/\alpha)$.*
- *UB on the diameter is $d = O(\log n/\alpha)$.*
- *For each $v \in V$, with probability $\geq 1/2$ it holds that $\mathcal{N}_{\Theta(1/\alpha)}[v] \subseteq C_i$ for some i.*

A "simple" algorithm

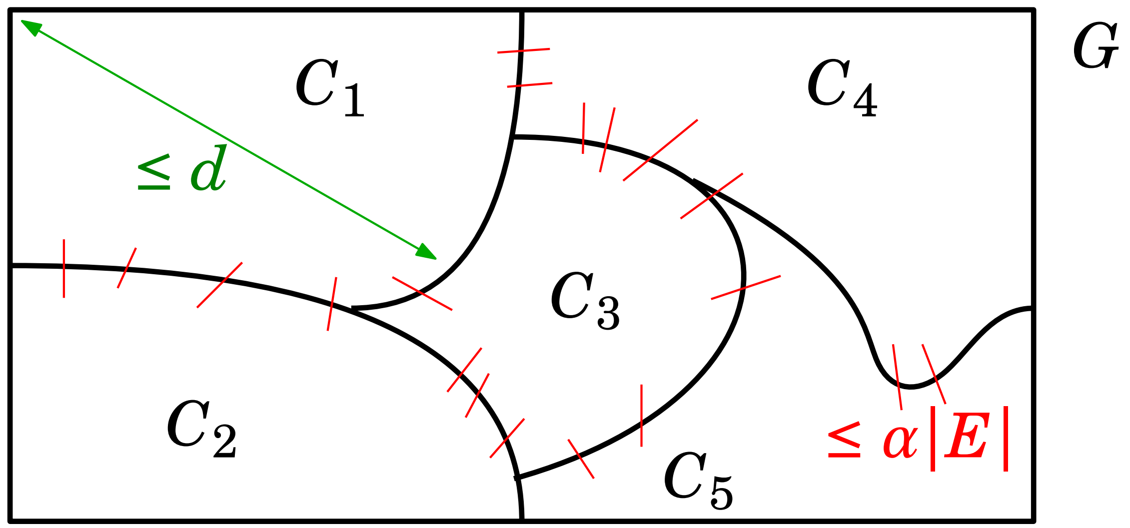


A "simple" algorithm



- *Brute-force solution with minimum potential in each cluster (time* $\boldsymbol{O(d)}$*)*

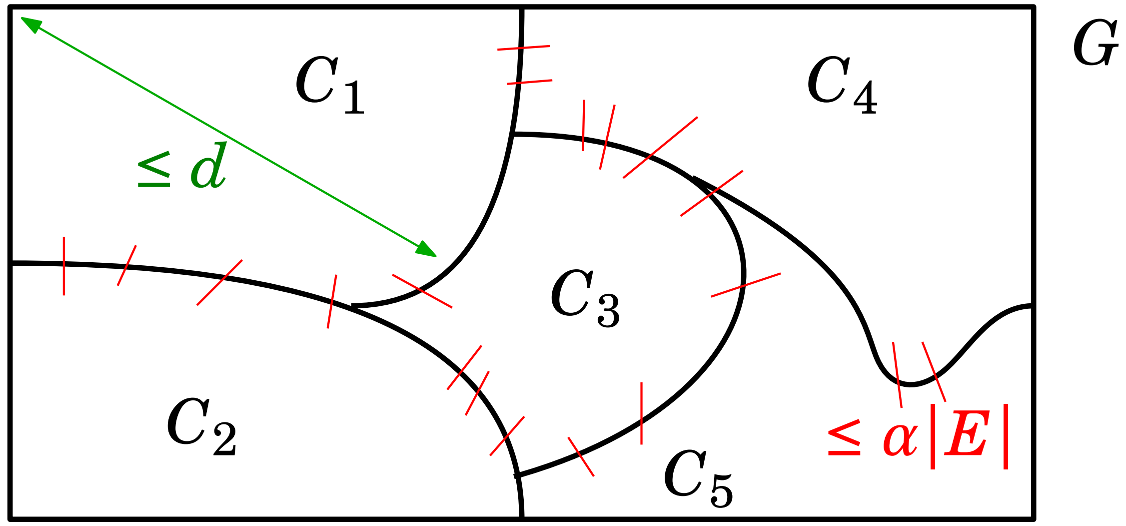
A “simple” algorithm



- *Brute-force solution with minimum potential in each cluster (time $O(d)$)*

  - distance from global minimum of the potential is $O(\alpha|E|)$

A "simple" algorithm

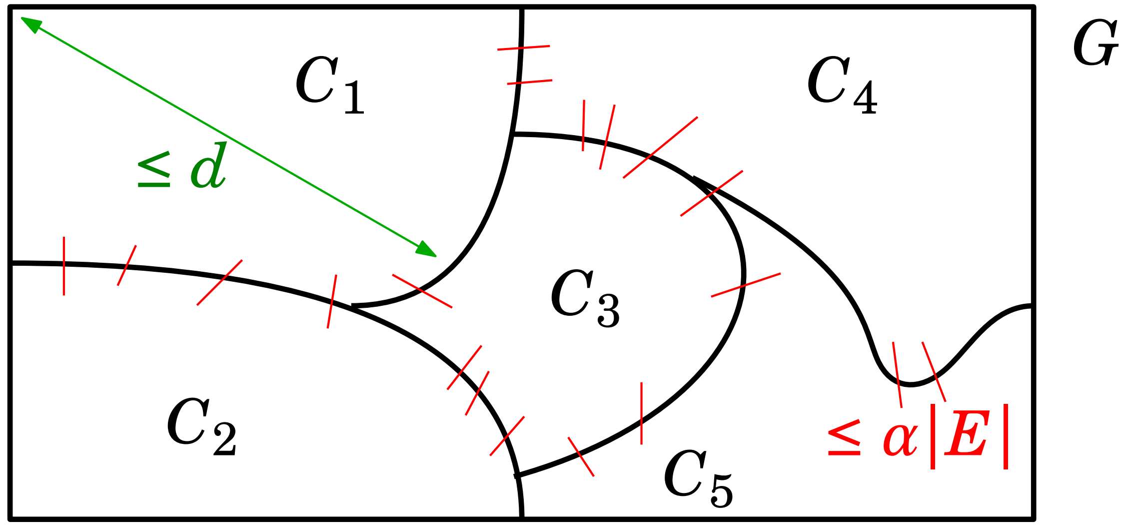


- *Brute-force solution with minimum potential in each cluster (time $\boldsymbol{O(d)}$)*

  - distance from global minimum of the potential is $\boldsymbol{O(\alpha|E|)}$

- *Simulate "fixing procedure" (at most $\boldsymbol{O(\alpha|E|)}$ rounds)*

A "simple" algorithm



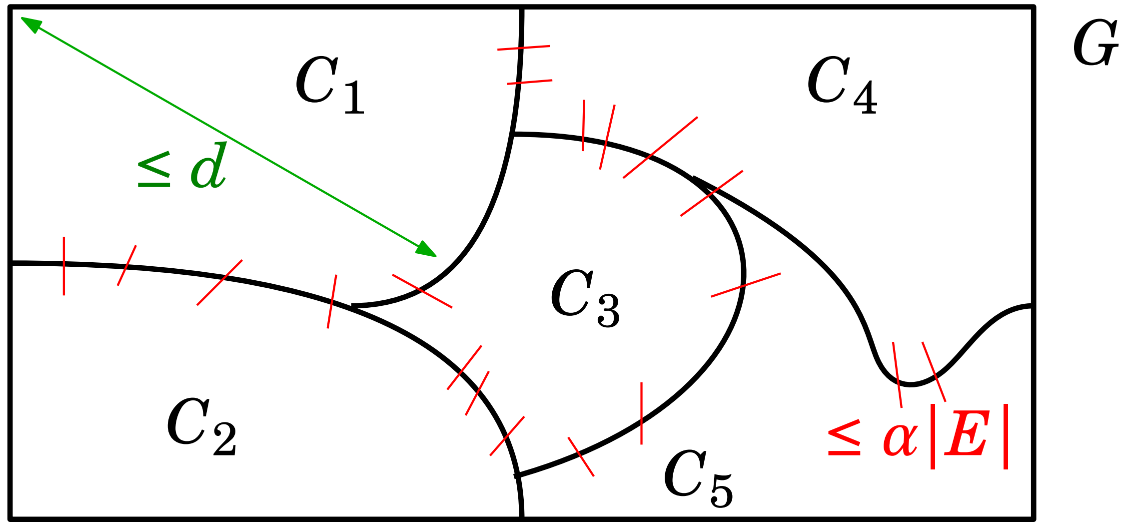
- *Brute-force solution with minimum potential in each cluster (time* $\boldsymbol{O(d)}$*)*

  - distance from global minimum of the potential is $\boldsymbol{O(\alpha|E|)}$

- *Simulate "fixing procedure" (at most* $\boldsymbol{O(\alpha|E|)}$ *rounds)*

Overall running time: $\text{cost}(\mathcal{MPX}) + O(d) + O(\alpha|E|) = O(\log n/\alpha + \alpha|E|)$

A "simple" algorithm



- *Brute-force solution with minimum potential in each cluster (time $O(d)$)*

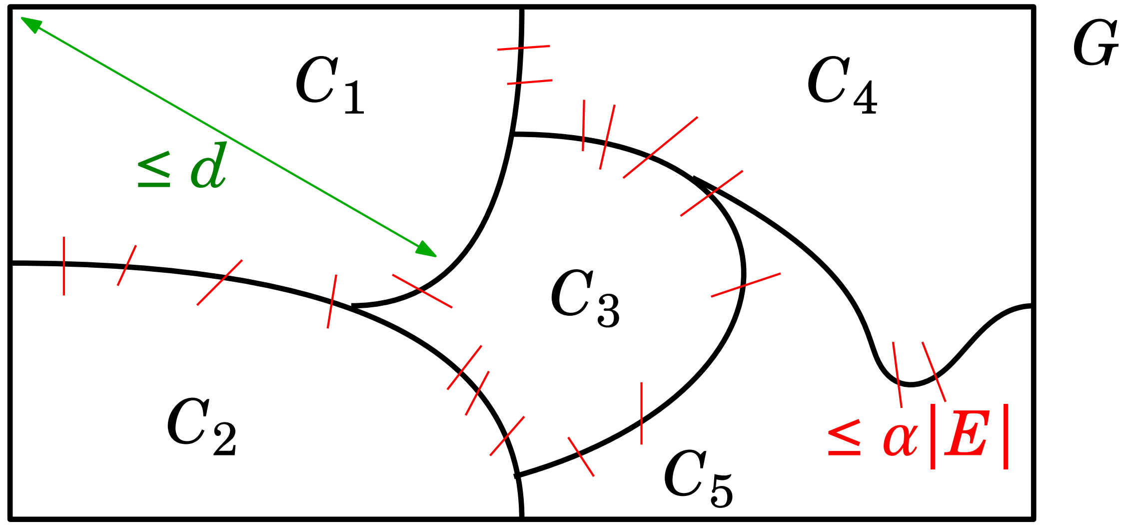
  - distance from global minimum of the potential is $O(\alpha|E|)$

- *Simulate "fixing procedure" (at most $O(\alpha|E|)$ rounds)*

Overall running time: $\text{cost}(\mathcal{MPX}) + O(d) + O(\alpha|E|) = O(\log n/\alpha + \alpha|E|)$

- bounded-degree graphs: running time $O(\sqrt{n\log n})$ (minimized by $\alpha = \sqrt{\log n/n}$)

A “simple” algorithm



- *Brute-force solution with minimum potential in each cluster (time $O(d)$)*

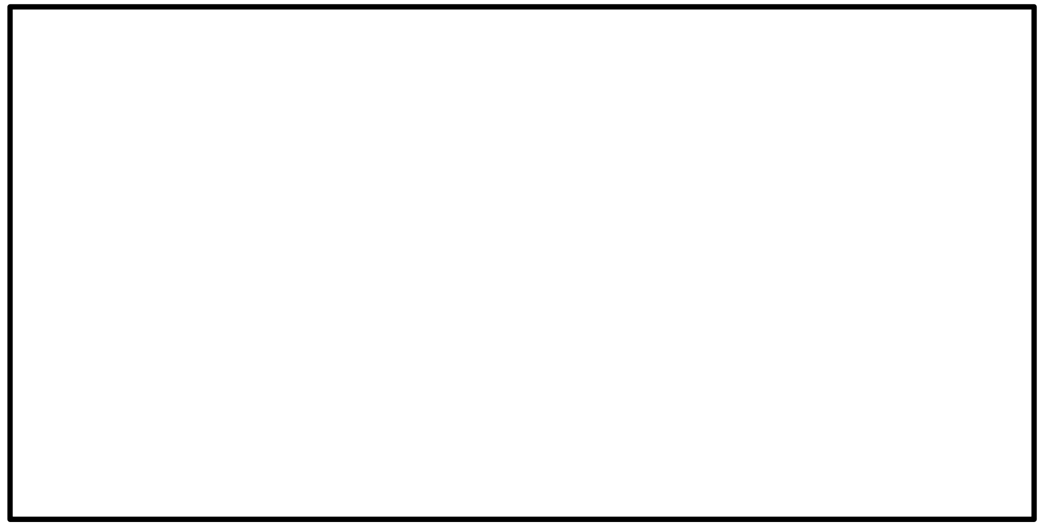
  - distance from global minimum of the potential is $O(\alpha|E|)$

- *Simulate “fixing procedure” (at most $O(\alpha|E|)$ rounds)*

Overall running time: cost$(\mathcal{MPX}) + O(d) + O(\alpha|E|) = O(\log n/\alpha + \alpha|E|)$

- bounded-degree graphs: running time $O(\sqrt{n \log n})$ (minimized by $\alpha = \sqrt{\log n/n}$)
- still far from the lower bounds … How to do better?

Repeating “brute-force” does not work



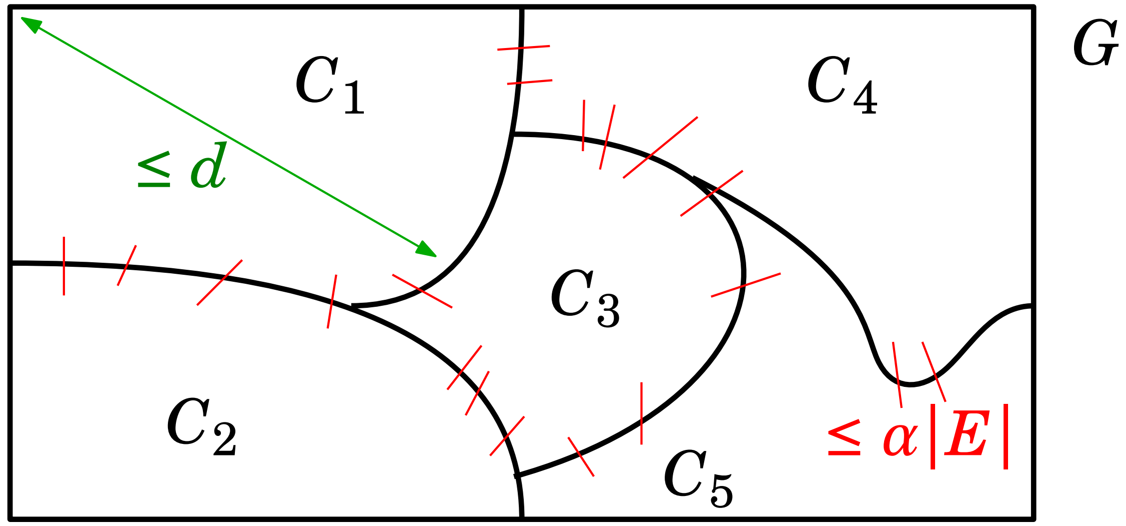
G

- *Repeat:*

  - *Run $\mathcal{MPX}$ to get (α, d)-network decomposition (time $O(d)$)*

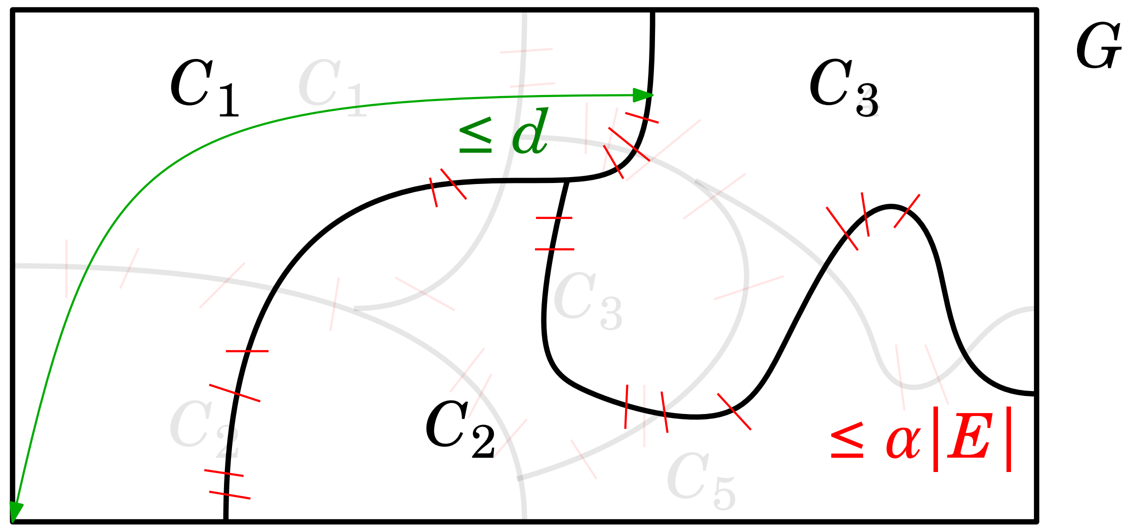
  - *Brute-force solution with minimum potential in each cluster (time $O(d)$)*

Repeating "brute-force" does not work



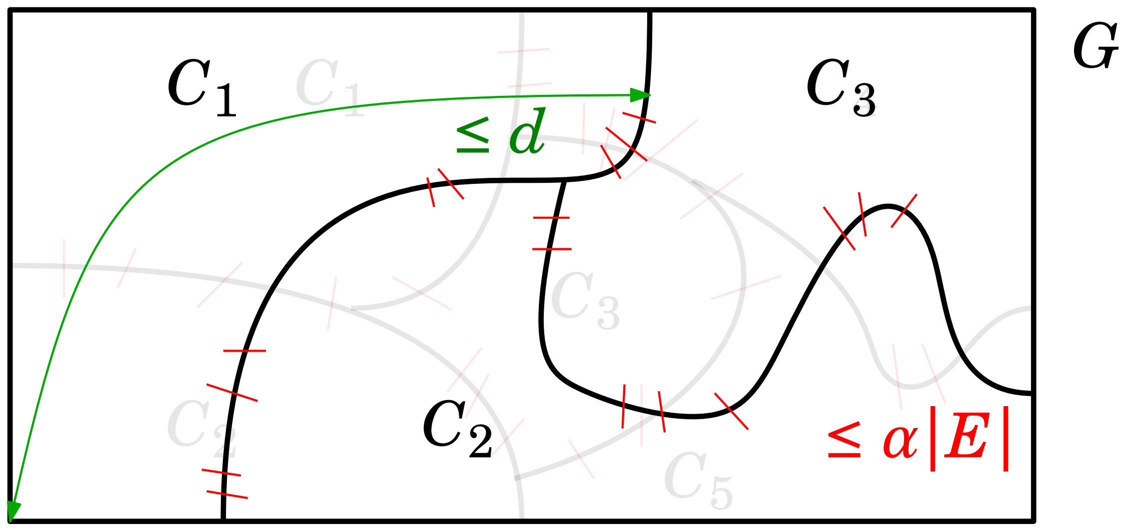
- *Repeat:*
  - *Run $\mathcal{MPX}$ to get (α, d)-network decomposition (time $O(d)$)*
  - *Brute-force solution with minimum potential in each cluster (time $O(d)$)*

Repeating “brute-force” does not work



- *Repeat:*
  - *- Run $\mathcal{MPX}$ to get (α, d)-network decomposition (time $O(d)$)*
  - *- Brute-force solution with minimum potential in each cluster (time $O(d)$)*

Repeating "brute-force" does not work



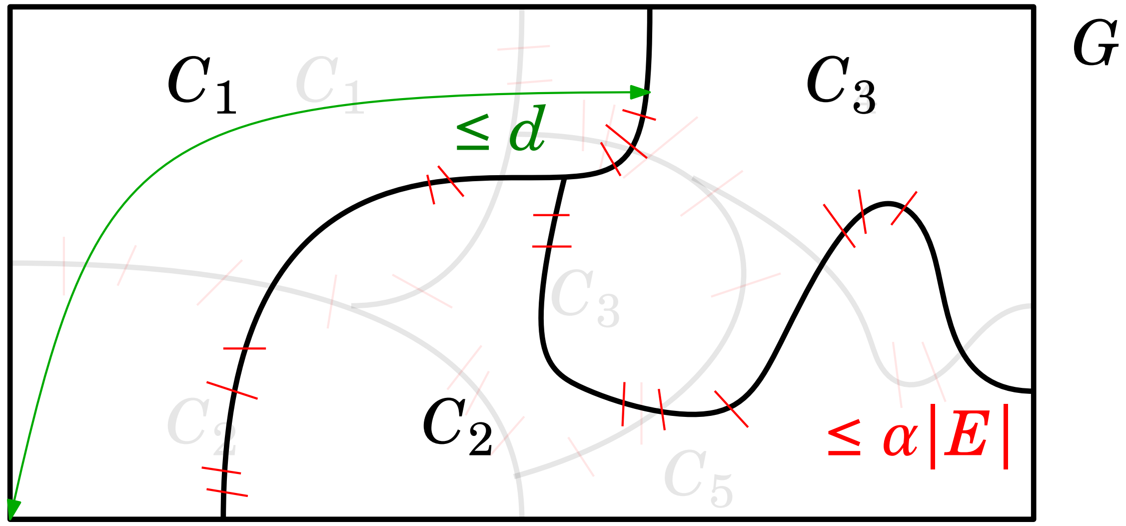
- *Repeat:*

  - *Run $\mathcal{MPX}$ to get (α, d)-network decomposition (time $O(d)$)*

  - *Brute-force solution with minimum potential in each cluster (time $O(d)$)*

- Distance from global minimum of the potential keeps at $O(\alpha|E|)$

Repeating "brute-force" does not work



- *Repeat:*
  - *Run $\mathcal{MPX}$ to get (α, d)-network decomposition (time $O(d)$)*
  - *Brute-force solution with minimum potential in each cluster (time $O(d)$)*
- Distance from global minimum of the potential keeps at $O(\alpha|E|)$
  - what to do?

Improving sets

Improving set in a 2-colored graph $G = (V, E)$

- Subset $A \subseteq V$ such that by flipping the colors of nodes in A the potential decreases

Improving sets

Improving set in a 2-colored graph $G = (V, E)$

- Subset $A \subseteq V$ such that by flipping the colors of nodes in A the potential decreases
- $\mathrm{Imp}(A)$ = improvement in the potential

Improving sets

Improving set in a 2-colored graph $G = (V, E)$

- Subset $A \subseteq V$ such that by flipping the colors of nodes in A the potential decreases

- $\mathrm{Imp}(A)$ = improvement in the potential

- Improving ratio: $\mathrm{IR}(A) = \mathrm{Imp}(A)/|A|$

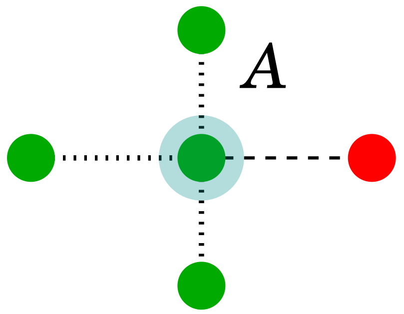
Improving sets

Improving set in a 2-colored graph $G = (V, E)$

- Subset $A \subseteq V$ such that by flipping the colors of nodes in A the potential decreases

- $\text{Imp}(A)$ = improvement in the potential

- Improving ratio: $\text{IR}(A) = \text{Imp}(A)/|A|$

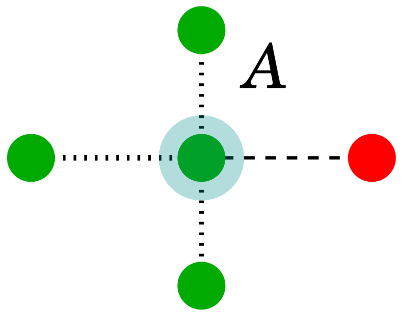


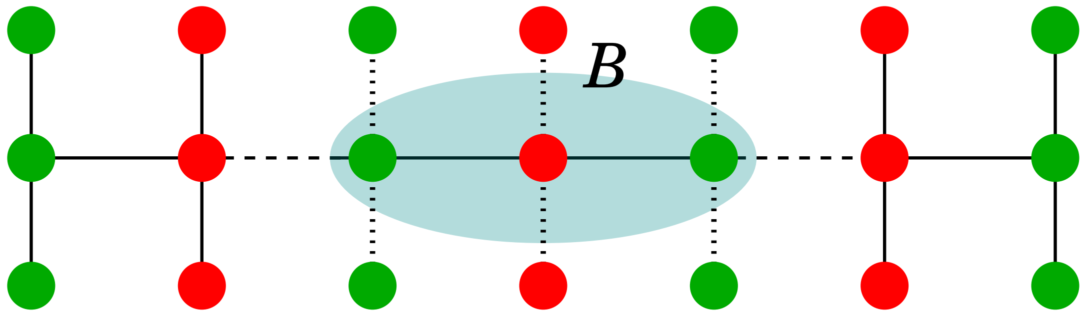
$\text{Imp}(A) = 3 - 1 = 2$

$\text{IR}(A) = 2/1 = 2$

Improving set in a 2-colored graph $G = (V, E)$

- Subset $A \subseteq V$ such that by flipping the colors of nodes in A the potential decreases
- Imp(A) = improvement in the potential
- Improving ratio: IR(A) = Imp$(A)/|A|$





Imp$(A) = 3 - 1 = 2$

IR$(A) = 2/1 = 2$

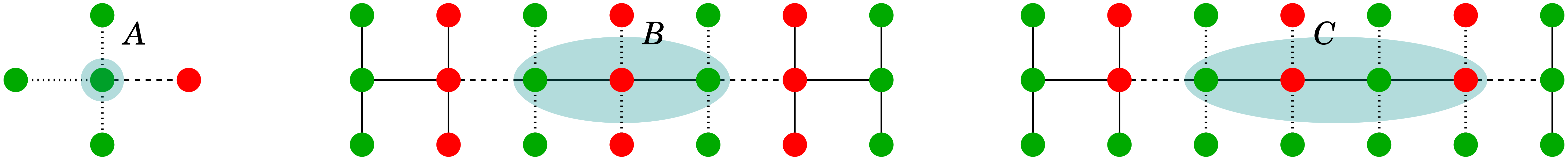
Imp$(B) = 4$

IR$(B) = 4/3$

Improving sets

Improving set in a 2-colored graph $G = (V, E)$

- Subset $A \subseteq V$ such that by flipping the colors of nodes in A the potential decreases
- $\text{Imp}(A)$ = improvement in the potential
- Improving ratio: $\text{IR}(A) = \text{Imp}(A)/|A|$



$\text{Imp}(A) = 3 - 1 = 2$

$\text{IR}(A) = 2/1 = 2$

$\text{Imp}(B) = 4$

$\text{IR}(B) = 4/3$

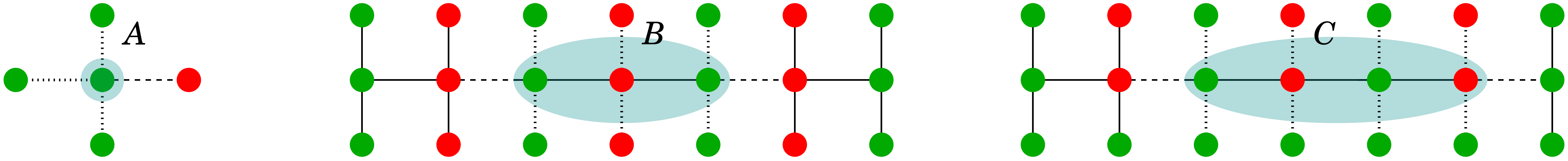
$\text{Imp}(C) = 2$

$\text{IR}(C) = 1/2$

Improving sets

Improving set in a 2-colored graph $G = (V, E)$

- Subset $A \subseteq V$ such that by flipping the colors of nodes in A the potential decreases
- $\text{Imp}(A)$ = improvement in the potential
- Improving ratio: $\text{IR}(A) = \text{Imp}(A)/|A|$



$\text{Imp}(A) = 3 - 1 = 2$ $\quad$ $\text{Imp}(B) = 4$ $\quad$ $\text{Imp}(C) = 2$

$\text{IR}(A) = 2/1 = 2$ $\quad$ $\text{IR}(B) = 4/3$ $\quad$ $\text{IR}(C) = 1/2$

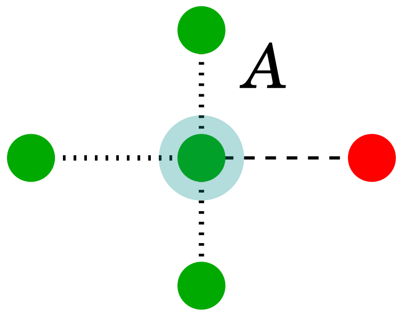
Minimal improving set: improving set A such that

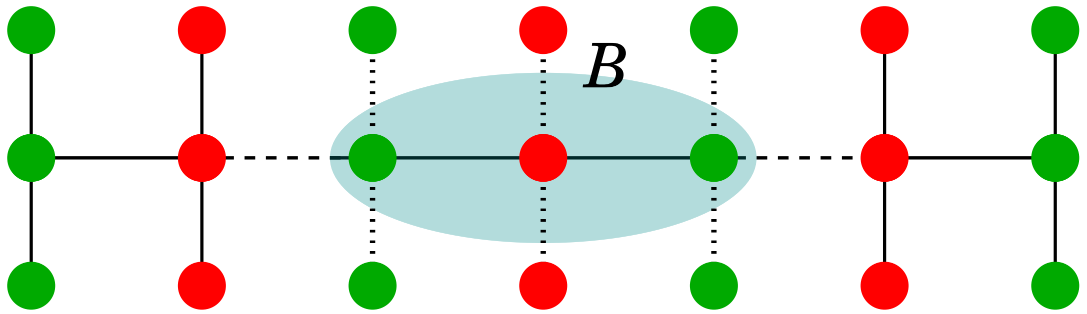
- There is no subset $A' \subseteq A$ with $\text{IR}(A') > \text{IR}(A)$ $\longrightarrow$ "quality": is there "useless stuff"?

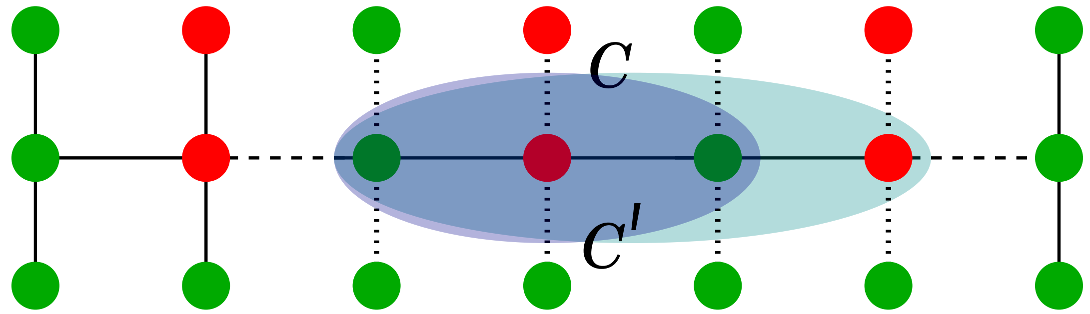
Improving set in a 2-colored graph $G = (V, E)$

- Subset $A \subseteq V$ such that by flipping the colors of nodes in A the potential decreases
- $\text{Imp}(A)$ = improvement in the potential
- Improving ratio: $\text{IR}(A) = \text{Imp}(A)/|A|$

C not minimal







$\text{Imp}(A) = 3 - 1 = 2$

$\text{IR}(A) = 2/1 = 2$

$\text{Imp}(B) = 4$

$\text{IR}(B) = 4/3$

$\text{Imp}(C) = 2$

$\text{IR}(C) = 1/2$

$\text{Imp}(C') = 2$

$\text{IR}(C') = 2/3$

Minimal improving set: improving set A such that

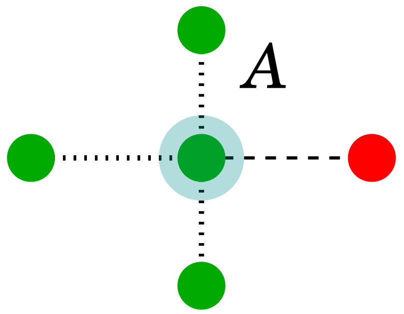
- There is no subset $A' \subseteq A$ with $\text{IR}(A') > \text{IR}(A)$ $\longrightarrow$ "quality": is there "useless stuff"?

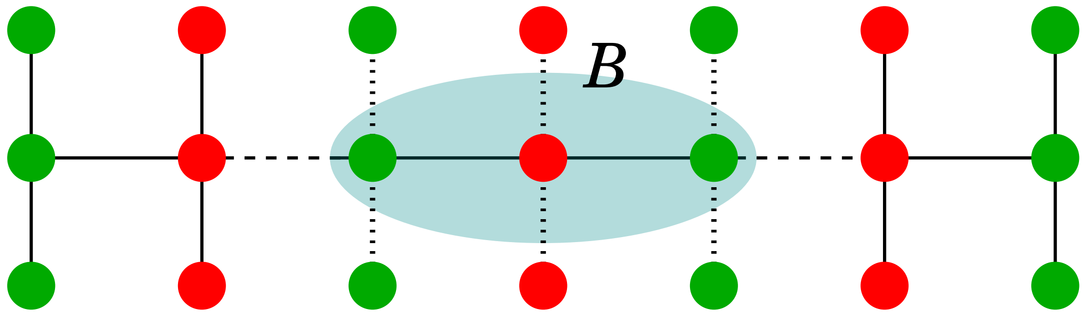
Improving sets

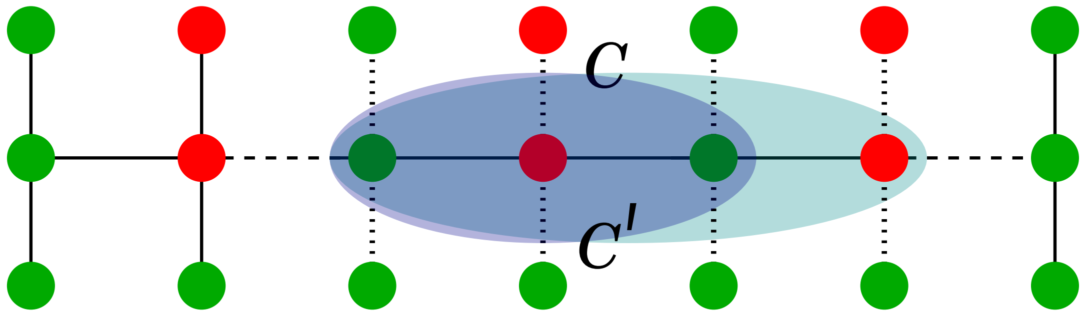
Improving set in a 2-colored graph $G = (V, E)$

- Subset $A \subseteq V$ such that by flipping the colors of nodes in A the potential decreases
- $\text{Imp}(A)$ = improvement in the potential
- Improving ratio: $\text{IR}(A) = \text{Imp}(A)/|A|$

C not minimal







$\text{Imp}(A) = 3 - 1 = 2$

$\text{IR}(A) = 2/1 = 2$

$\text{Imp}(B) = 4$

$\text{IR}(B) = 4/3$

$\text{Imp}(C) = 2$

$\text{IR}(C) = 1/2$

$\text{Imp}(C') = 2$

$\text{IR}(C') = 2/3$

An **error** is always a minimal improving set

Minimal improving set: improving set A such that

- There is no subset $A' \subseteq A$ with $\text{IR}(A') > \text{IR}(A)$ $\longrightarrow$ "quality": is there "useless stuff"?

Improving sets

Property 1: on minimal improving sets

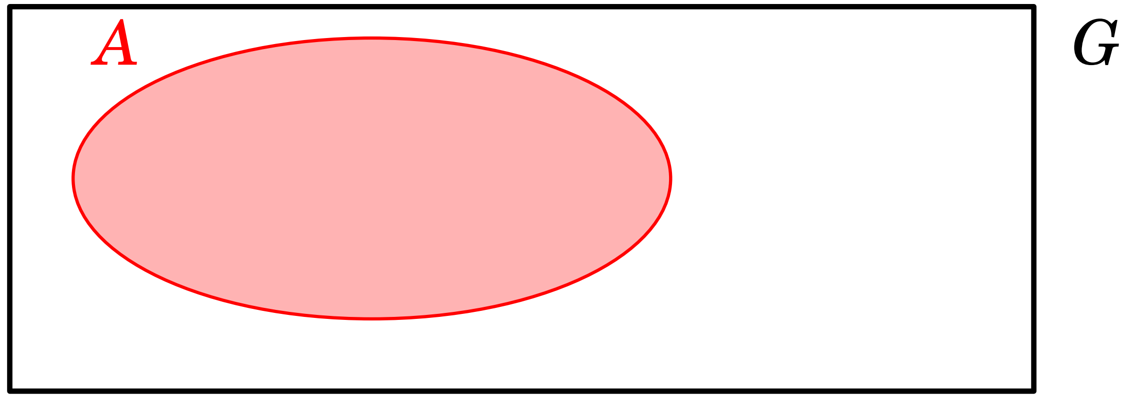
G

- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n / \varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Improving sets

Property 1: on minimal improving sets

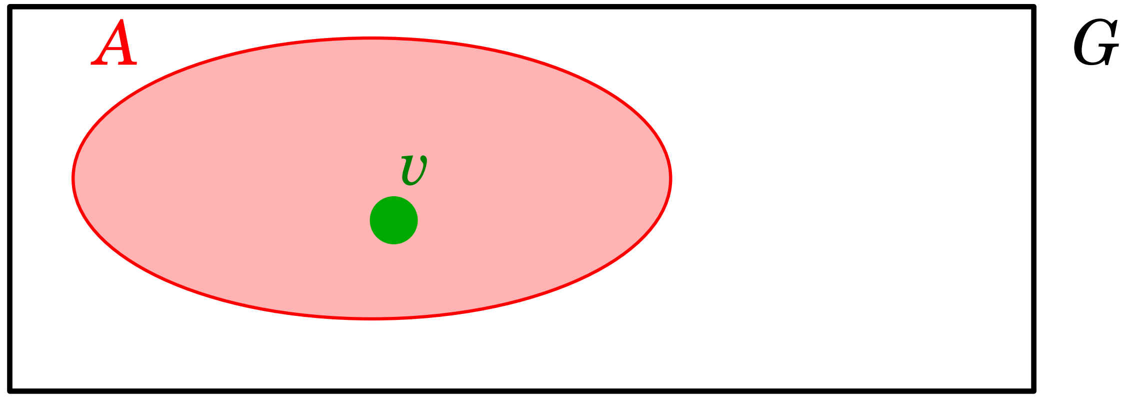


- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Improving sets

Property 1: on minimal improving sets

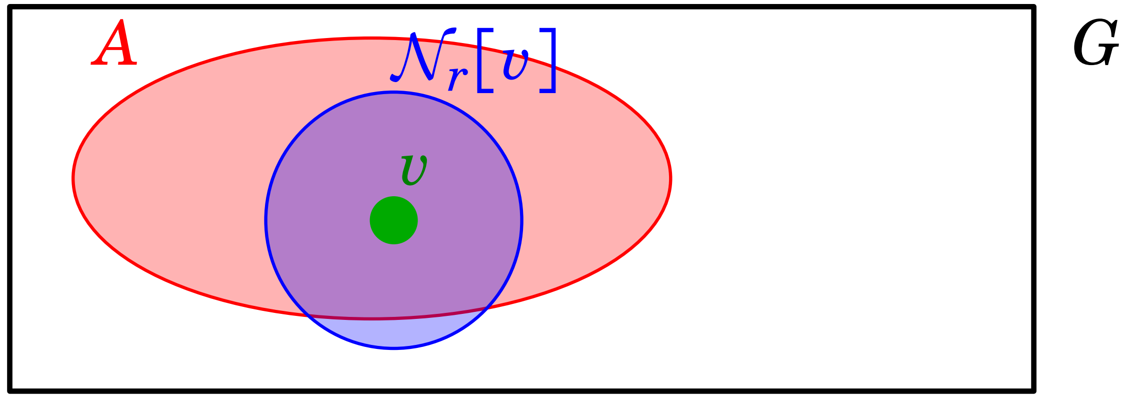


- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n / \varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Property 1: on minimal improving sets

- $A \subseteq V$ minimal improving set
- $\text{IR}(A) \geq x$
- $\varepsilon < x$

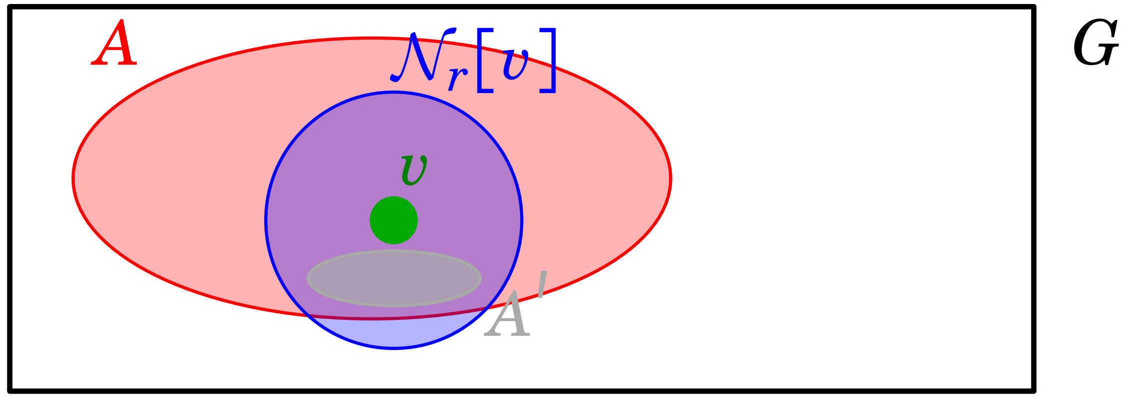


$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\text{IR}(A') \geq x - \varepsilon$

Improving sets

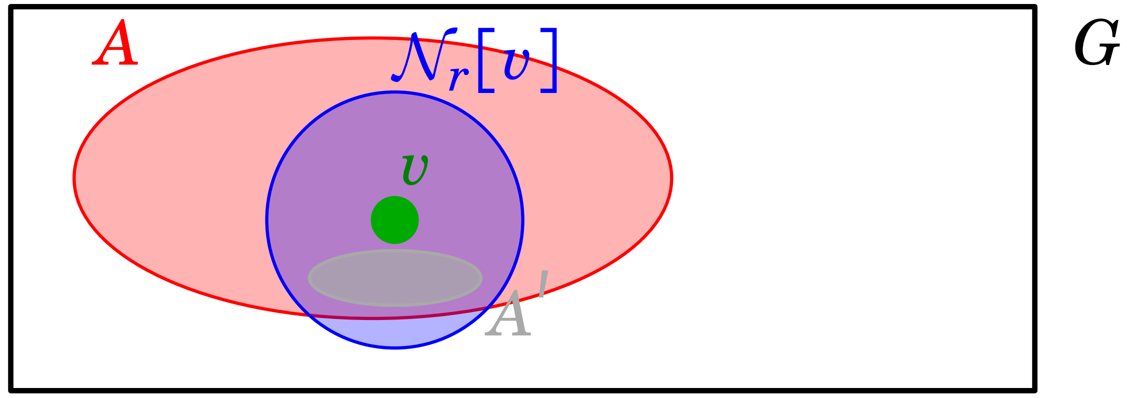
Property 1: on minimal improving sets

- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$



$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n / \varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Improving sets



Property 1: on minimal improving sets

- $A \subseteq V$ minimal improving set
- $\text{IR}(A) \geq x$
- $\varepsilon < x$

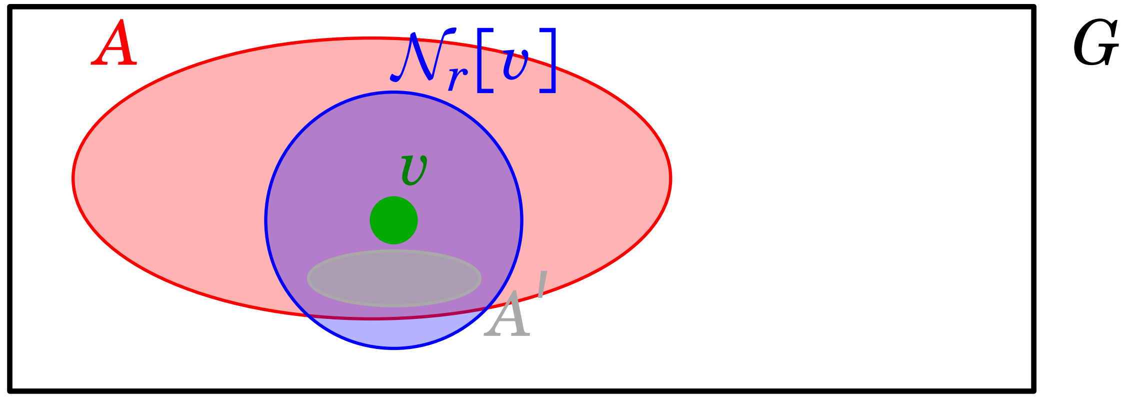
$\implies$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\text{IR}(A') \geq x - \varepsilon$

Sequence of x-improving sets:

- $A_1, \ldots, A_k \subseteq V$
- A_1 minimal improving set with $\text{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\text{IR}(A_i) \geq x$ after having flipped $A_1, \ldots, A_{i-1}$

Property 1: on minimal improving sets

- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$



$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

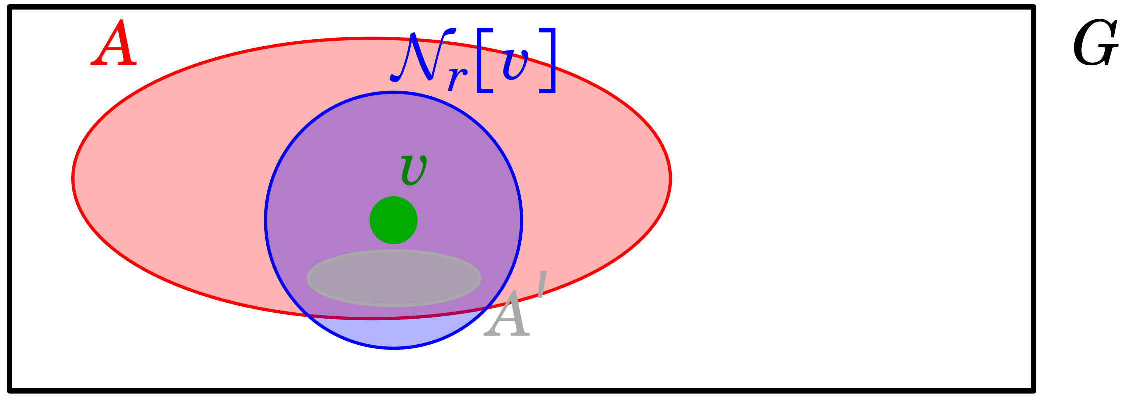
Sequence of x-improving sets:

- $A_1, \ldots, A_k \subseteq V$
- A_1 minimal improving set with $\mathrm{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\mathrm{IR}(A_i) \geq x$ after having flipped $A_1, \ldots, A_{i-1}$

Improving sets

Property 1: on minimal improving sets

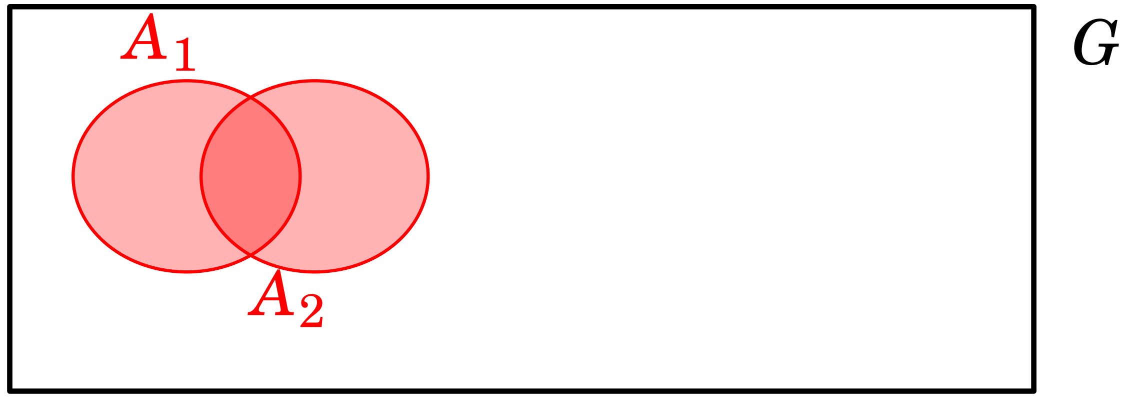
- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$



$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

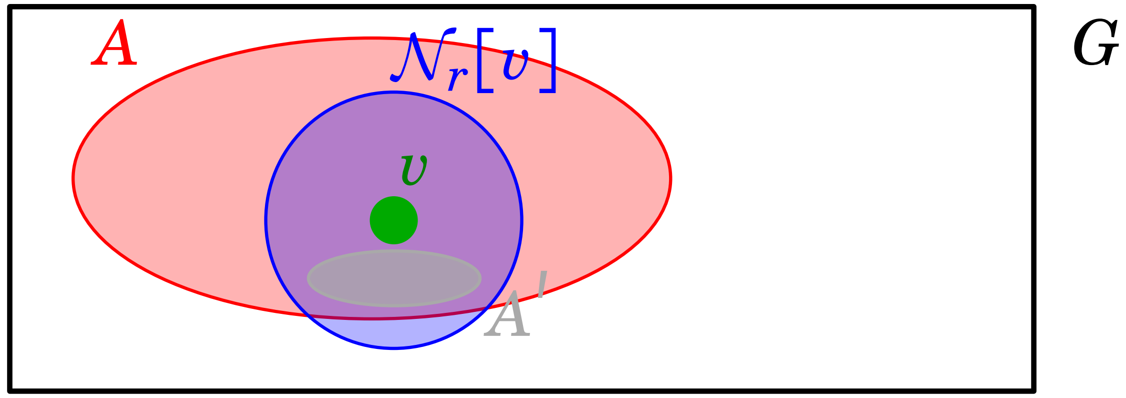
Sequence of x-improving sets:

- $A_1, \ldots, A_k \subseteq V$
- A_1 minimal improving set with $\mathrm{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\mathrm{IR}(A_i) \geq x$ after having flipped $A_1, \ldots, A_{i-1}$



Improving sets

Property 1: on minimal improving sets

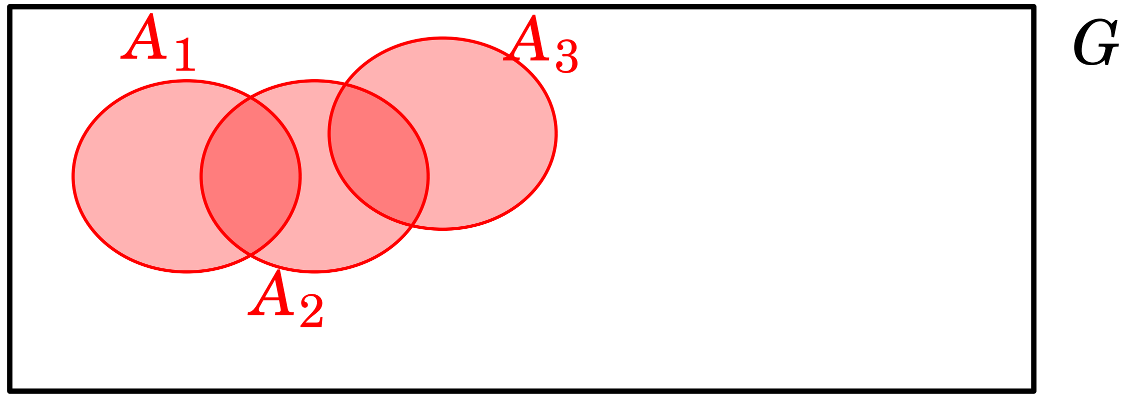


- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Sequence of x-improving sets:

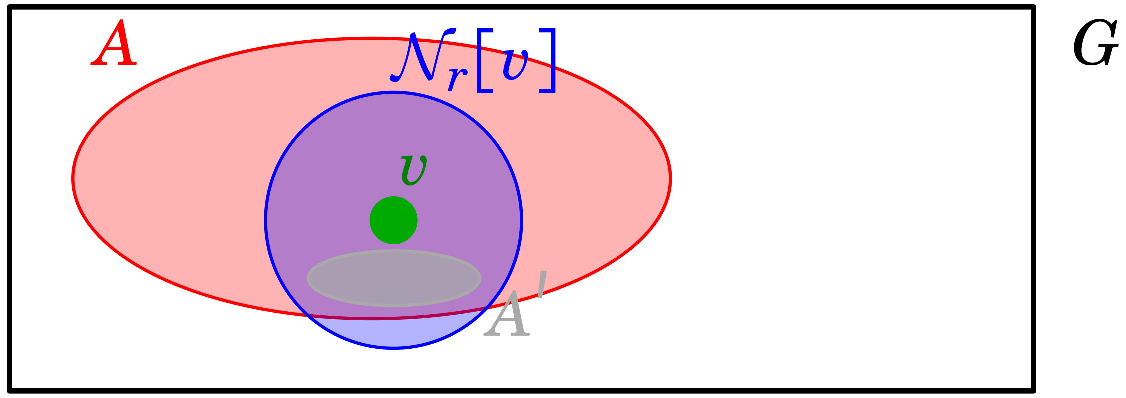
- $A_1, \ldots, A_k \subseteq V$
- A_1 minimal improving set with $\mathrm{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\mathrm{IR}(A_i) \geq x$ after having flipped $A_1, \ldots, A_{i-1}$



Improving sets

Property 1: on minimal improving sets

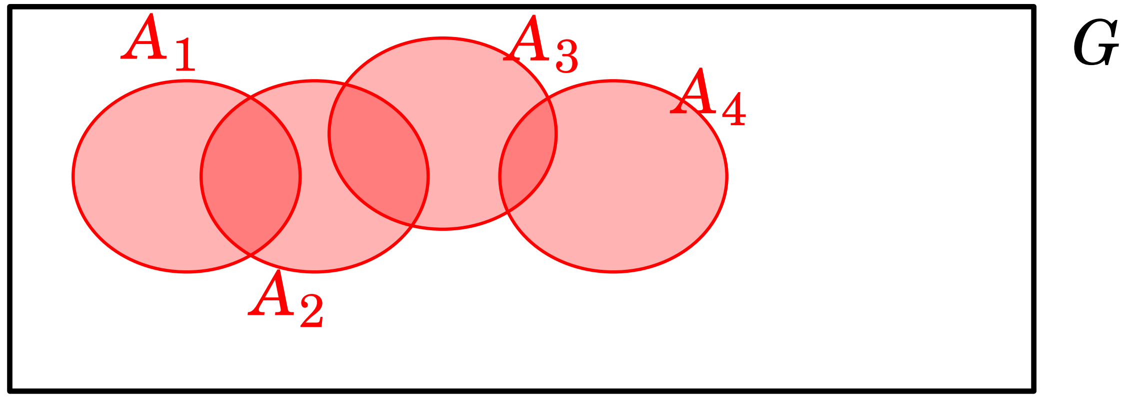
- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$



$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

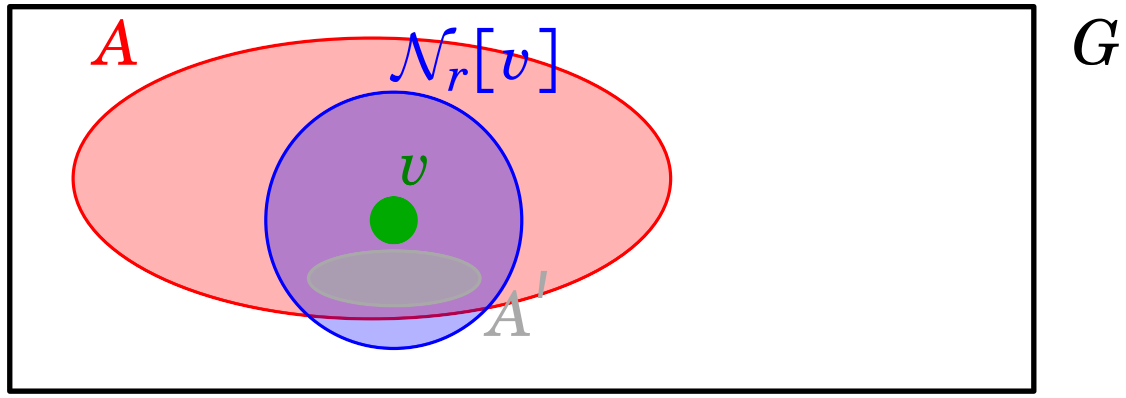
Sequence of x-improving sets:

- $A_1, \ldots, A_k \subseteq V$
- A_1 minimal improving set with $\mathrm{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\mathrm{IR}(A_i) \geq x$ after having flipped $A_1, \ldots, A_{i-1}$



Improving sets

Property 1: on minimal improving sets

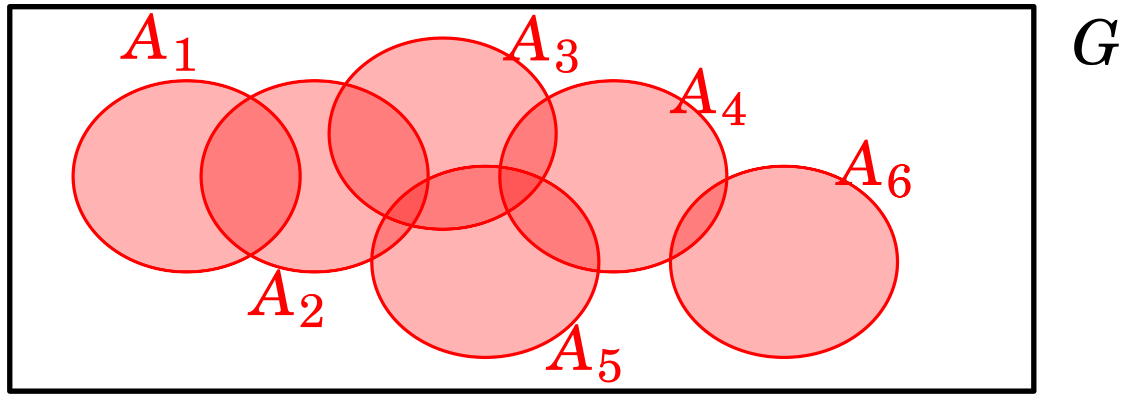


- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

$\implies$ for all $v \in A$, $\exists r = O(\log n / \varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Sequence of x-improving sets:

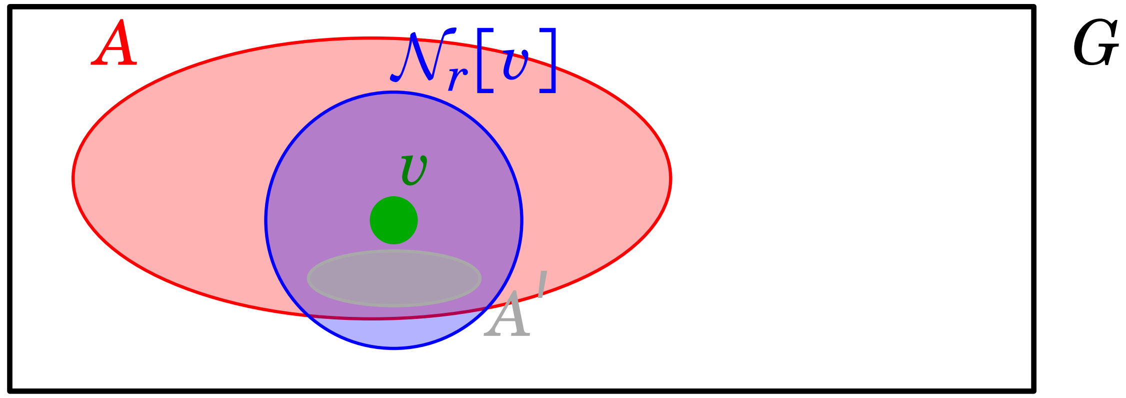
- $A_1, \ldots, A_k \subseteq V$
- A_1 minimal improving set with $\mathrm{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\mathrm{IR}(A_i) \geq x$ after having flipped $A_1, \ldots, A_{i-1}$



Improving sets

Property 1: on minimal improving sets

- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$



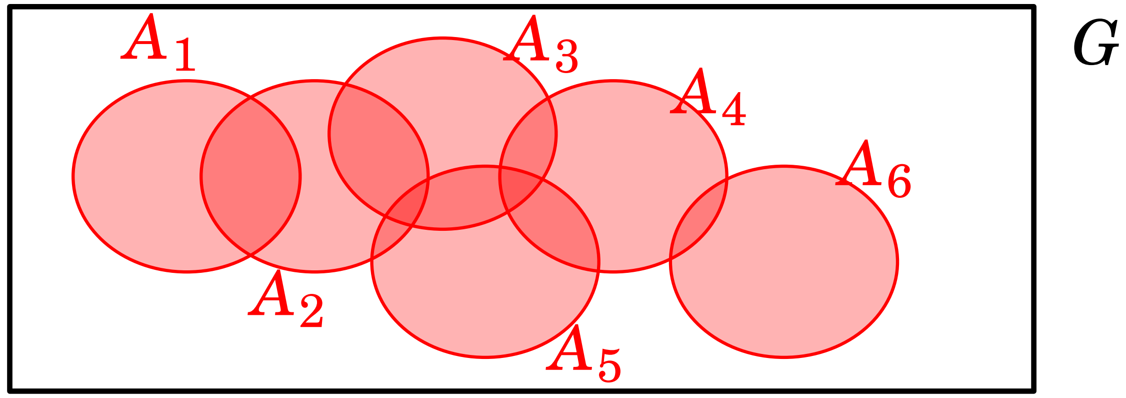
$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Sequence of x-improving sets:

- $A_1, \ldots, A_k \subseteq V$
- A_1 minimal improving set with $\mathrm{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\mathrm{IR}(A_i) \geq x$ after having flipped $A_1, \ldots, A_{i-1}$

Property 2: on sequences of x-improving sets

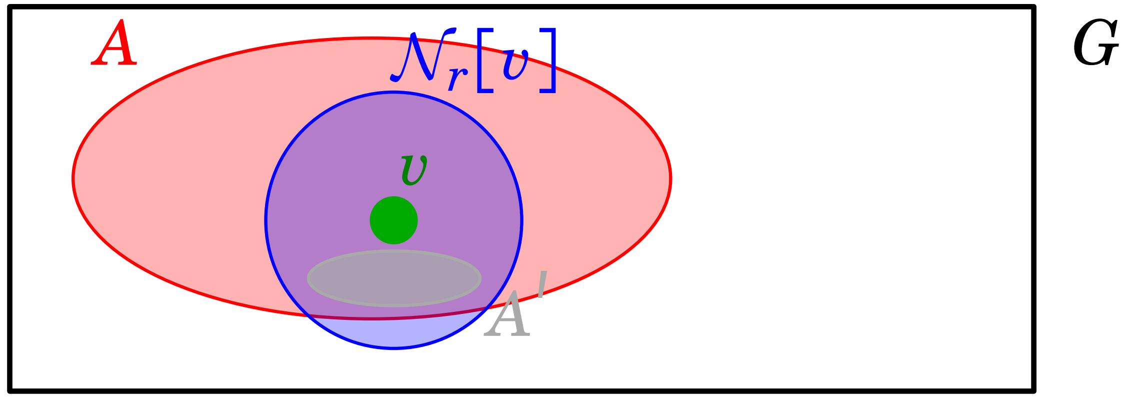
- $A_1, \ldots, A_k \subseteq V$ sequence of x-improving sets
- $\mathrm{diam}(A_i) = O(\log n/\varepsilon)$
- $\varepsilon < x$
- $A = \cup_i A_i$



$\Longrightarrow$ for all i, for all $v \in A$, $\exists r = O(\log^2 n/\varepsilon^2)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Improving sets

Property 1: on minimal improving sets



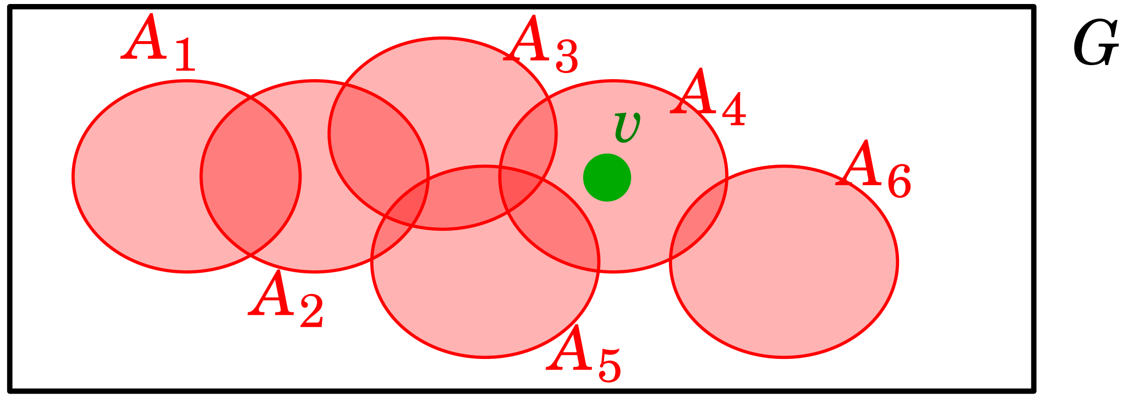
- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Sequence of x-improving sets:

- $A_1, \ldots, A_k \subseteq V$
- A_1 minimal improving set with $\mathrm{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\mathrm{IR}(A_i) \geq x$ after having flipped $A_1, \ldots, A_{i-1}$

Property 2: on sequences of x-improving sets

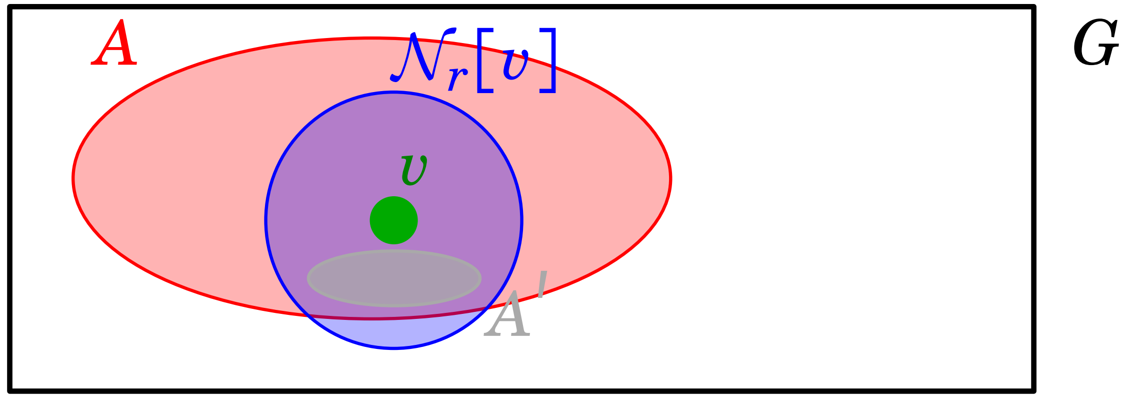


- $A_1, \ldots, A_k \subseteq V$ sequence of x-improving sets
- $\mathrm{diam}(A_i) = O(\log n/\varepsilon)$
- $\varepsilon < x$
- $A = \cup_i A_i$

$\Longrightarrow$ for all i, for all $v \in A$, $\exists r = O(\log^2 n/\varepsilon^2)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Improving sets

Property 1: on minimal improving sets



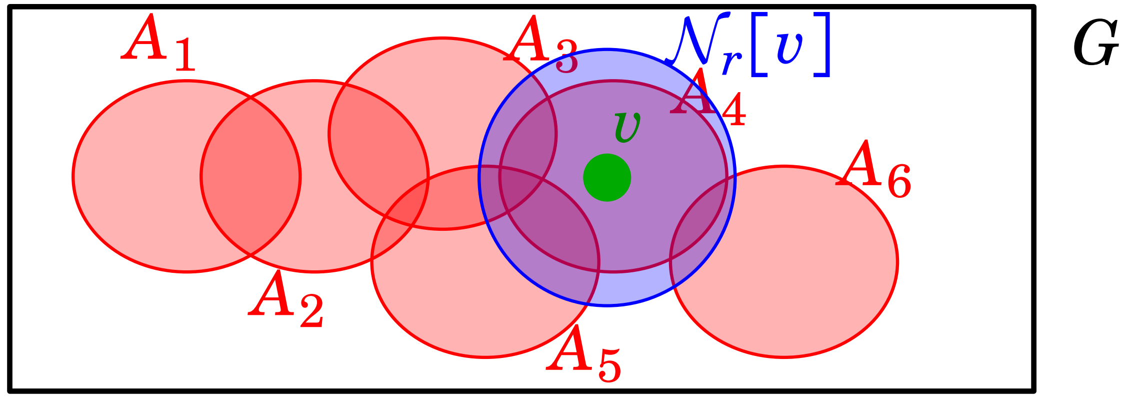
- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Sequence of x-improving sets:

- $A_1, \dots, A_k \subseteq V$
- A_1 minimal improving set with $\mathrm{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\mathrm{IR}(A_i) \geq x$ after having flipped $A_1, \dots, A_{i-1}$

Property 2: on sequences of x-improving sets

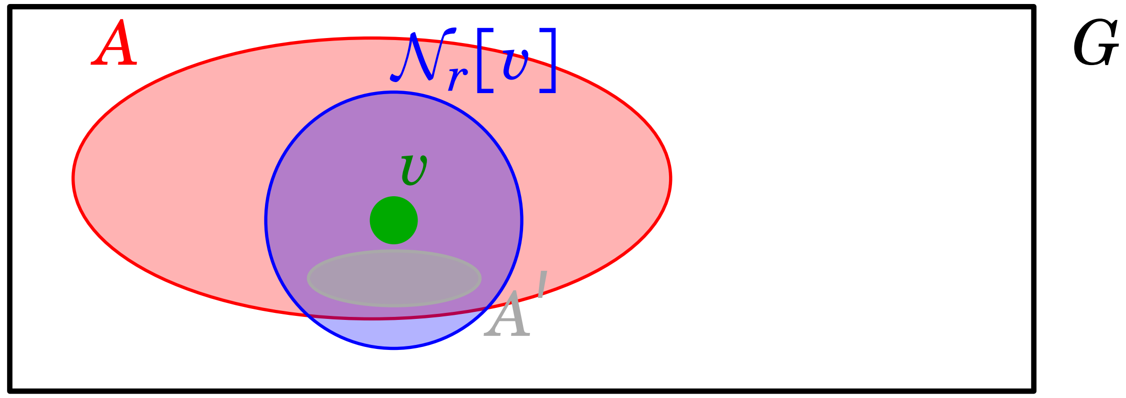


- $A_1, \dots, A_k \subseteq V$ sequence of x-improving sets
- $\mathrm{diam}(A_i) = O(\log n/\varepsilon)$
- $\varepsilon < x$
- $A = \cup_i A_i$

$\Longrightarrow$ for all i, for all $v \in A$, $\exists r = O(\log^2 n/\varepsilon^2)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Improving sets

Property 1: on minimal improving sets



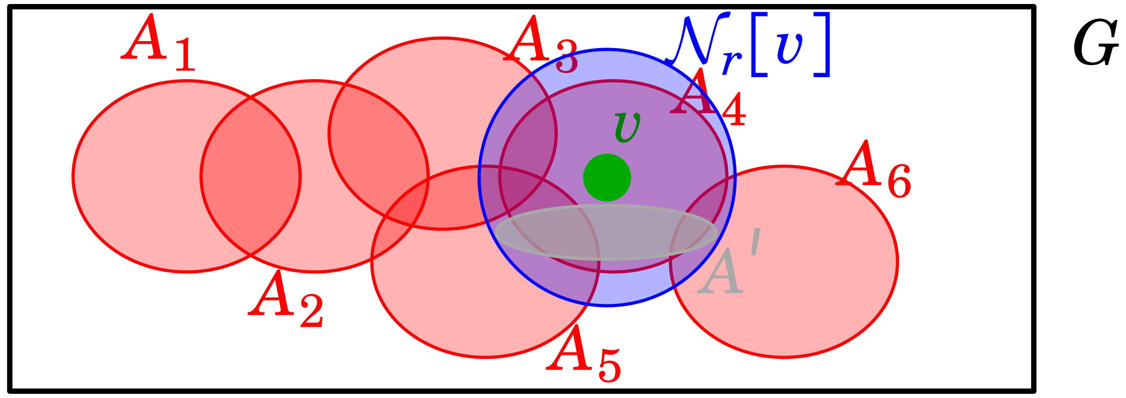
- $A \subseteq V$ minimal improving set
- $\text{IR}(A) \geq x$
- $\varepsilon < x$

$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n / \varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\text{IR}(A') \geq x - \varepsilon$

Sequence of x-improving sets:

- $A_1, \ldots, A_k \subseteq V$
- A_1 minimal improving set with $\text{IR}(A_1) \geq x$
- for all $i > 1$, A_i minimal improving set with $\text{IR}(A_i) \geq x$ after having flipped $A_1, \ldots, A_{i-1}$

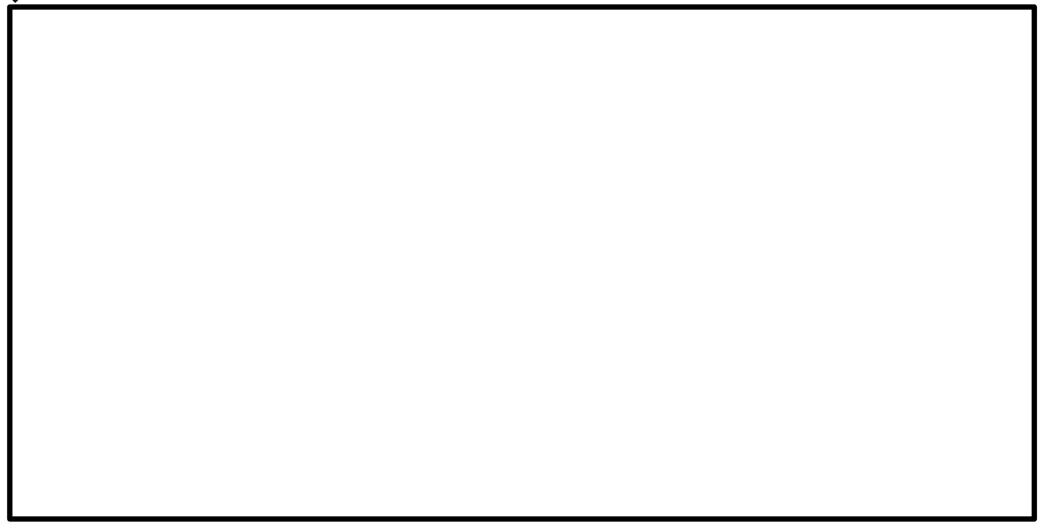
Property 2: on sequences of x-improving sets



- $A_1, \ldots, A_k \subseteq V$ sequence of x-improving sets
- $\text{diam}(A_i) = O(\log n / \varepsilon)$
- $\varepsilon < x$
- $A = \cup_i A_i$

$\Longrightarrow$ for all i, for all $v \in A$, $\exists r = O(\log^2 n / \varepsilon^2)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\text{IR}(A') \geq x - \varepsilon$

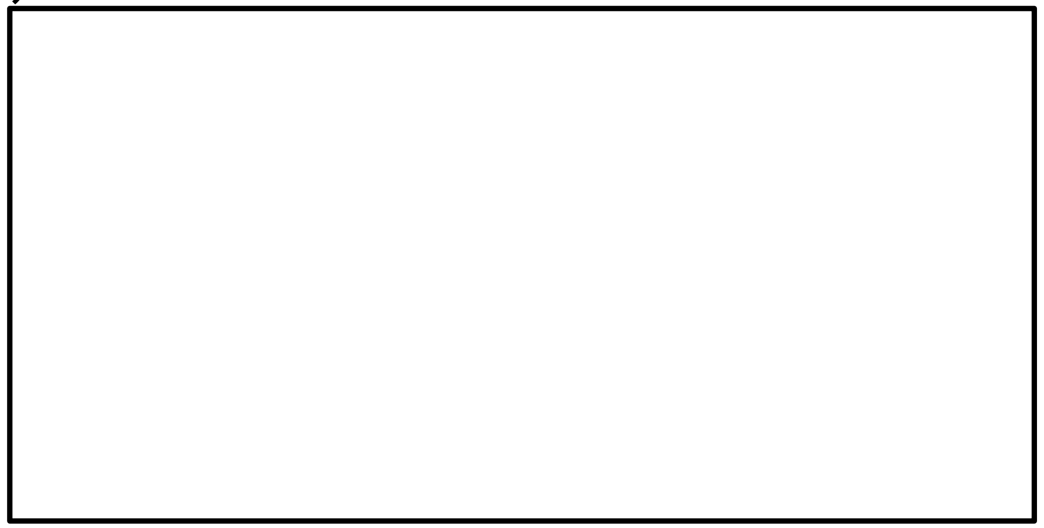
- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties* 1,2), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment



G

The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do

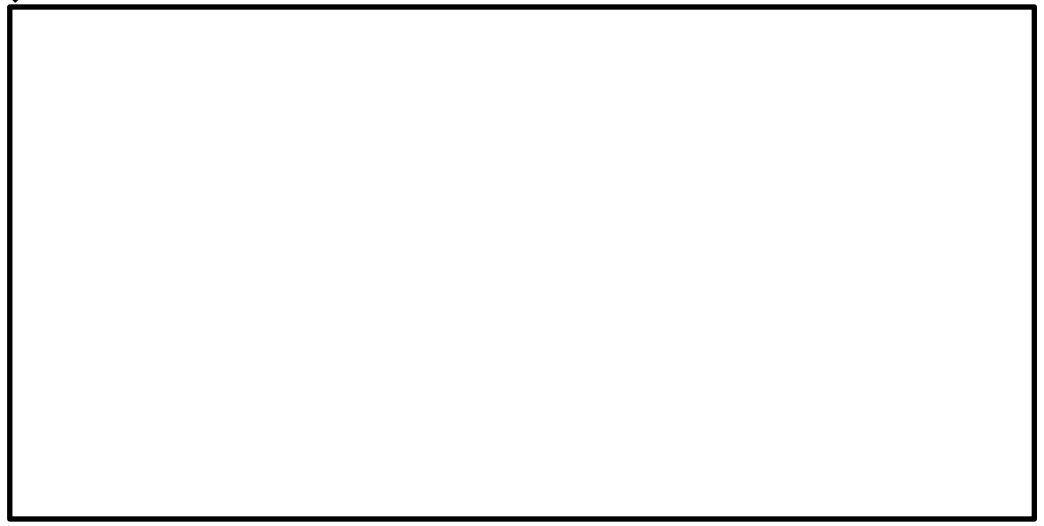


G

The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do

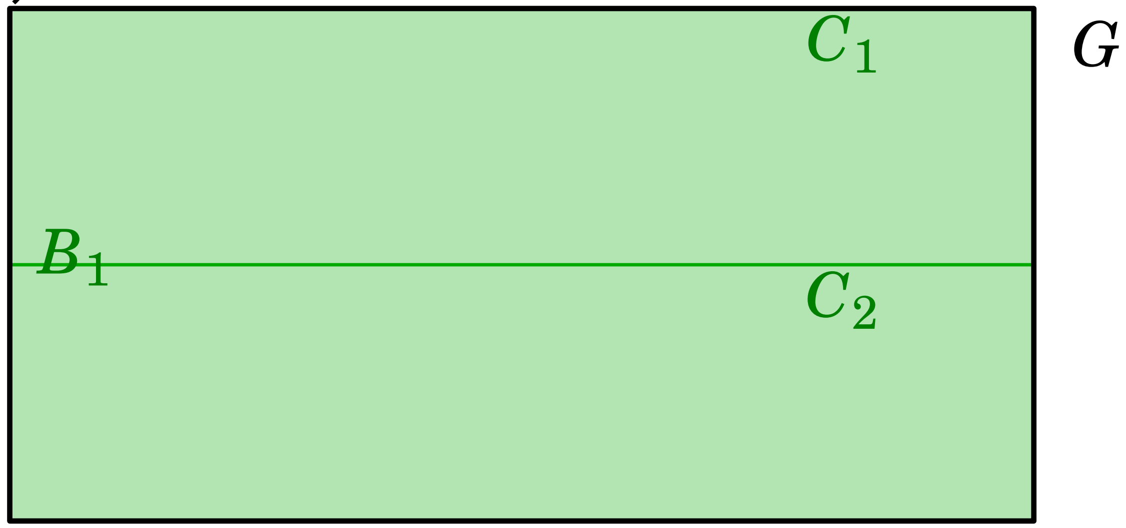
  - Run $\mathcal{MPX}$ to get an (α, d)-decomposition



G

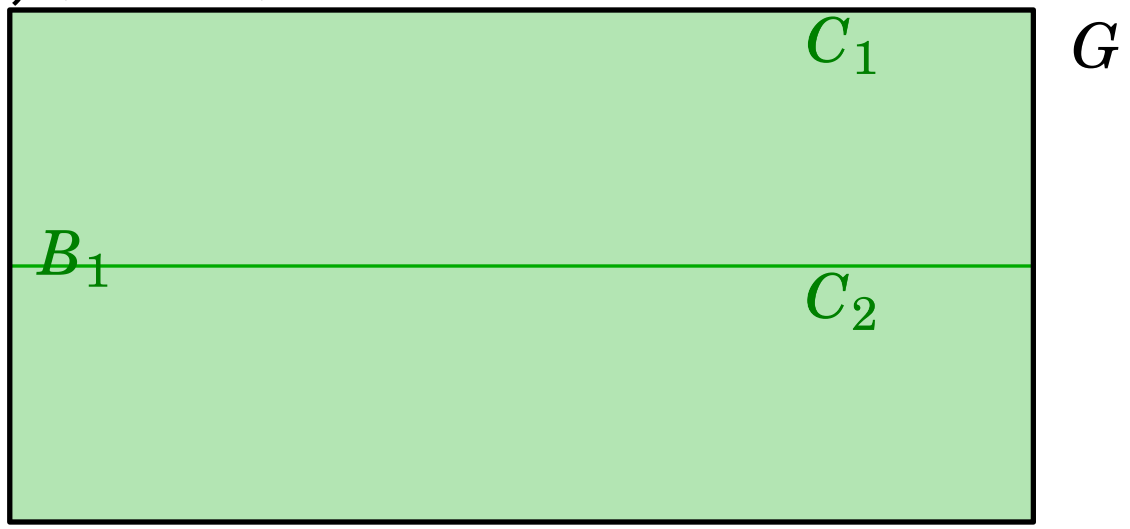
The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do
  - Run $\mathcal{MPX}$ to get an (α, d)-decomposition



The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)



- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do
  - Run $\mathcal{MPX}$ to get an (α, d)-decomposition
  - $\mathcal{C}$ = current cluster
  - Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes

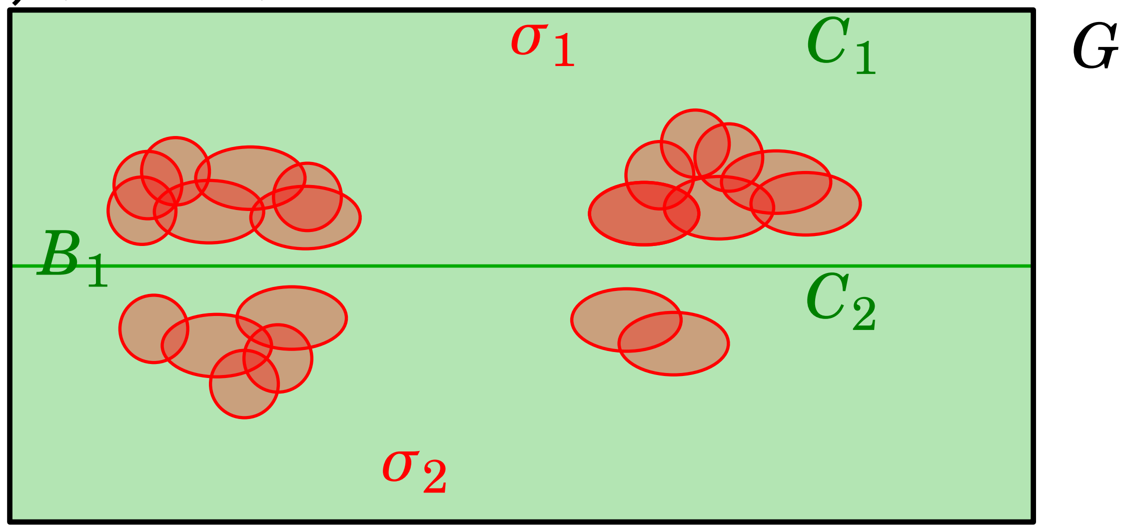
The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do

  - Run $\mathcal{MPX}$ to get an (α, d)-decomposition

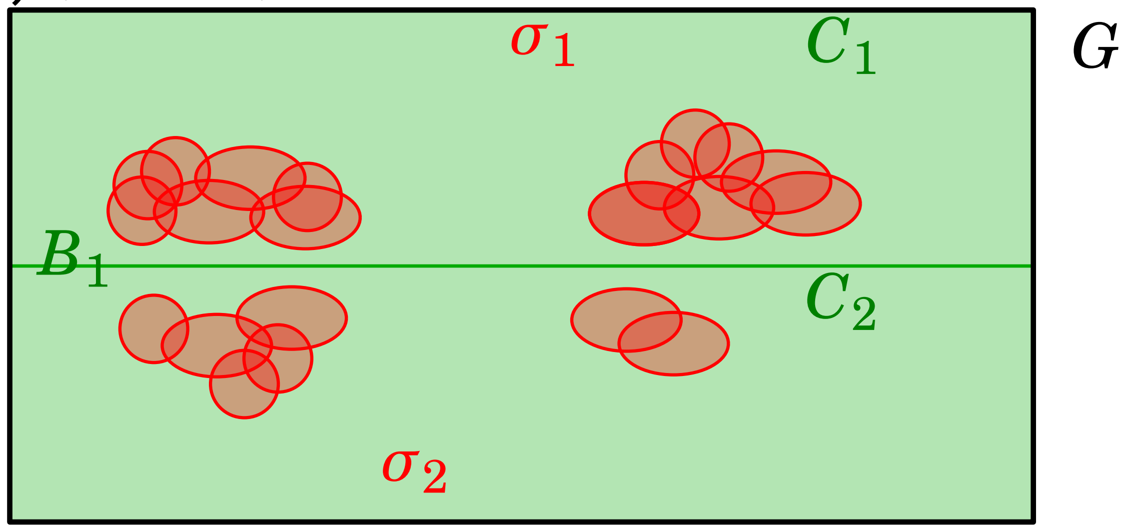
  - $\mathcal{C}$ = current cluster

  - Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes



The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do
  - Run $\mathcal{MPX}$ to get an (α, d)-decomposition
  - $\mathcal{C}$ = current cluster
  - Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
  - Flip all sets in σ, in order



The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000\log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100\log n$ times, do

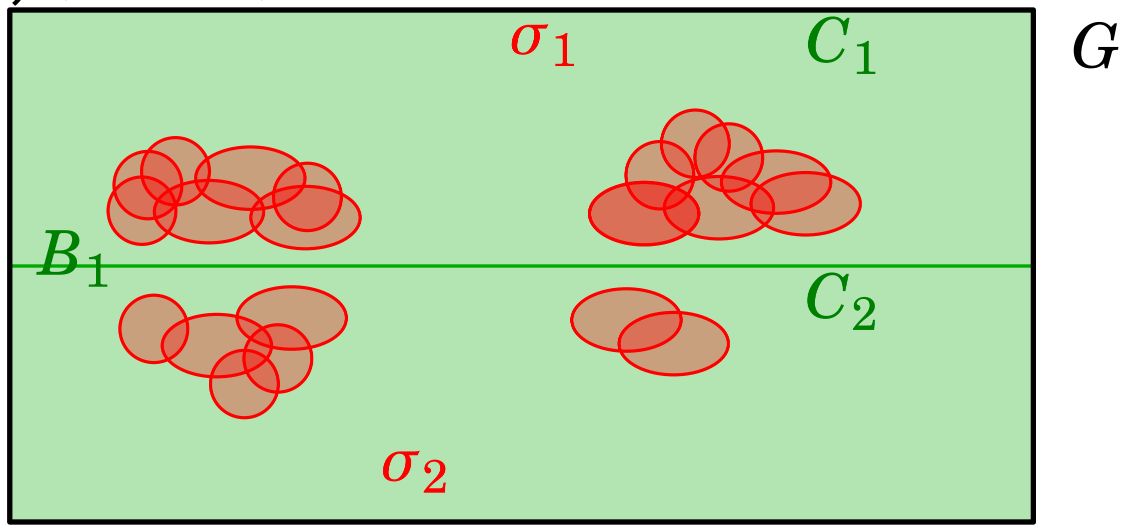
  - Run $\mathcal{MPX}$ to get an (α, d)-decomposition

  - $\mathcal{C}$ = current cluster

  - Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes

  - Flip all sets in σ, in order

  - $\lambda_{i+1} = \lambda_i + 10\varepsilon$

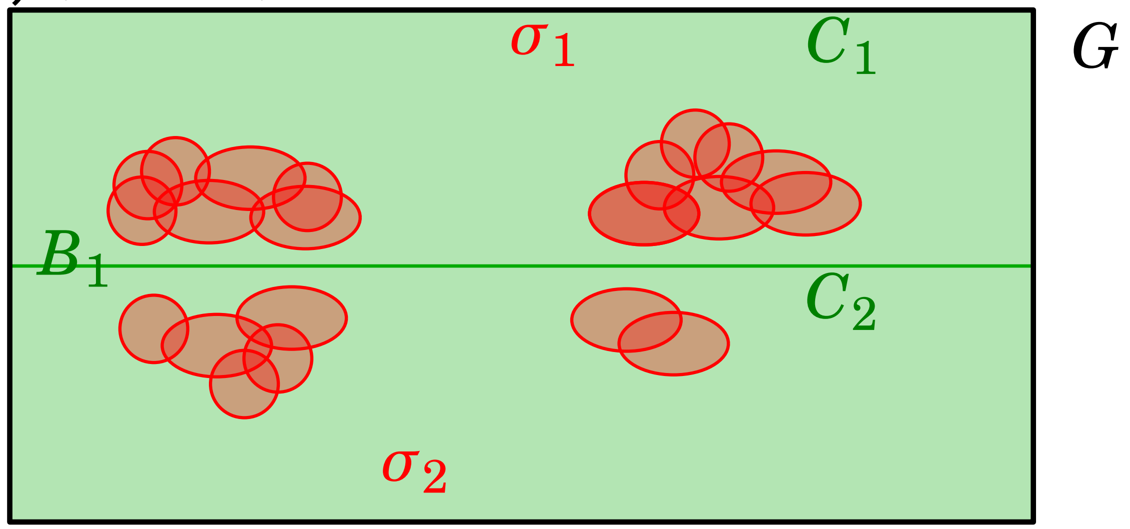


The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000\log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100\log n$ times, do

PHASE

- Run $\mathcal{MPX}$ to get an (α, d)-decomposition
- $\mathcal{C}$ = current cluster
- Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
- Flip all sets in σ, in order
- $\lambda_{i+1} = \lambda_i + 10\varepsilon$

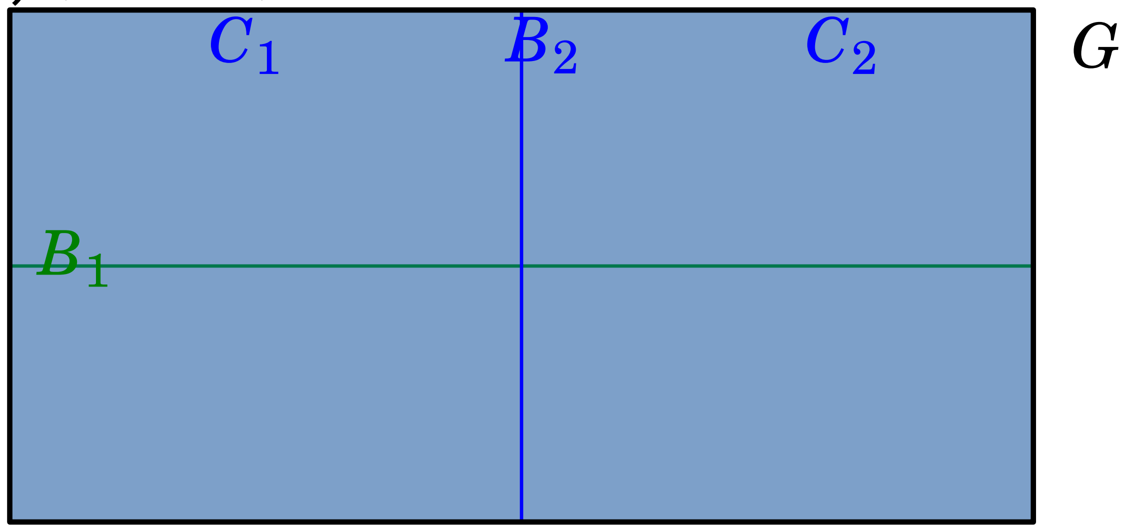


The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties* 1,2), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do

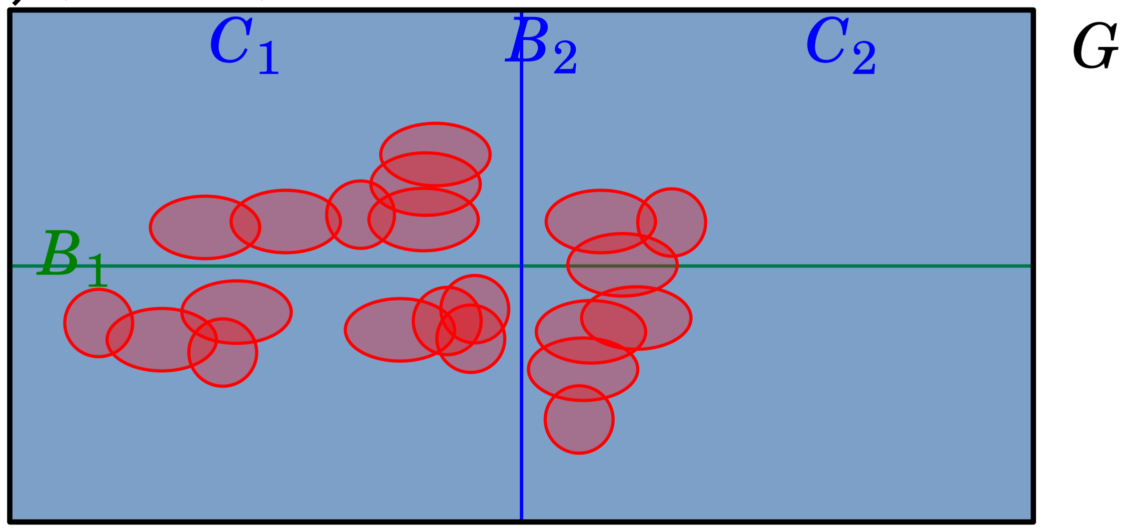
PHASE

- Run $\mathcal{MPX}$ to get an (α, d)-decomposition
- $\mathcal{C}$ = current cluster
- Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
- Flip all sets in σ, in order
- $\lambda_{i+1} = \lambda_i + 10\varepsilon$



The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do



PHASE

- Run $\mathcal{MPX}$ to get an (α, d)-decomposition
- $\mathcal{C}$ = current cluster
- Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
- Flip all sets in σ, in order
- $\lambda_{i+1} = \lambda_i + 10\varepsilon$

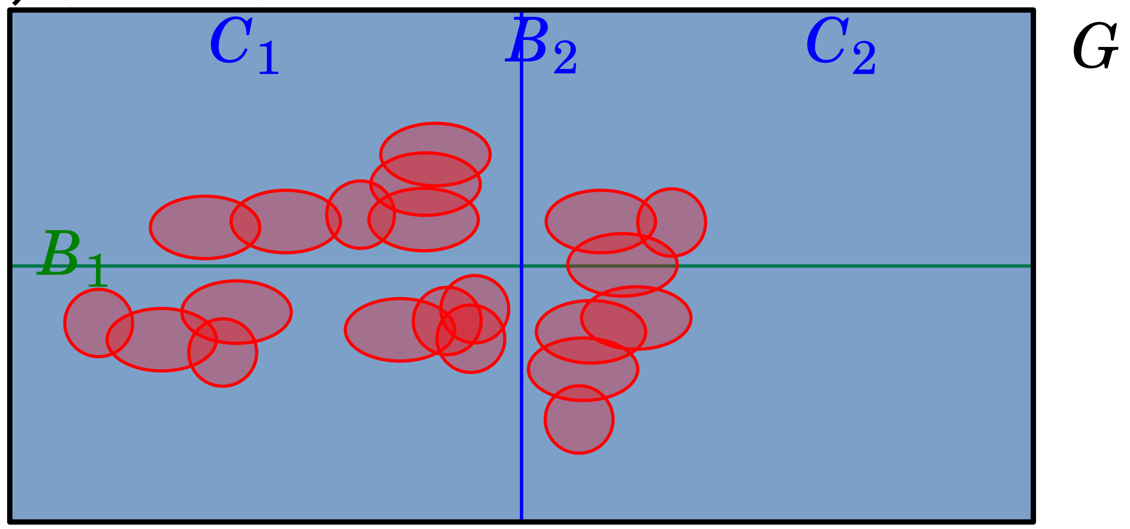
The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do

PHASE

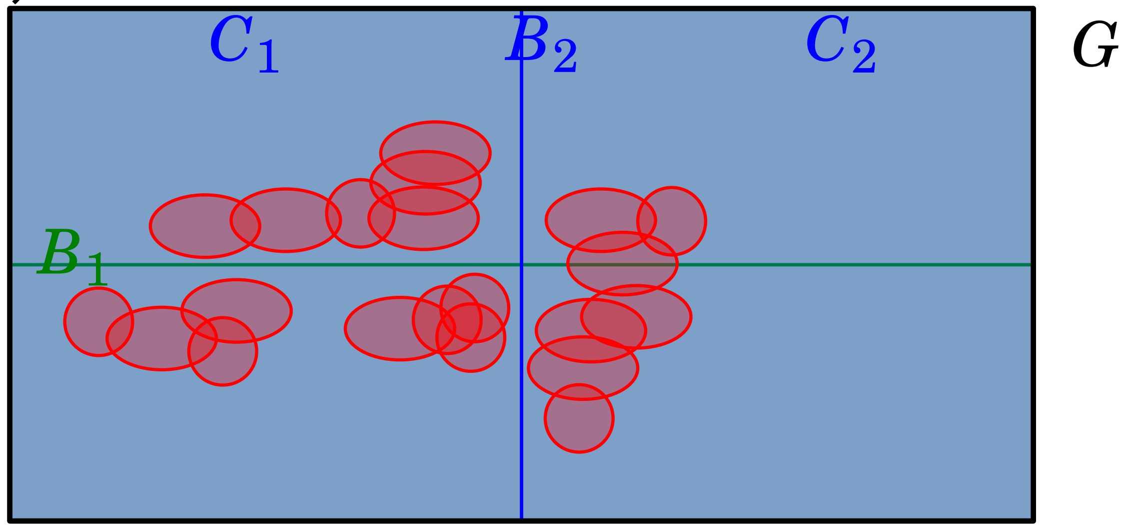
  - Run $\mathcal{MPX}$ to get an (α, d)-decomposition
  - $\mathcal{C}$ = current cluster
  - Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
  - Flip all sets in σ, in order
  - $\lambda_{i+1} = \lambda_i + 10\varepsilon$

- Return current coloring



The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000\log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100\log n$ times, do



PHASE

- Run $\mathcal{MPX}$ to get an (α, d)-decomposition
- $\mathcal{C}$ = current cluster
- Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
- Flip all sets in σ, in order
- $\lambda_{i+1} = \lambda_i + 10\varepsilon$

- Return current coloring

Claim 1: *The running time of the algorithm is* $O(d \cdot \log n) = O(\log^2 n/\alpha) = O(\log^6 n)$

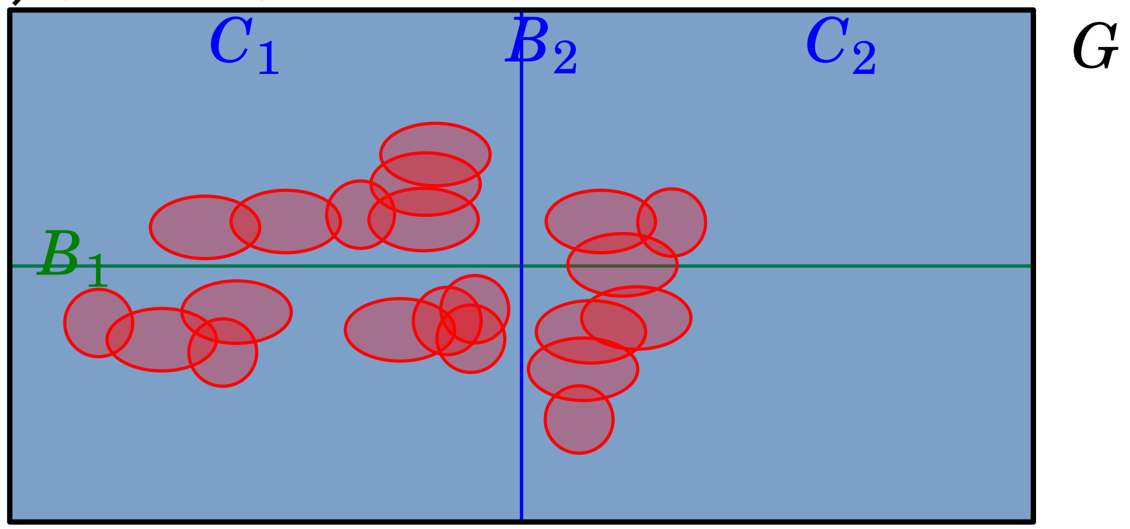
The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do

PHASE

- Run $\mathcal{MPX}$ to get an (α, d)-decomposition
- $\mathcal{C}$ = current cluster
- Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
- Flip all sets in σ, in order
- $\lambda_{i+1} = \lambda_i + 10\varepsilon$

- Return current coloring



Claim 1: *The running time of the algorithm is* $O(d \cdot \log n) = O(\log^2 n/\alpha) = O(\log^6 n)$

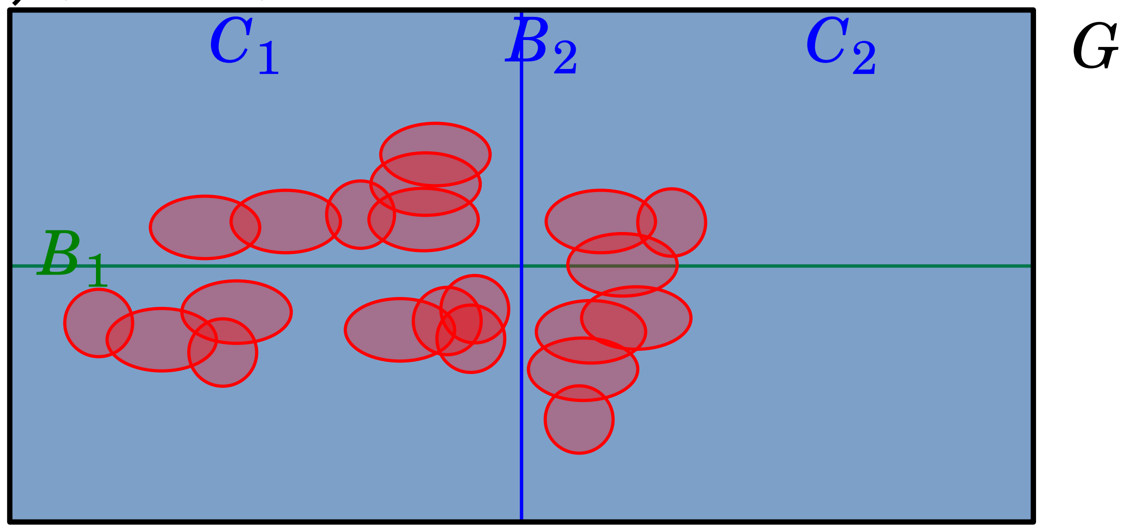
Proof 1: $O(\log n)$ phases. Each phase costs $O(d)$. By $\mathcal{MPX}$, $d = O(\log n/\alpha)$. By def. $\alpha = O(1/\log^4 n)$.

The algorithm

- Set $\lambda_1 = 1/4$ *(initial IR)*, $\varepsilon = \lambda/(2000 \log n)$ *(Properties 1,2)*, $\alpha = \Theta(\varepsilon^2/\log^2 n)$ $(\mathcal{MPX})$
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do

PHASE

- Run $\mathcal{MPX}$ to get an (α, d)-decomposition
- $\mathcal{C}$ = current cluster
- Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
- Flip all sets in σ, in order
- $\lambda_{i+1} = \lambda_i + 10\varepsilon$



- Return current coloring

Claim 1: *The running time of the algorithm is* $O(d \cdot \log n) = O(\log^2 n/\alpha) = O(\log^6 n)$

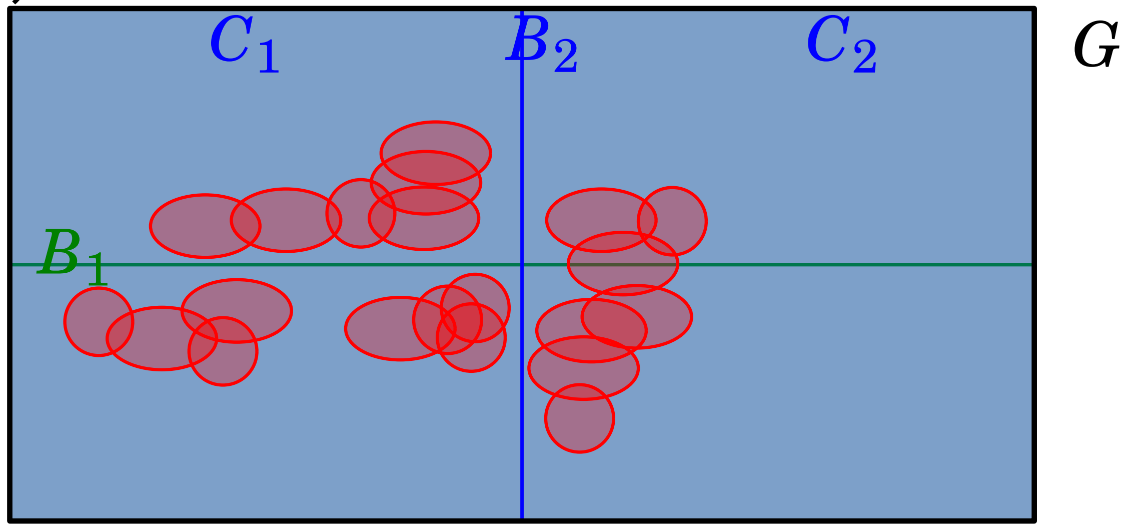
Claim 2: *After phase* i*, any min. imp. set with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies "very close" to all previous border nodes*

The algorithm

- Set $\lambda_1 = 1/4$ *(initial IR)*, $\varepsilon = \lambda/(2000 \log n)$ *(Properties 1,2)*, $\alpha = \Theta(\varepsilon^2/\log^2 n)$ $(\mathcal{MPX})$
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do

PHASE

- Run $\mathcal{MPX}$ to get an (α, d)-decomposition
- $\mathcal{C}$ = current cluster
- Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
- Flip all sets in σ, in order
- $\lambda_{i+1} = \lambda_i + 10\varepsilon$



- Return current coloring

Property 1 + Property 2

Claim 1: *The running time of the algorithm is* $O(d \cdot \log n) = O(\log^2 n/\alpha) = O(\log^6 n)$

Claim 2: *After phase* i*, any min. imp. set with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies "very close" to all previous border nodes*

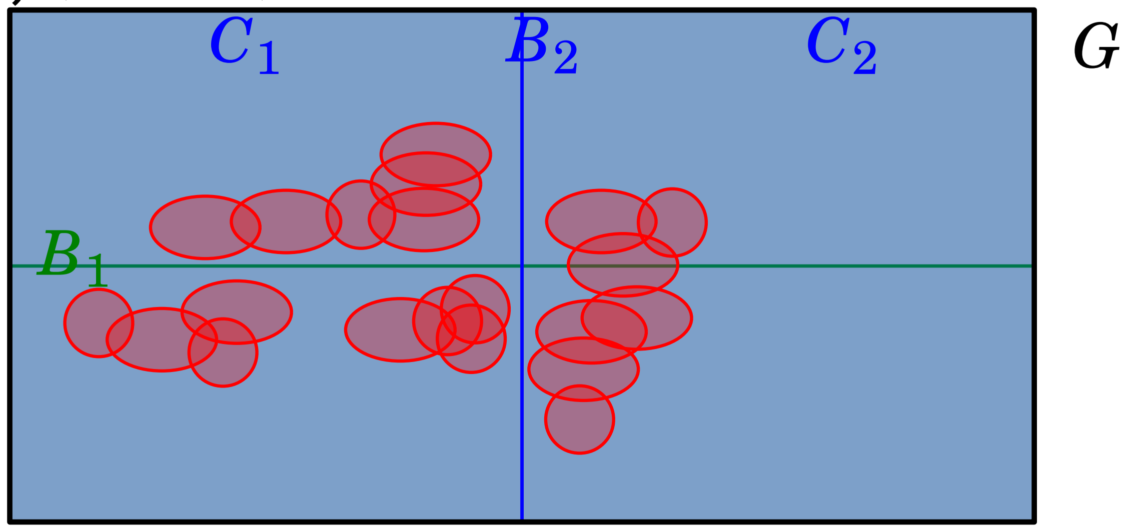
The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000\log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100\log n$ times, do

PHASE

- Run $\mathcal{MPX}$ to get an (α, d)-decomposition
- $\mathcal{C}$ = current cluster
- Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
- Flip all sets in σ, in order
- $\lambda_{i+1} = \lambda_i + 10\varepsilon$

- Return current coloring



Property 1 + Property 2

Claim 1: *The running time of the algorithm is* $O(d \cdot \log n) = O(\log^2 n/\alpha) = O(\log^6 n)$

Claim 2: *After phase* i*, any min. imp. set with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies "very close" to all previous border nodes*

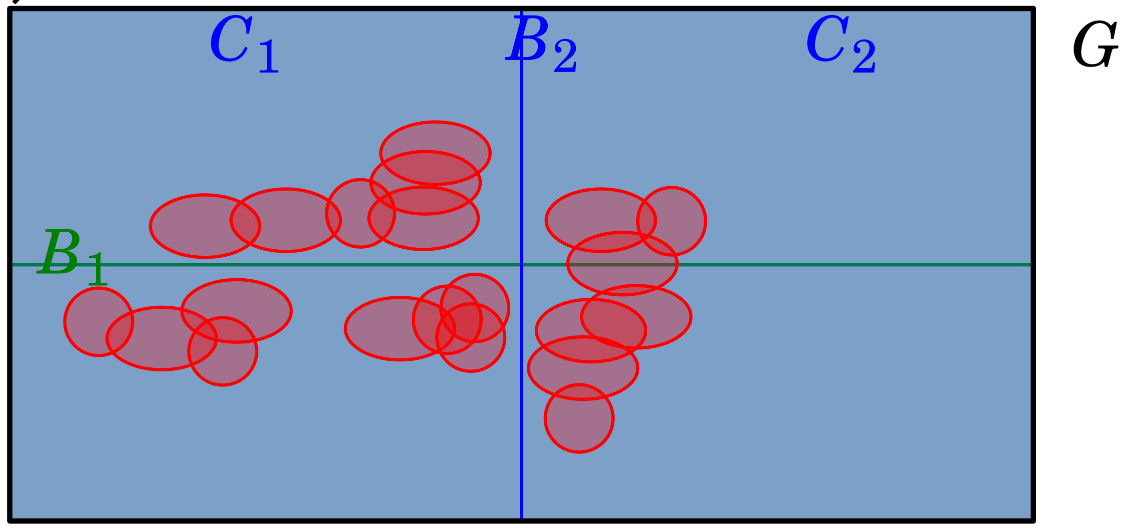
Claim 3: *With probability* $\geq 1 - 1/n^{10}$*, there is no error after* $100\log n$ *phases*

The algorithm

- Set $\lambda_1 = 1/4$ (*initial IR*), $\varepsilon = \lambda/(2000 \log n)$ (*Properties 1,2*), $\alpha = \Theta(\varepsilon^2/\log^2 n)$ ($\mathcal{MPX}$)
- Start with a random coloring assignment
- For $i = 1$ to $100 \log n$ times, do

PHASE

- Run $\mathcal{MPX}$ to get an (α, d)-decomposition
- $\mathcal{C}$ = current cluster
- Find maximal sequence σ of λ_i-improving sets of diam. $O(\log n/\varepsilon)$ inside $G[\mathcal{C}]$ that do not contain border nodes
- Flip all sets in σ, in order
- $\lambda_{i+1} = \lambda_i + 10\varepsilon$



- Return current coloring

Property 1 + Property 2

Claim 1: *The running time of the algorithm is* $O(d \cdot \log n) = O(\log^2 n/\alpha) = O(\log^6 n)$

Claim 2: *After phase* i*, any min. imp. set with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies "very close" to all previous border nodes*

Claim 3: *With probability* $\geq 1 - 1/n^{10}$*, there is no error after* $100 \log n$ *phases* ← $\mathcal{MPX}$ guarantees + Claim 2

Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius $\boldsymbol{r}$ and max degree $\boldsymbol{\Delta}$

Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ

locally optimal cut with
$\boldsymbol{\Delta = 3}$ (r = 1)

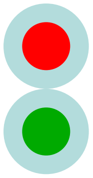
valid neighborhoods

Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ

locally optimal cut with
$\Delta = 3$ (r = 1)

valid neighborhoods



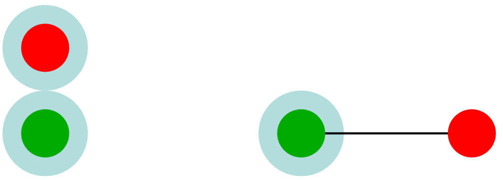
Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius $\boldsymbol{r}$ and max degree $\boldsymbol{\Delta}$

locally optimal cut with

$\boldsymbol{\Delta = 3}$ (r = 1)

valid neighborhoods

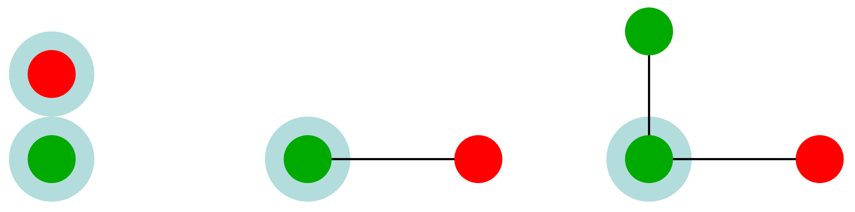


Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ

locally optimal cut with
$\Delta = 3$ (r = 1)

valid neighborhoods



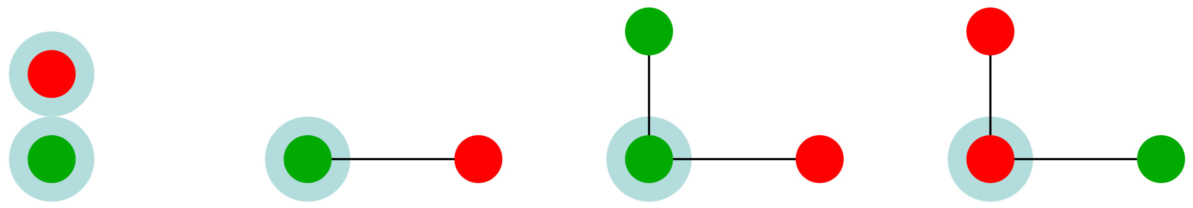
Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius $\boldsymbol{r}$ and max degree Δ

locally optimal cut with

$\Delta = 3$ (r = 1)

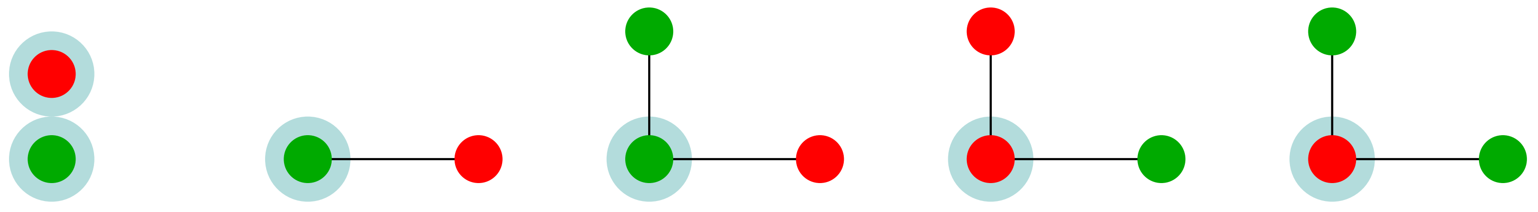
valid neighborhoods



- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ

locally optimal cut with
$\Delta = 3$ (r = 1)

valid neighborhoods



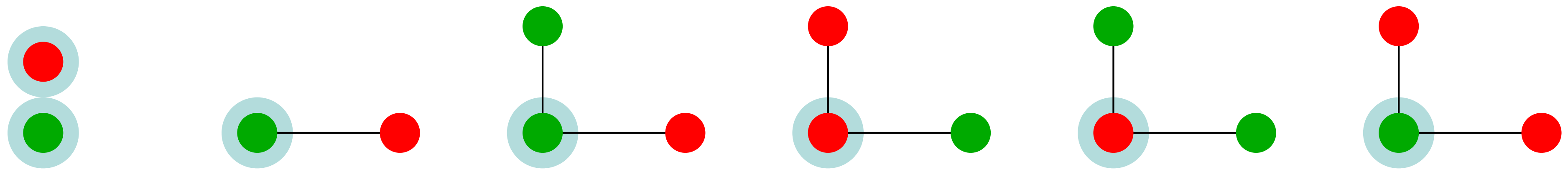
Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ

locally optimal cut with

$\Delta = 3$ (r = 1)

valid neighborhoods

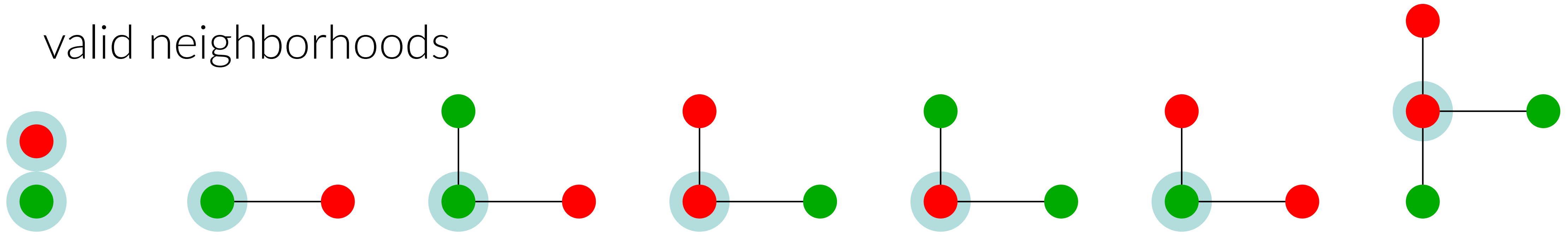


- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ

locally optimal cut with

$\Delta = 3$ (r = 1)

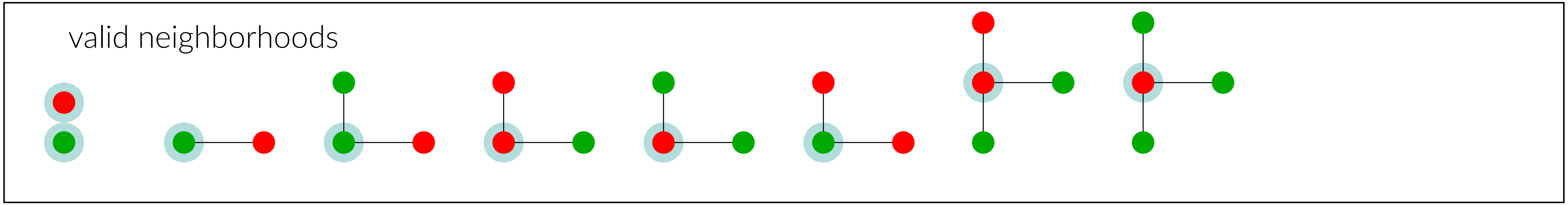
valid neighborhoods



Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ

locally optimal cut with
$\Delta = 3$ (r = 1)

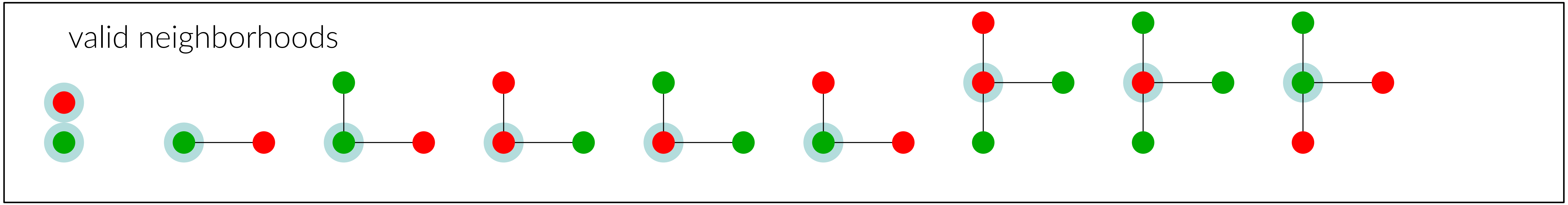


Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ

locally optimal cut with

$\Delta = 3$ (r = 1)

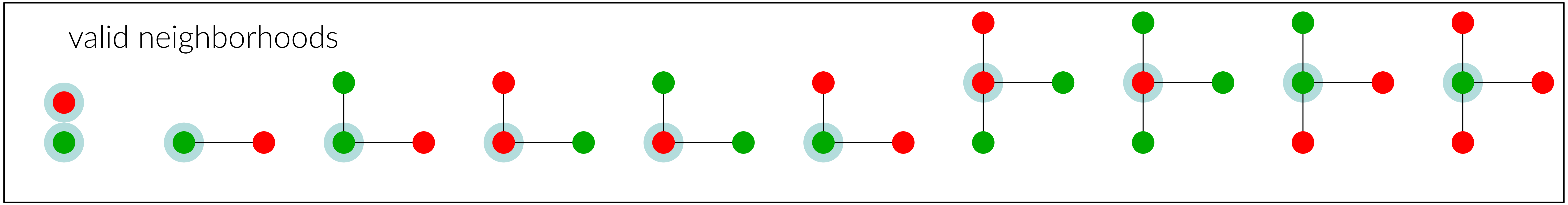


Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius $\boldsymbol{r}$ and max degree $\boldsymbol{\Delta}$

locally optimal cut with

$\boldsymbol{\Delta = 3}$ (r = 1)



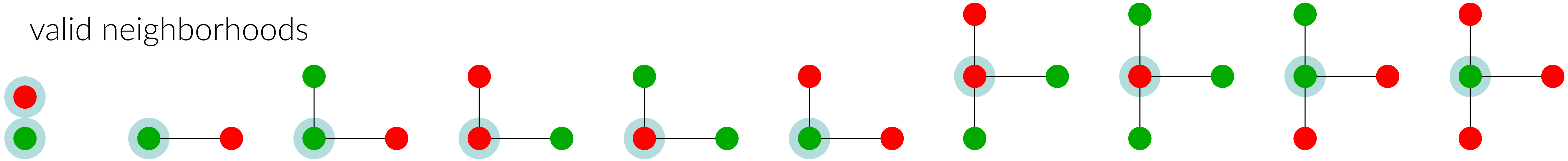
Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ

locally optimal cut with

$\Delta = 3$ (r = 1)

valid neighborhoods

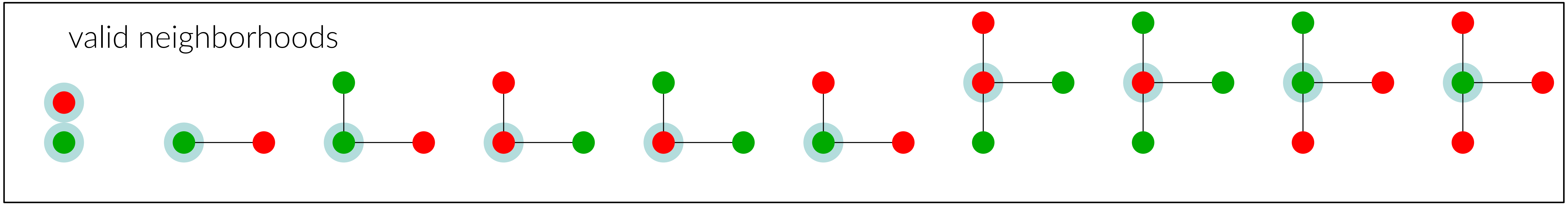


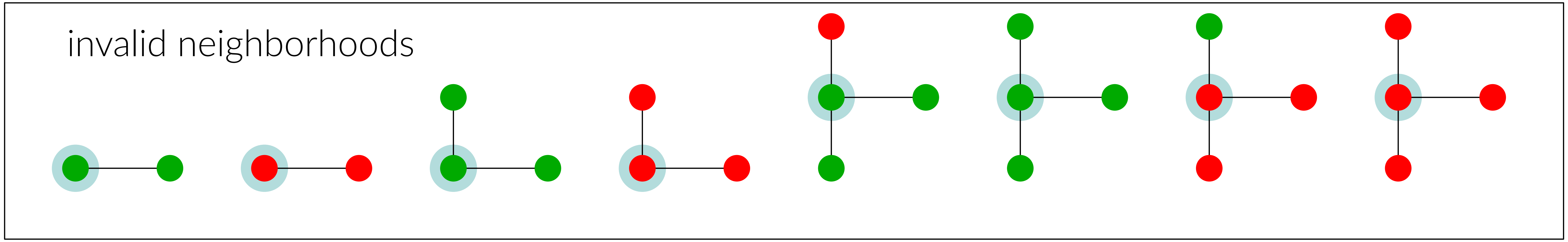
invalid neighborhoods

Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius $\boldsymbol{r}$ and max degree $\boldsymbol{\Delta}$

locally optimal cut with
$\boldsymbol{\Delta = 3}$ (r = 1)

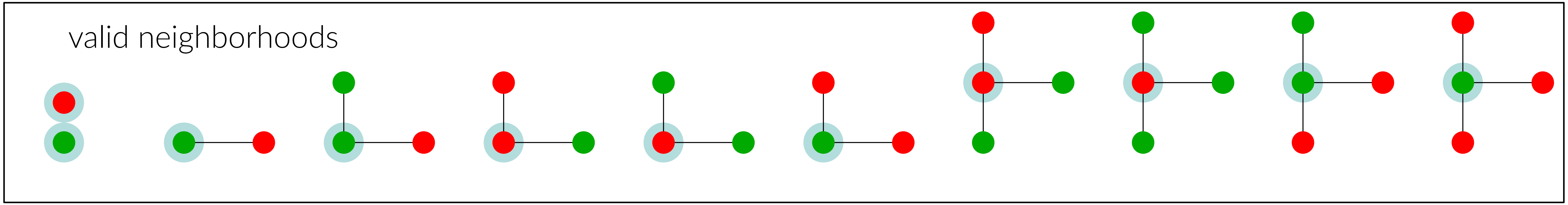


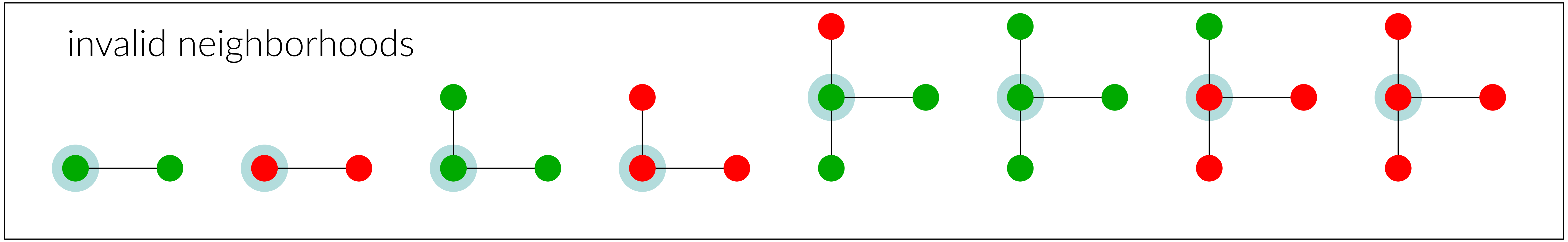


Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ
- $\mathrm{Pot} : \mathcal{G}_{r,\Delta} \to \mathbb{R}_{\geq 0}$ assigns a potential to every labeled neighborhood of radius r and max degree Δ

locally optimal cut with
$\Delta = 3$ (r = 1)

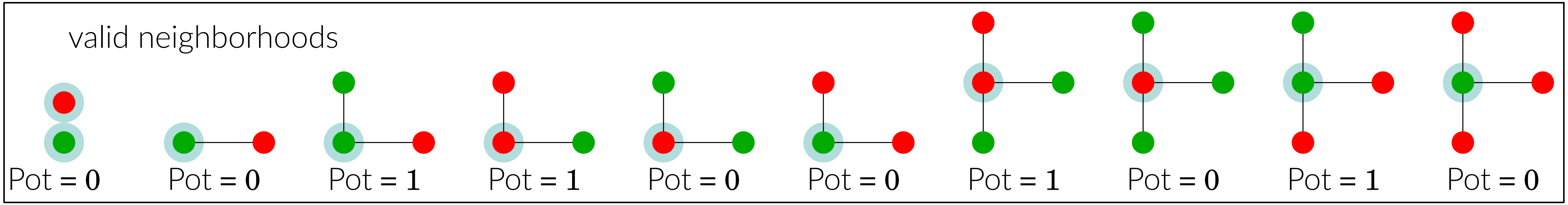


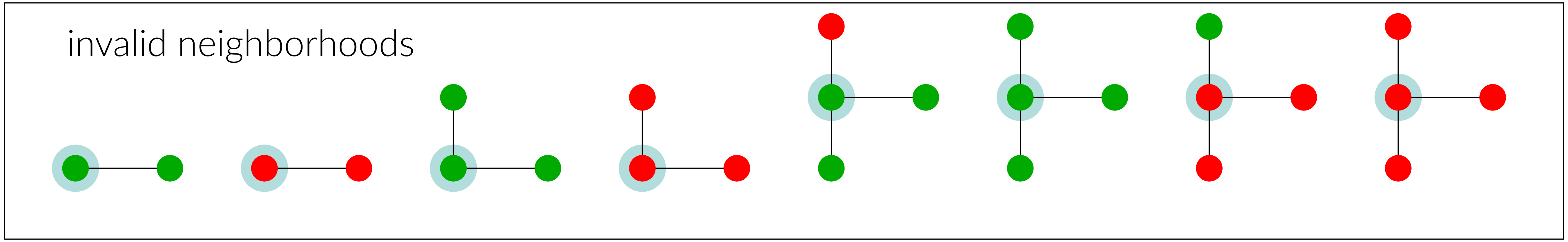


Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ
- $\text{Pot} : \mathcal{G}_{r,\Delta} \to \mathbb{R}_{\geq 0}$ assigns a potential to every labeled neighborhood of radius r and max degree Δ

locally optimal cut with
$\Delta = 3$ (r = 1)

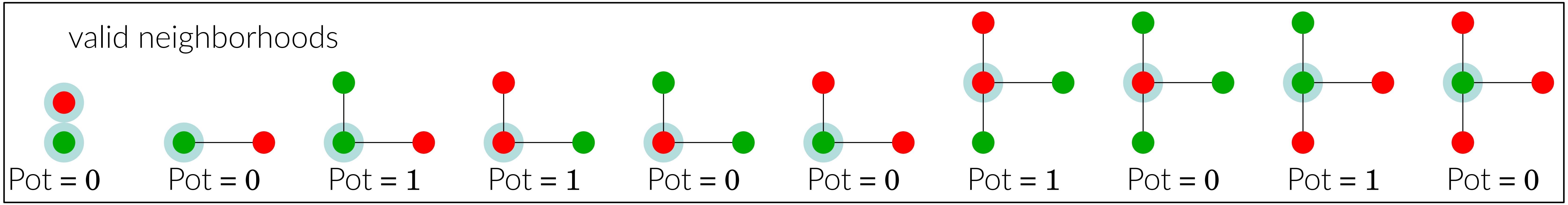


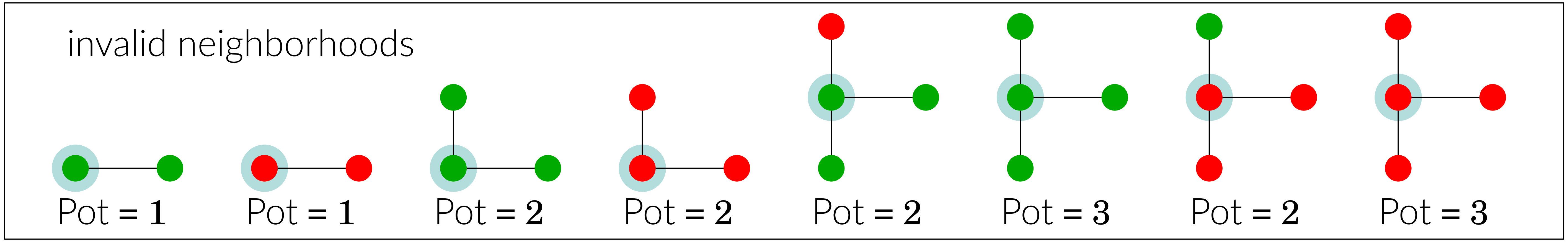


Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ
- $\text{Pot} : \mathcal{G}_{r,\Delta} \to \mathbb{R}_{\geq 0}$ assigns a potential to every labeled neighborhood of radius r and max degree Δ

locally optimal cut with
$\Delta = 3$ (r = 1)



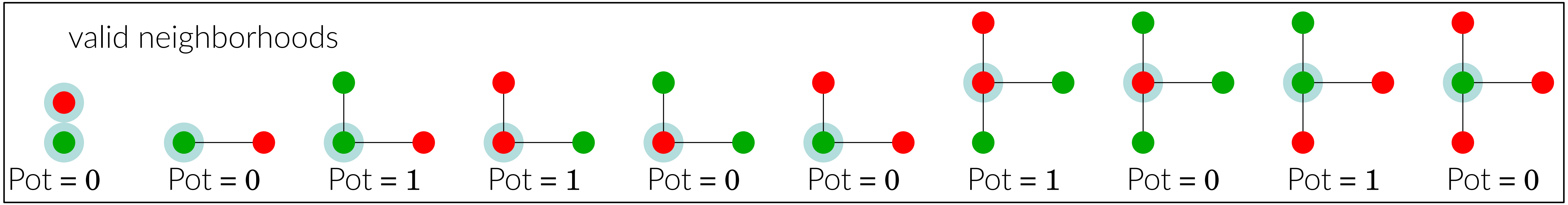


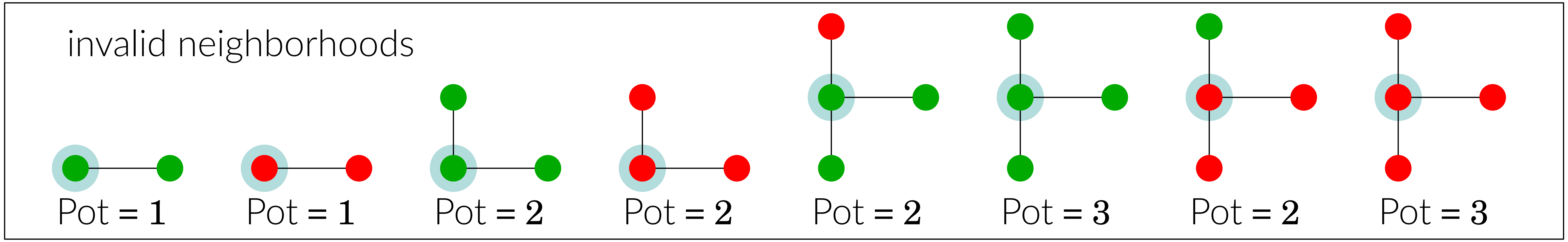
Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ
- $\mathrm{Pot} : \mathcal{G}_{r,\Delta} \to \mathbb{R}_{\geq 0}$ assigns a potential to every labeled neighborhood of radius r and max degree Δ
- For every *invalid neighborhood*, there is a way to relabel the central node to decrease the local (thus, global) potential

locally optimal cut with

$\Delta = 3$ (r = 1)

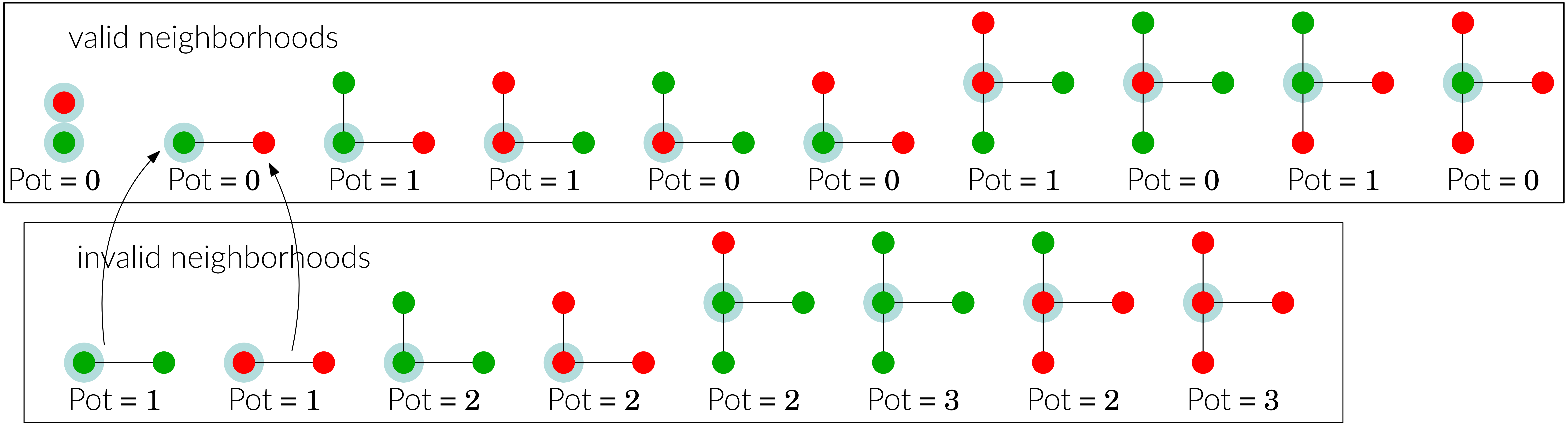




Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ
- $\mathrm{Pot} : \mathcal{G}_{r,\Delta} \to \mathbb{R}_{\geq 0}$ assigns a potential to every labeled neighborhood of radius r and max degree Δ
- For every *invalid neighborhood*, there is a way to relabel the central node to decrease the local (thus, global) potential

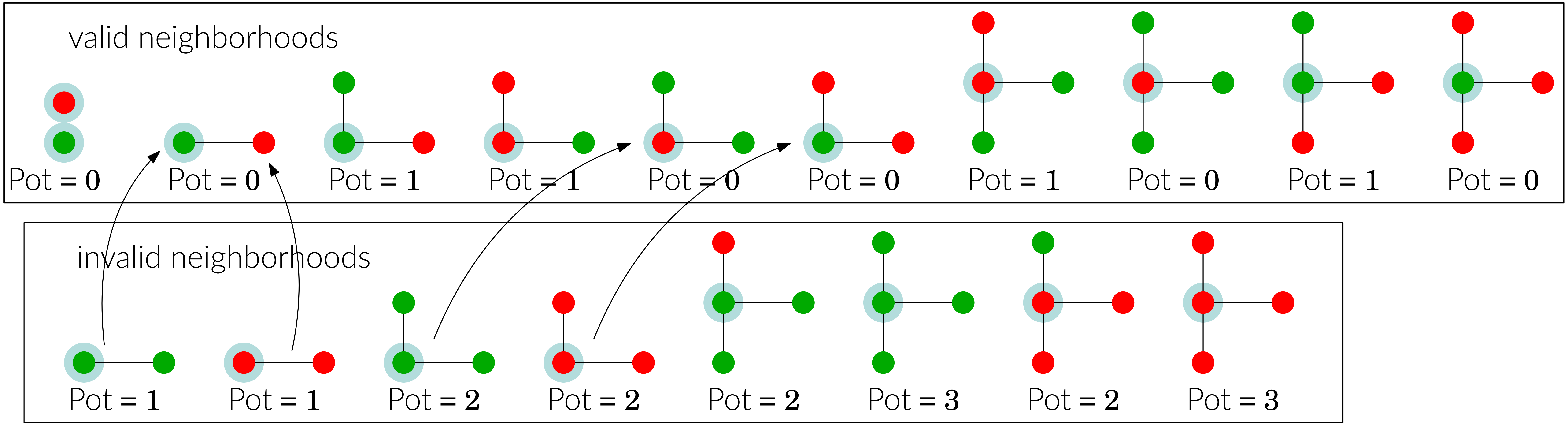
locally optimal cut with $\Delta = 3$ (r = 1)



Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ
- $\mathrm{Pot} : \mathcal{G}_{r,\Delta} \to \mathbb{R}_{\geq 0}$ assigns a potential to every labeled neighborhood of radius r and max degree Δ
- For every *invalid neighborhood*, there is a way to relabel the central node to decrease the local (thus, global) potential

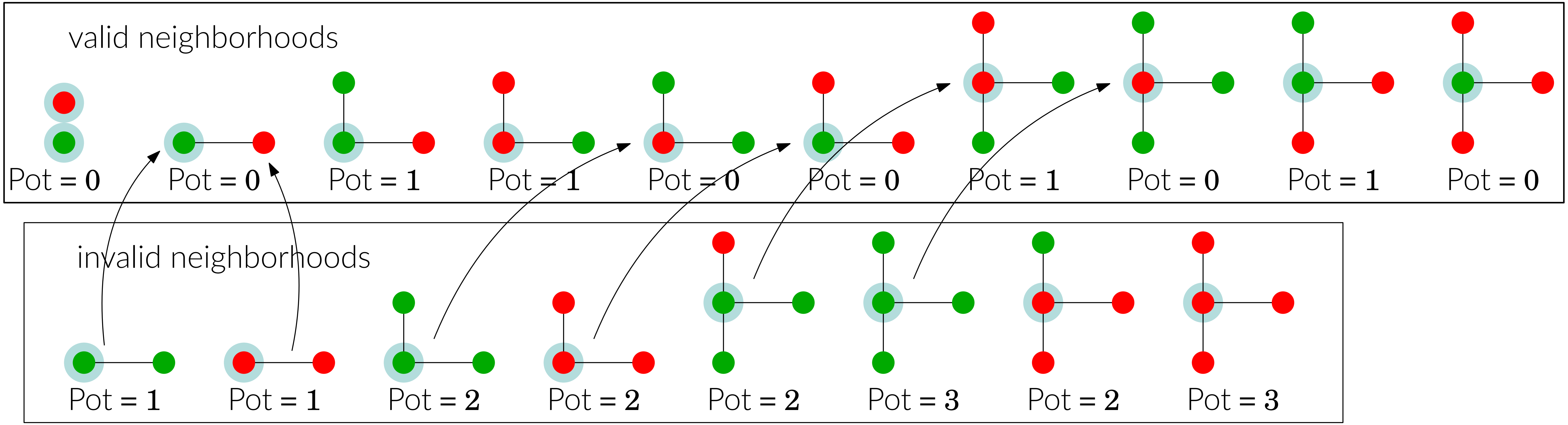
locally optimal cut with
$\Delta = 3$ (r = 1)



Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ
- $\mathrm{Pot} : \mathcal{G}_{r,\Delta} \to \mathbb{R}_{\geq 0}$ assigns a potential to every labeled neighborhood of radius r and max degree Δ
- For every *invalid neighborhood*, there is a way to relabel the central node to decrease the local (thus, global) potential

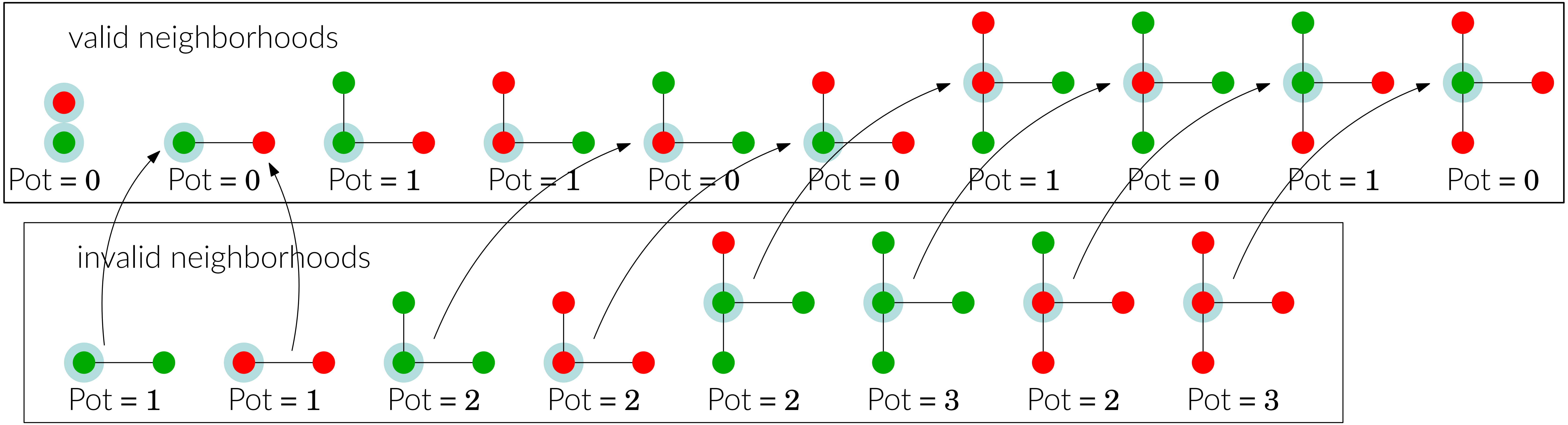
locally optimal cut with
$\Delta = 3$ (r = 1)



Local potential problems

- $r > 0$, $\Delta > 0$
- $\mathcal{C}$ list of valid neighborhoods of radius r and max degree Δ
- $\mathrm{Pot} : \mathcal{G}_{r,\Delta} \to \mathbb{R}_{\geq 0}$ assigns a potential to every labeled neighborhood of radius r and max degree Δ
- For every *invalid neighborhood*, there is a way to relabel the central node to decrease the local (thus, global) potential

locally optimal cut with $\Delta = 3$ (r = 1)



Recap & conclusions

Theorem:

For any local potential problem Π, *there exists a randomized LOCAL algorithm that solves* Π *with high probability in time* $O(\log^6 n)$. *The latter can be derandomized into a deterministic LOCAL algorithm that solves* Π *in time* $O(\log^8 n \text{ poly}(\log\log n))$.

Recap & conclusions

Theorem:

For any local potential problem Π, there exists a randomized LOCAL algorithm that solves Π with high probability in time $O(\log^6 n)$. The latter can be derandomized into a deterministic LOCAL algorithm that solves Π in time $O(\log^8 n$ poly$(\log\log n))$.

Lower bound:

- $\Omega(\log n)$-rounds in deterministic LOCAL (in *bounded-degree* trees)
- $\Omega(\log\log n)$-rounds in randomized LOCAL (in *bounded-degree* trees)

Recap & conclusions

Theorem:

For any local potential problem Π, there exists a randomized LOCAL algorithm that solves Π with high probability in time $O(\log^6 n)$. The latter can be derandomized into a deterministic LOCAL algorithm that solves Π in time $O(\log^8 n \text{ poly}(\log\log n))$.

Lower bound:

- $\Omega(\log n)$-rounds in deterministic LOCAL (in *bounded-degree* trees)
- $\Omega(\log\log n)$-rounds in randomized LOCAL (in *bounded-degree* trees)

Questions:

- Right deterministic complexity? Polynomial gap

Recap & conclusions

Theorem:

For any local potential problem Π, *there exists a randomized LOCAL algorithm that solves* Π *with high probability in time* $O(\log^6 n)$. *The latter can be derandomized into a deterministic LOCAL algorithm that solves* Π *in time* $O(\log^8 n \ \textit{poly}(\log\log n))$.

Lower bound:

- $\Omega(\log n)$-rounds in deterministic LOCAL (in *bounded-degree* trees)
- $\Omega(\log\log n)$-rounds in randomized LOCAL (in *bounded-degree* trees)

Questions:

- Right deterministic complexity? Polynomial gap
- Right randomized complexity? Exponential gap

Recap & conclusions

Theorem:

For any local potential problem Π, there exists a randomized LOCAL algorithm that solves Π with high probability in time $O(\log^6 n)$. The latter can be derandomized into a deterministic LOCAL algorithm that solves Π in time $O(\log^8 n \; poly(\log\log n))$.

Lower bound:

- $\Omega(\log n)$-rounds in deterministic LOCAL (in *bounded-degree* trees)
- $\Omega(\log\log n)$-rounds in randomized LOCAL (in *bounded-degree* trees)

Questions:

- Right deterministic complexity? Polynomial gap
- Right randomized complexity? Exponential gap
- Dependency in Δ? We actually give a $O(\text{poly}(\Delta)\text{poly}(\log n))$-time randomized/deterministic algorithm

Recap & conclusions

Theorem:

For any local potential problem Π, there exists a randomized LOCAL algorithm that solves Π with high probability in time $O(\log^6 n)$. The latter can be derandomized into a deterministic LOCAL algorithm that solves Π in time $O(\log^8 n$ poly$(\log\log n))$.

Lower bound:
- $\Omega(\log n)$-rounds in deterministic LOCAL (in *bounded-degree* trees)
- $\Omega(\log\log n)$-rounds in randomized LOCAL (in *bounded-degree* trees)

Questions:
- Right deterministic complexity? Polynomial gap
- Right randomized complexity? Exponential gap
- Dependency in Δ? We actually give a $O(\text{poly}(\Delta)\text{poly}(\log n))$-time randomized/deterministic algorithm
  - some *further assumption on the problem family*: it includes locally optimal cut

Recap & conclusions

Theorem:

For any local potential problem Π, there exists a randomized LOCAL algorithm that solves Π with high probability in time $O(\log^6 n)$. The latter can be derandomized into a deterministic LOCAL algorithm that solves Π in time $O(\log^8 n \text{ poly}(\log\log n))$.

Lower bound:

- $\Omega(\log n)$-rounds in deterministic LOCAL (in *bounded-degree* trees)
- $\Omega(\log\log n)$-rounds in randomized LOCAL (in *bounded-degree* trees)

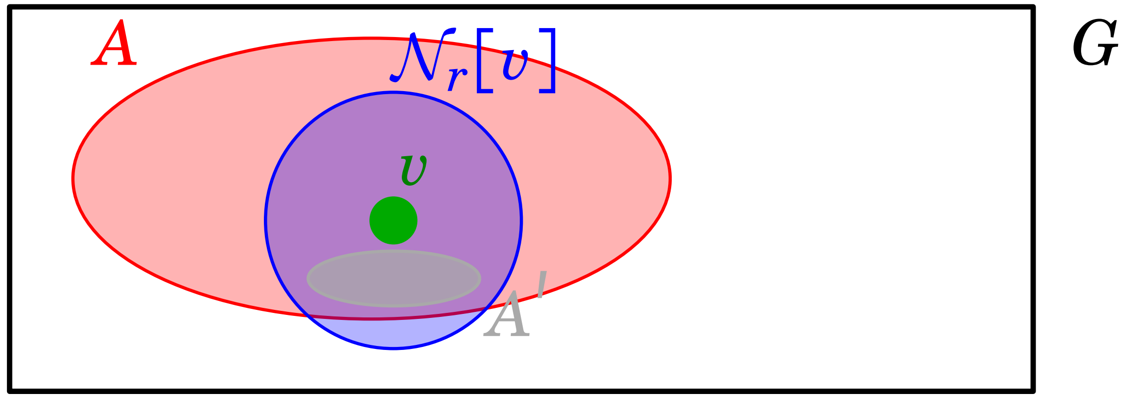
Questions:

- Right deterministic complexity? Polynomial gap
- Right randomized complexity? Exponential gap
- Dependency in Δ? We actually give a $O(\text{poly}(\Delta)\text{poly}(\log n))$-time randomized/deterministic algorithm
  - some *further assumption on the problem family*: it includes locally optimal cut

THANKS

Proof of Property 1

Property 1: on minimal improving sets

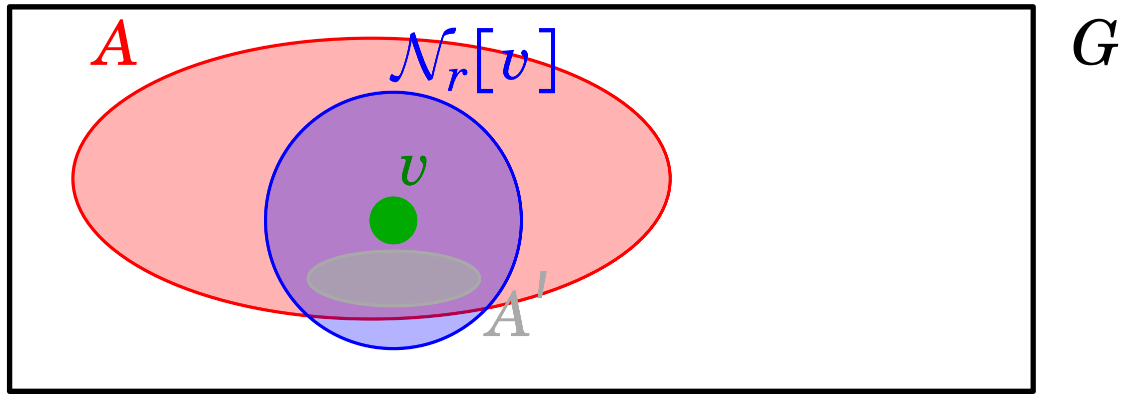


- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

$\implies$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Proof of Property 1

Property 1: on minimal improving sets



- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

$\implies$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

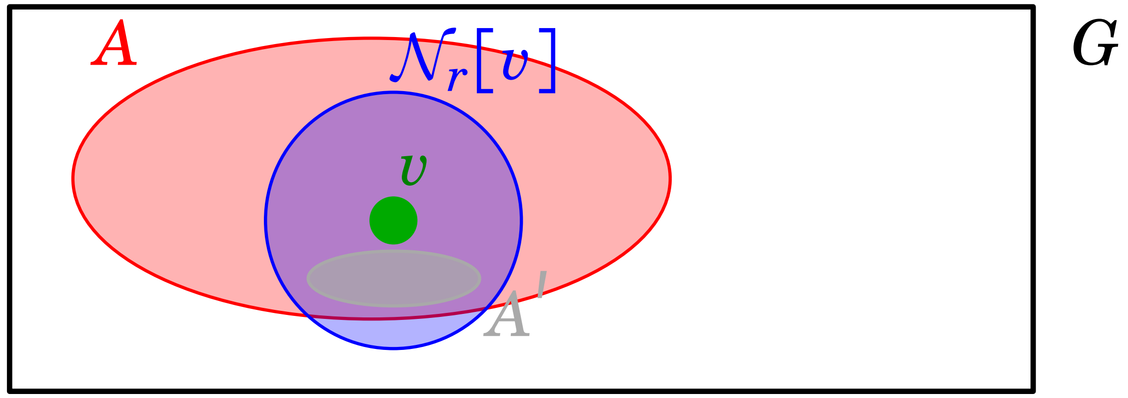
Proof 1

- $S_i = \mathcal{N}_i[v] \cap A$

Proof of Property 1

Property 1: on minimal improving sets

- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$



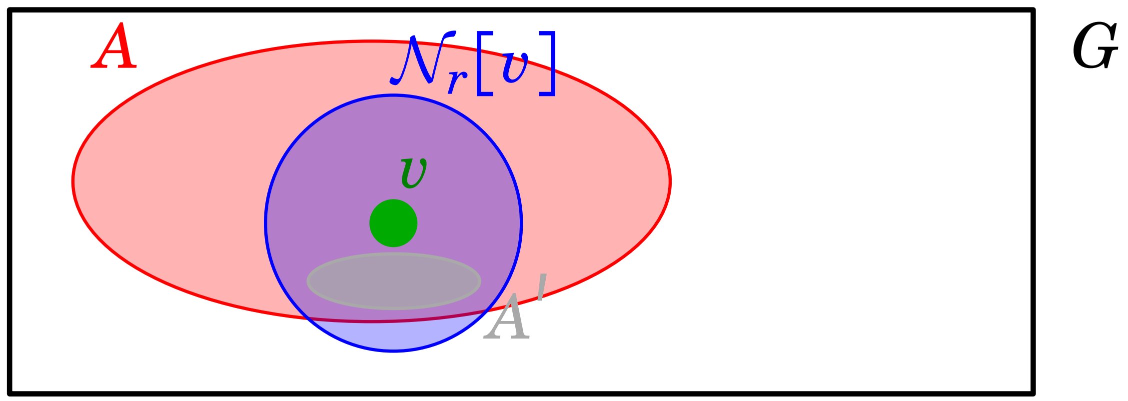
$\Longrightarrow$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Proof 1

- $S_i = \mathcal{N}_i[v] \cap A$
- $\exists i = O(\log n/\varepsilon)$ such that # edges in the cut (S_i, A) is $\leq (\varepsilon/2)|S_i|$

Proof of Property 1

Property 1: on minimal improving sets



- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

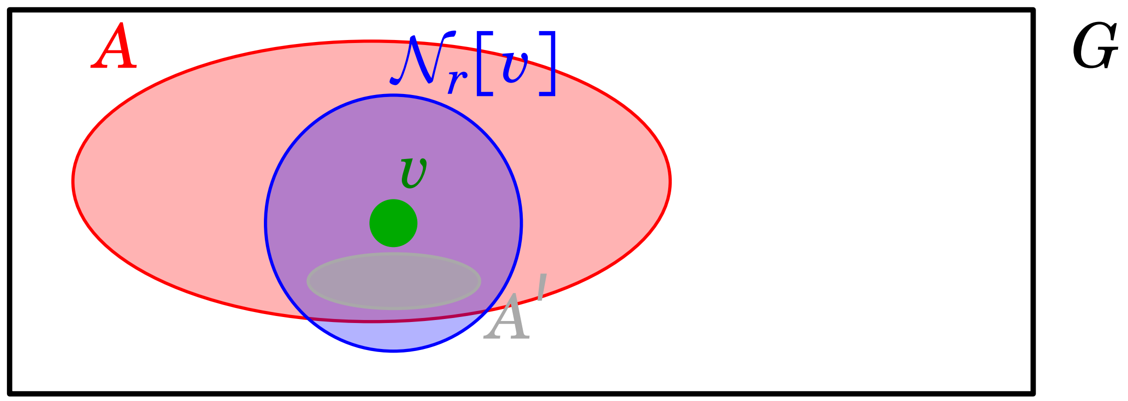
$\implies$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Proof 1

- $S_i = \mathcal{N}_i[v] \cap A$
- $\exists i = O(\log n/\varepsilon)$ such that # edges in the cut (S_i, A) is $\leq (\varepsilon/2)|S_i|$
  - if not, exponential growth of S_i: $|S_i| \geq (1+\varepsilon/(2\Delta))^i \implies |S_{100\Delta \log n/\varepsilon}| \geq e^{50 \log n}$

Proof of Property 1

Property 1: on minimal improving sets



- $A \subseteq V$ minimal improving set
- $\text{IR}(A) \geq x$
- $\varepsilon < x$

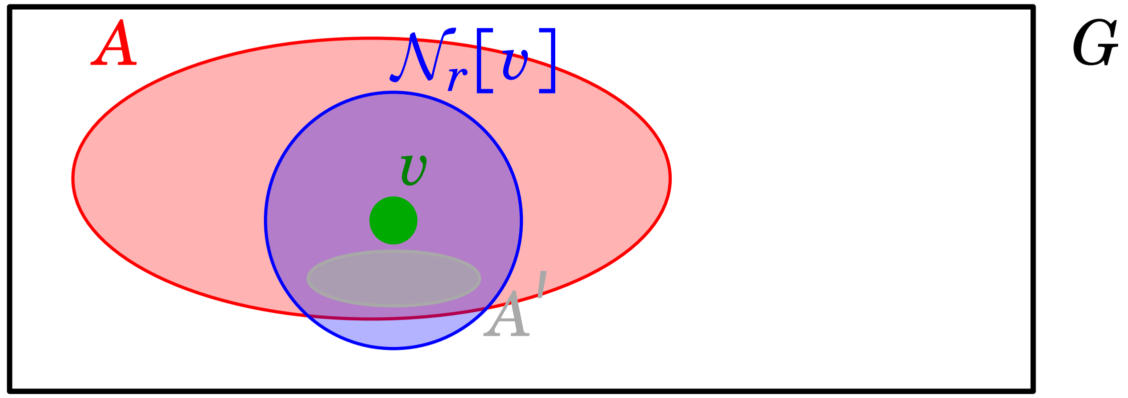
$\implies$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\text{IR}(A') \geq x - \varepsilon$

Proof 1

- $S_i = \mathcal{N}_i[v] \cap A$
- $\exists i = O(\log n/\varepsilon)$ such that # edges in the cut (S_i, A) is $\leq (\varepsilon/2)|S_i|$
  - if not, exponential growth of S_i: $|S_i| \geq (1 + \varepsilon/(2\Delta))^i \implies |S_{100\Delta \log n/\varepsilon}| \geq e^{50 \log n}$
- $\text{Imp}(A) \leq \text{Imp}(S_i) + \text{Imp}(A \setminus S_i) + 2\text{size}(S_i, A)$

Proof of Property 1

Property 1: on minimal improving sets



- $A \subseteq V$ minimal improving set
- $\mathrm{IR}(A) \geq x$
- $\varepsilon < x$

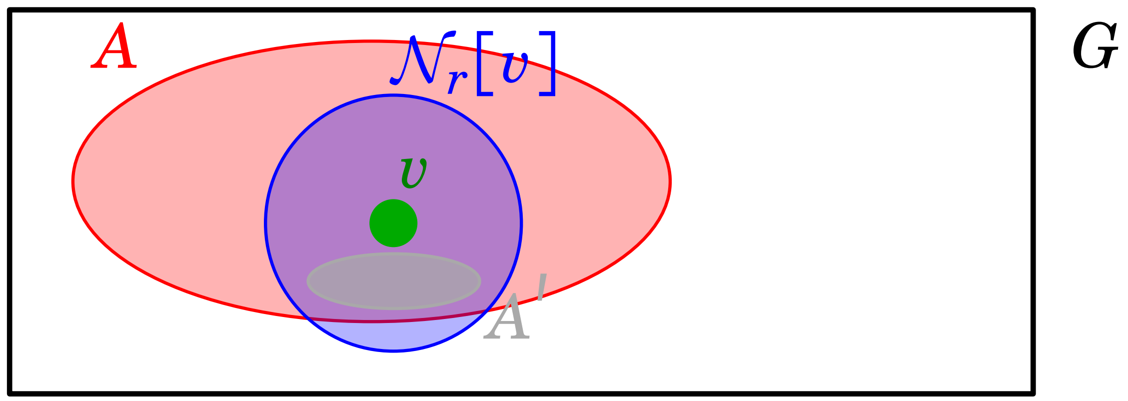
$\implies$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Proof 1

- $S_i = \mathcal{N}_i[v] \cap A$
- $\exists i = O(\log n/\varepsilon)$ such that # edges in the cut (S_i, A) is $\leq (\varepsilon/2)|S_i|$
  - if not, exponential growth of S_i: $|S_i| \geq (1 + \varepsilon/(2\Delta))^i \implies |S_{100\Delta \log n/\varepsilon}| \geq e^{50 \log n}$
- $\mathrm{Imp}(A) \leq \mathrm{Imp}(S_i) + \mathrm{Imp}(A \setminus S_i) + 2\mathrm{size}(S_i, A)$
- $\mathrm{Imp}(S_i) \geq \mathrm{IR}(A)(|A| - |A \setminus S_i|) - \varepsilon|S_i|$

Proof of Property 1

Property 1: on minimal improving sets



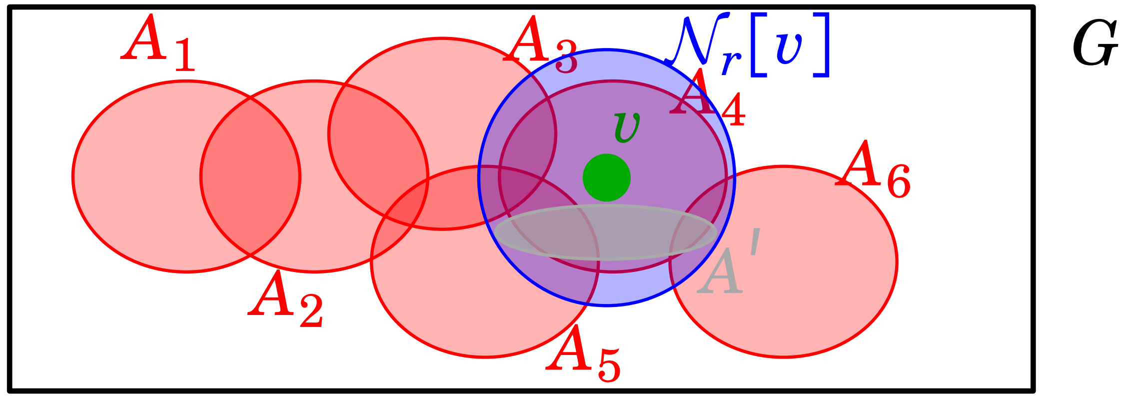
- $A \subseteq V$ minimal improving set
- $\text{IR}(A) \geq x$
- $\varepsilon < x$

$\implies$ for all $v \in A$, $\exists r = O(\log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\text{IR}(A') \geq x - \varepsilon$

Proof 1

- $S_i = \mathcal{N}_i[v] \cap A$
- $\exists i = O(\log n/\varepsilon)$ such that # edges in the cut (S_i, A) is $\leq (\varepsilon/2)|S_i|$
    - if not, exponential growth of S_i: $|S_i| \geq (1 + \varepsilon/(2\Delta))^i \implies |S_{100\Delta \log n/\varepsilon}| \geq e^{50 \log n}$
- $\text{Imp}(A) \leq \text{Imp}(S_i) + \text{Imp}(A \setminus S_i) + 2\text{size}(S_i, A)$
- $\text{Imp}(S_i) \geq \text{IR}(A)(|A| - |A \setminus S_i|) - \varepsilon|S_i|$
- $\text{Imp}(S_i) \geq \text{IR}(|S_i|) - \varepsilon|S_i| = (x - \varepsilon)|S_i|$

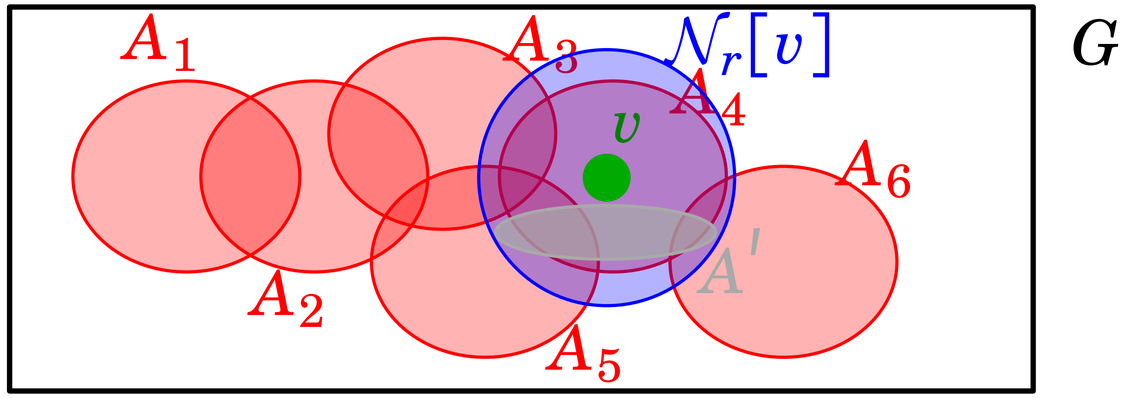
Property 2: on sequences of x-improving sets



- $A_1, \ldots, A_k \subseteq V$ sequence of x-improving sets
- $\mathrm{diam}(A_i) \leq d$
- $\varepsilon < x$
- $A = \cup_i A_i$

$\implies$ for all i, for all $v \in A$, $\exists r = O(d \log n / \varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Property 2: on sequences of x-improving sets



- $A_1, \ldots, A_k \subseteq V$ sequence of x-improving sets
- $\mathrm{diam}(A_i) \leq d$
- $\varepsilon < x$
- $A = \cup_i A_i$

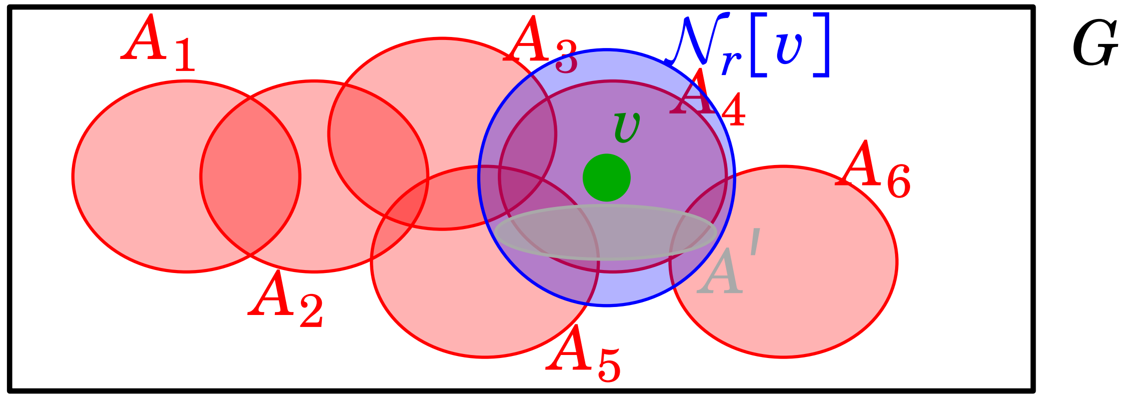
$\Longrightarrow$ for all i, for all $v \in A$, $\exists r = O(d \log n / \varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

Proof 2

- virtual graph H: nodes are A_is, edges are between influencing A_is

Property 2: on sequences of x-improving sets

- $A_1, \dots, A_k \subseteq V$ sequence of x-improving sets
- diam$(A_i) \le d$
- $\varepsilon < x$
- $A = \cup_i A_i$

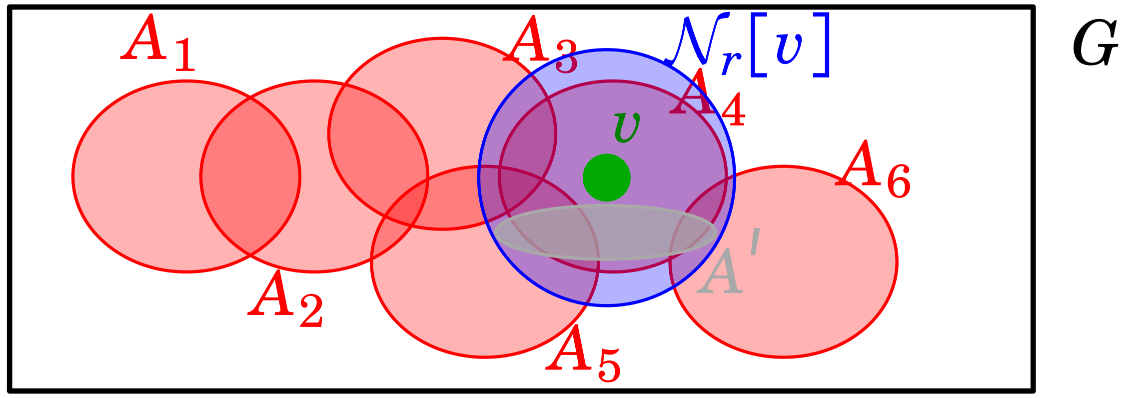


$\Longrightarrow$ for all i, for all $v \in A$, $\exists r = O(d \log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that IR$(A') \ge x - \varepsilon$

Proof 2

- virtual graph H: nodes are A_is, edges are between influencing A_is
- weights to nodes in $V(H)$: for each A_i, $w(A_i) = |A_i|$

Property 2: on sequences of x-improving sets



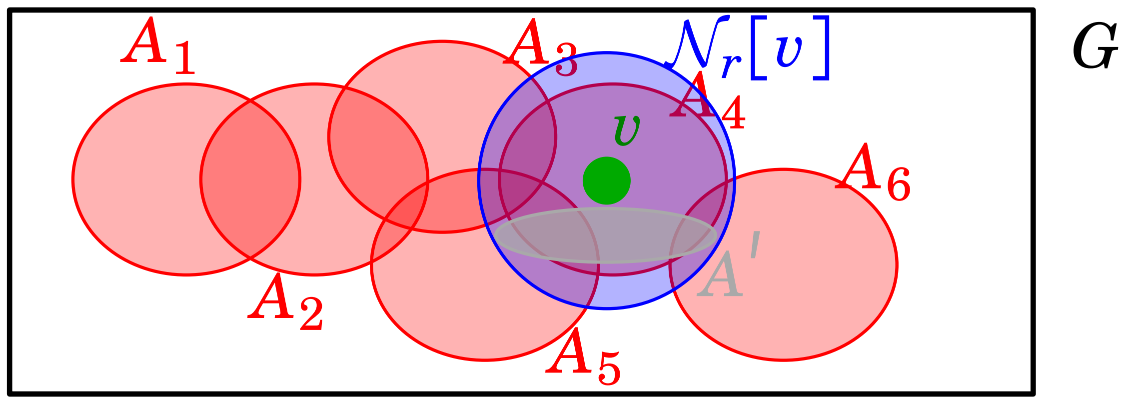
- $A_1, \ldots, A_k \subseteq V$ sequence of x-improving sets
- $\text{diam}(A_i) \leq d$
- $\varepsilon < x$
- $A = \cup_i A_i$

$\implies$ for all i, for all $v \in A$, $\exists r = O(d \log n / \varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\text{IR}(A') \geq x - \varepsilon$

Proof 2

- virtual graph H: nodes are A_is, edges are between influencing A_is
- weights to nodes in $V(H)$: for each A_i, $w(A_i) = |A_i|$
- $S_i = \mathcal{N}_i[A_i]$ (in H)

Property 2: on sequences of x-improving sets



- $A_1, \ldots, A_k \subseteq V$ sequence of x-improving sets
- diam$(A_i) \leq d$
- $\varepsilon < x$
- $A = \cup_i A_i$

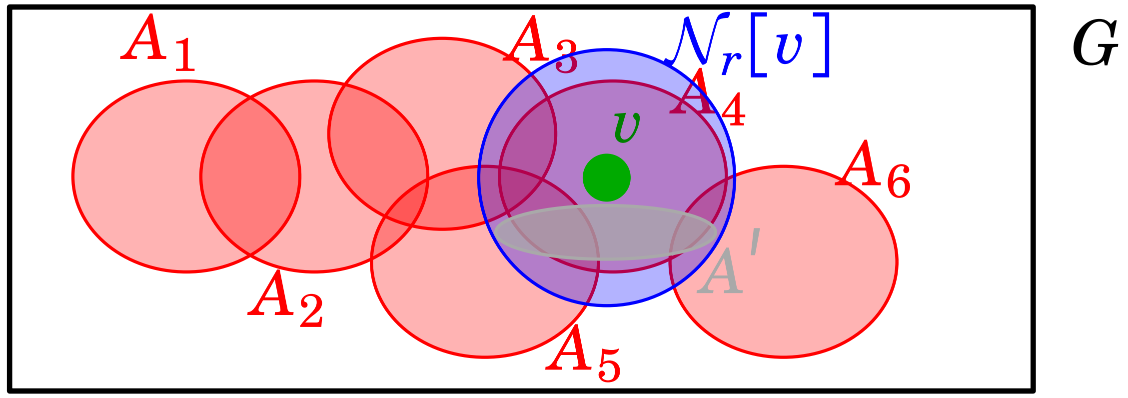
$\Longrightarrow$ for all i, for all $v \in A$, $\exists r = O(d \log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that IR$(A') \geq x - \varepsilon$

Proof 2

- virtual graph H: nodes are A_is, edges are between influencing A_is
- weights to nodes in $V(H)$: for each A_i, $w(A_i) = |A_i|$
- $S_i = \mathcal{N}_i[A_i]$ (in H)
- $\exists i = O(\log n/\varepsilon)$ such that $\sum_{A_j \in S_{i+1} \setminus S_i} w(A_j) \leq (\varepsilon/2) \sum_{A_j \in S_i} w(A_j)$

Proof of Property 2

Property 2: on sequences of x-improving sets



- $A_1, \ldots, A_k \subseteq V$ sequence of x-improving sets
- $\mathrm{diam}(A_i) \leq d$
- $\varepsilon < x$
- $A = \cup_i A_i$

$\implies$ for all i, for all $v \in A$, $\exists r = O(d \log n / \varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\mathrm{IR}(A') \geq x - \varepsilon$

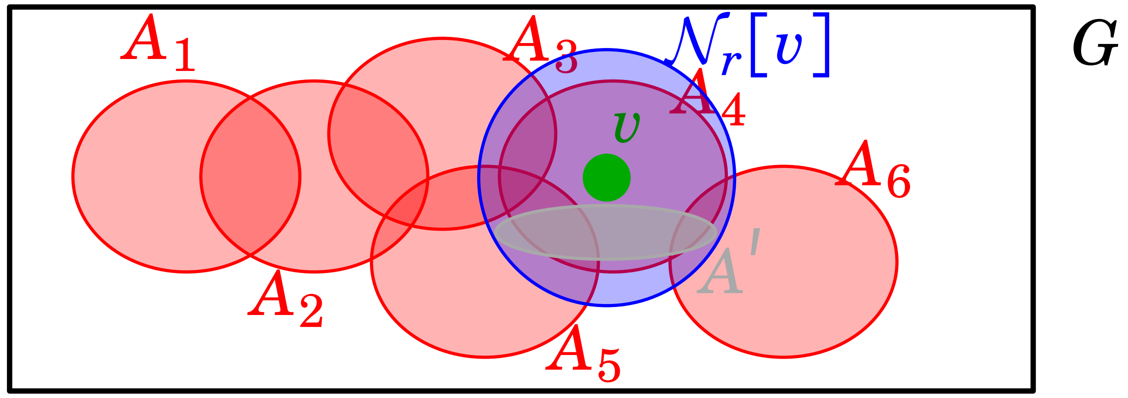
Proof 2

- virtual graph H: nodes are A_is, edges are between influencing A_is
- weights to nodes in $V(H)$: for each A_i, $w(A_i) = |A_i|$
- $S_i = \mathcal{N}_i[A_i]$ (in H)
- $\exists i = O(\log n / \varepsilon)$ such that $\sum_{A_j \in S_{i+1} \setminus S_i} w(A_j) \leq (\varepsilon/2) \sum_{A_j \in S_i} w(A_j)$
  - if not, exponential growth of weights but total weight is $O(n)$

Proof of Property 2

Property 2: on sequences of x-improving sets

- $A_1, \dots, A_k \subseteq V$ sequence of x-improving sets
- $\text{diam}(A_i) \leq d$
- $\varepsilon < x$
- $A = \cup_i A_i$



$\implies$ for all i, for all $v \in A$, $\exists r = O(d \log n/\varepsilon)$ and minimal improving set $A' \subseteq \mathcal{N}_r[v] \cap A$ such that $\text{IR}(A') \geq x - \varepsilon$

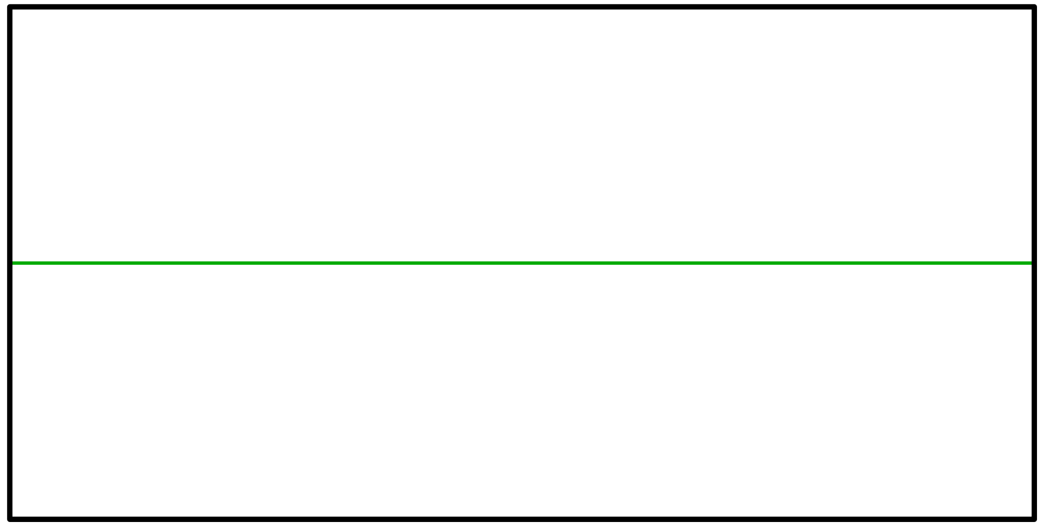
Proof 2

- virtual graph H: nodes are A_is, edges are between influencing A_is
- weights to nodes in $V(H)$: for each A_i, $w(A_i) = |A_i|$
- $S_i = \mathcal{N}_i[A_i]$ (in H)
- $\exists i = O(\log n/\varepsilon)$ such that $\sum_{A_j \in S_{i+1} \setminus S_i} w(A_j) \leq (\varepsilon/2) \sum_{A_j \in S_i} w(A_j)$
    - if not, exponential growth of weights but total weight is $O(n)$
- Similar to Lemma 1, but now to go back to G we need to multiply by $O(d)$ (diameter of the A_is)

Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase* i, *any MIS with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$



G

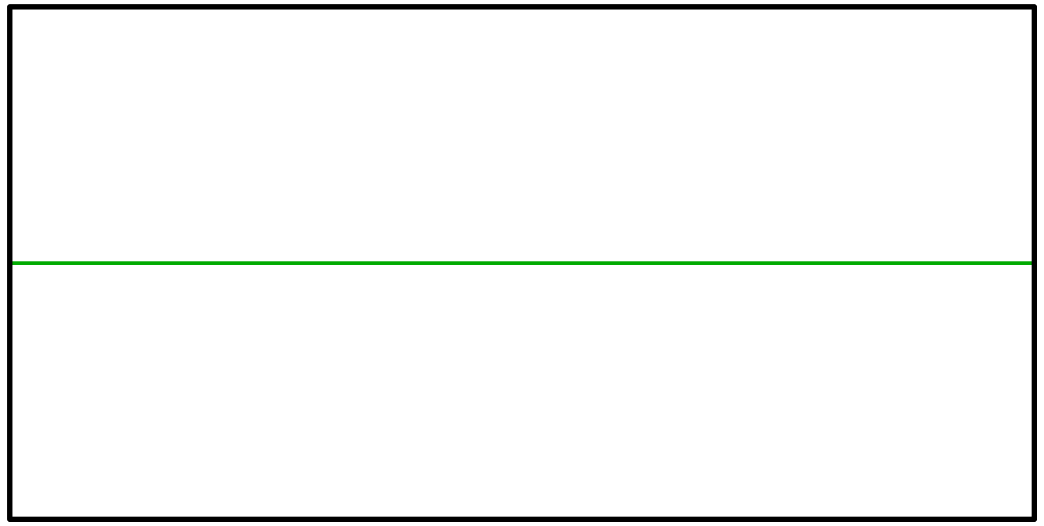
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000\log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$



G

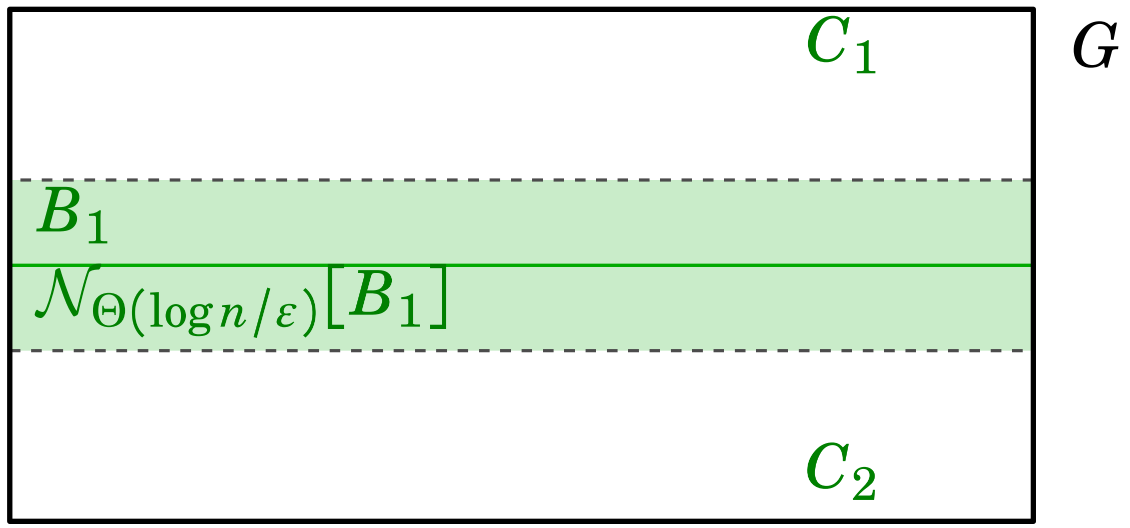
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$



Proof of Claim 2

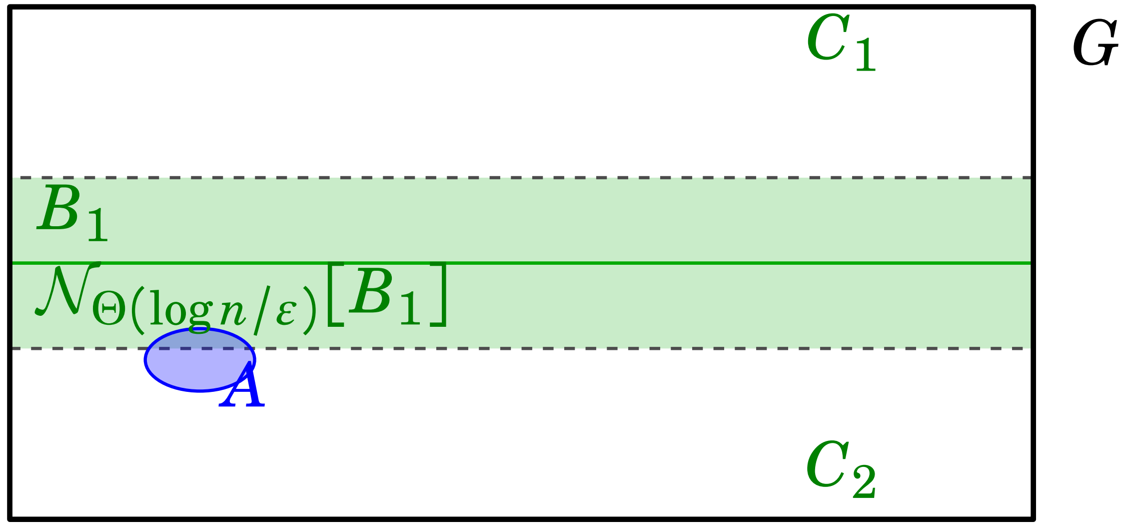
- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$

- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained



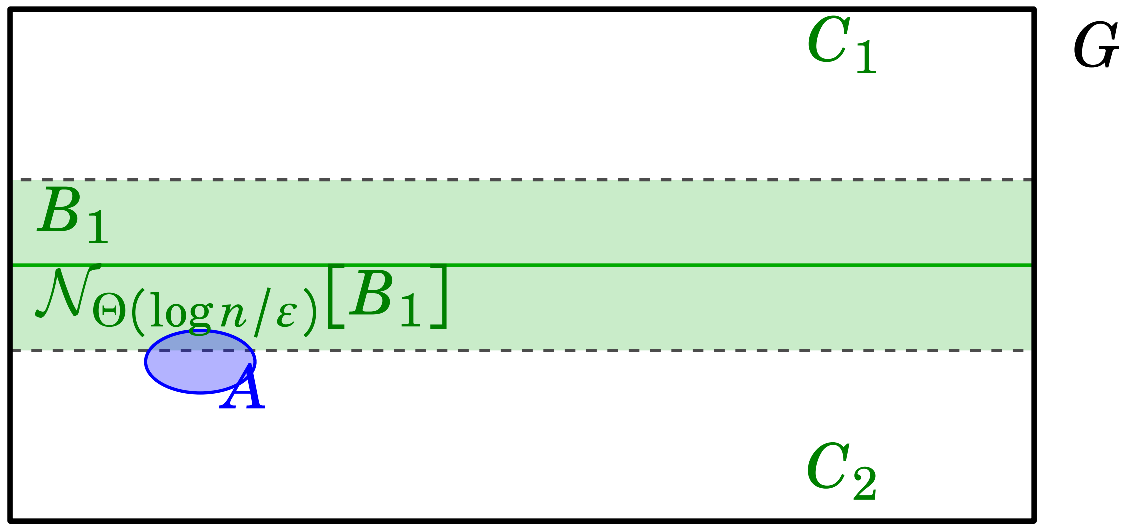
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000\log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\leq i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained
- Maximality of the λ_1-improving sequence! Contradiction



Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

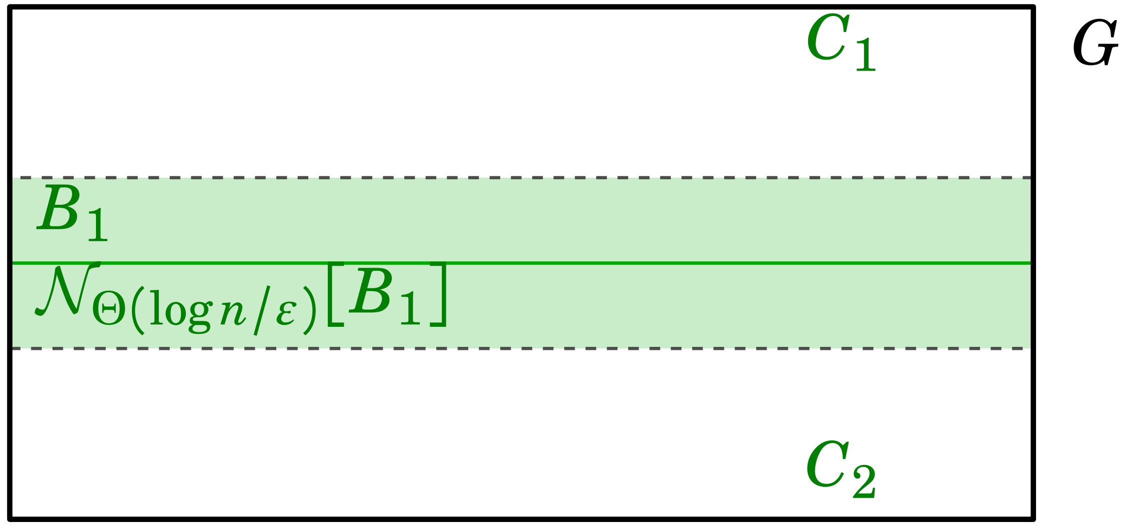
Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$

- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained

- Maximality of the λ_1-improving sequence! Contradiction

- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$



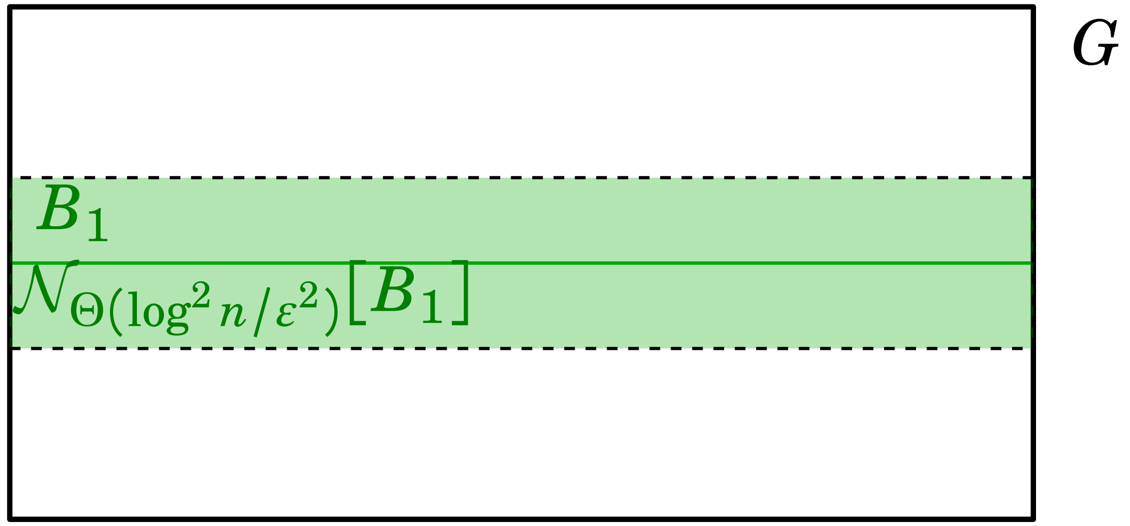
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained
- Maximality of the λ_1-improving sequence! Contradiction
- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$



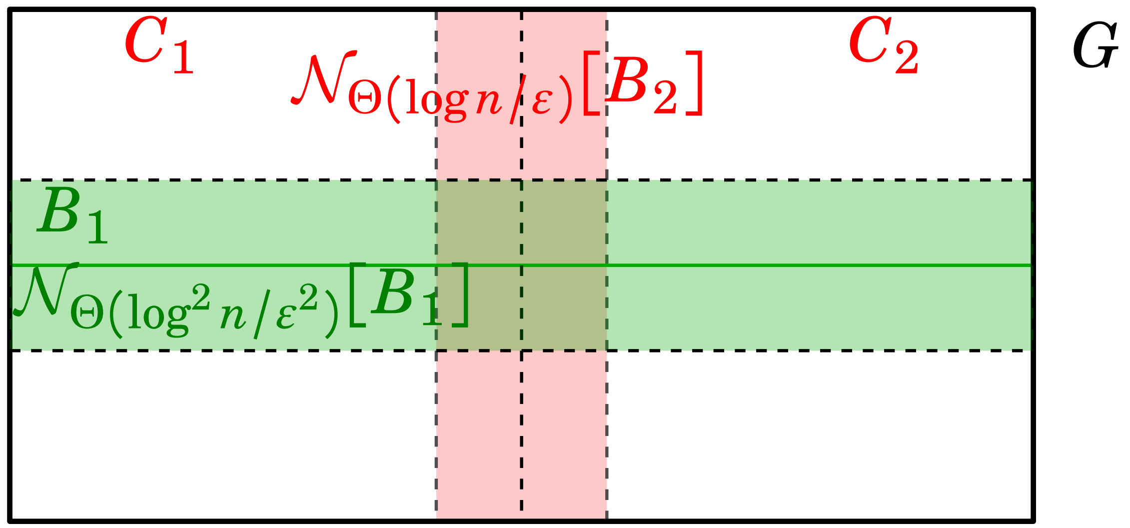
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained
- Maximality of the λ_1-improving sequence! Contradiction
- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$



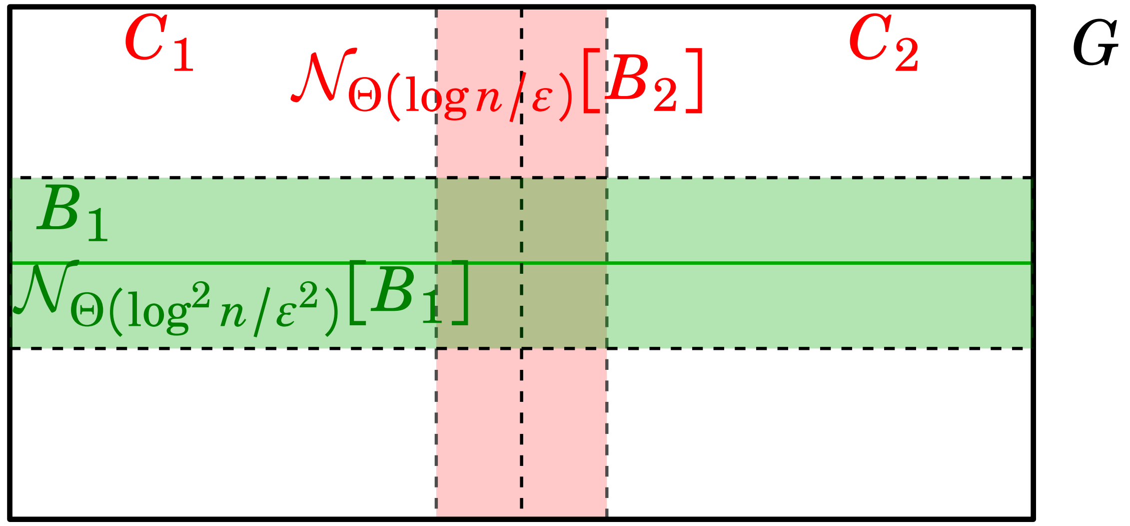
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000\log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\leq i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained
- Maximality of the λ_1-improving sequence! Contradiction
- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\leq i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_i$ is not contained



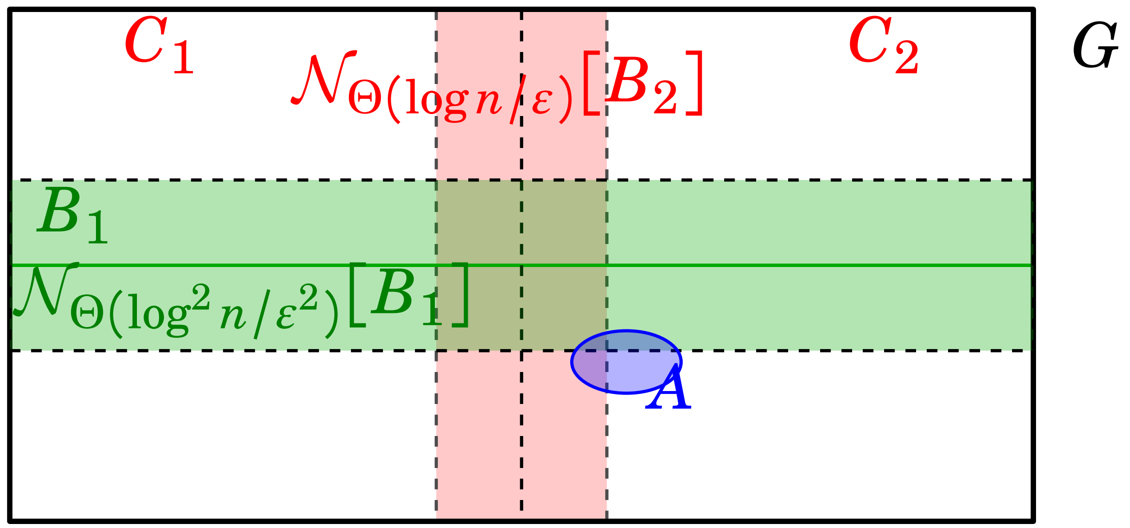
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR ≥ λ_i of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \le i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR ≥ λ_1 is not contained
- Maximality of the λ_1-improving sequence! Contradiction



- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \le i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$
- By contradiction, suppose min. imp. set A with IR ≥ λ_i is not contained

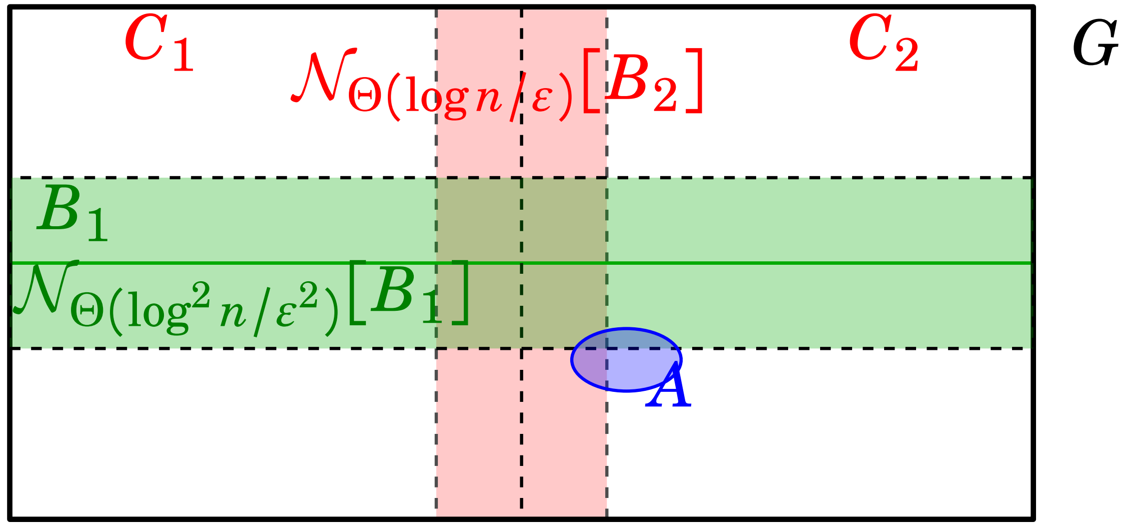
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000\log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\leq i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained
- Maximality of the λ_1-improving sequence! Contradiction



- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\leq i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_i$ is not contained
- A must be inside $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i]$ otherwise we break maximality

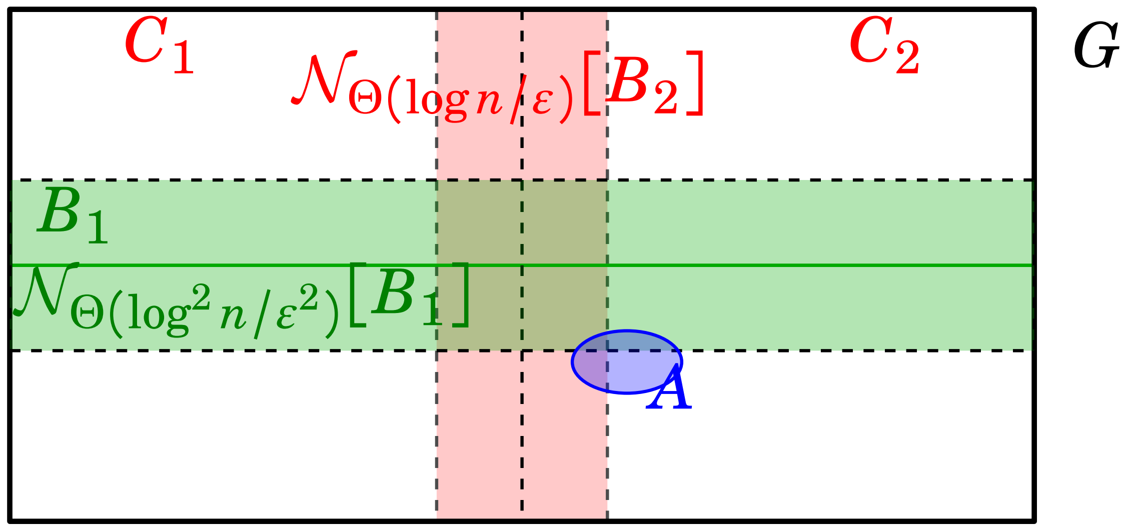
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained
- Maximality of the λ_1-improving sequence! Contradiction



- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_i$ is not contained
- A must be inside $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i]$ otherwise we break maximality
- $\exists j \leq i - 1$ such that A is not inside $\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j] \implies A$ is fully within some cluster C at phase j

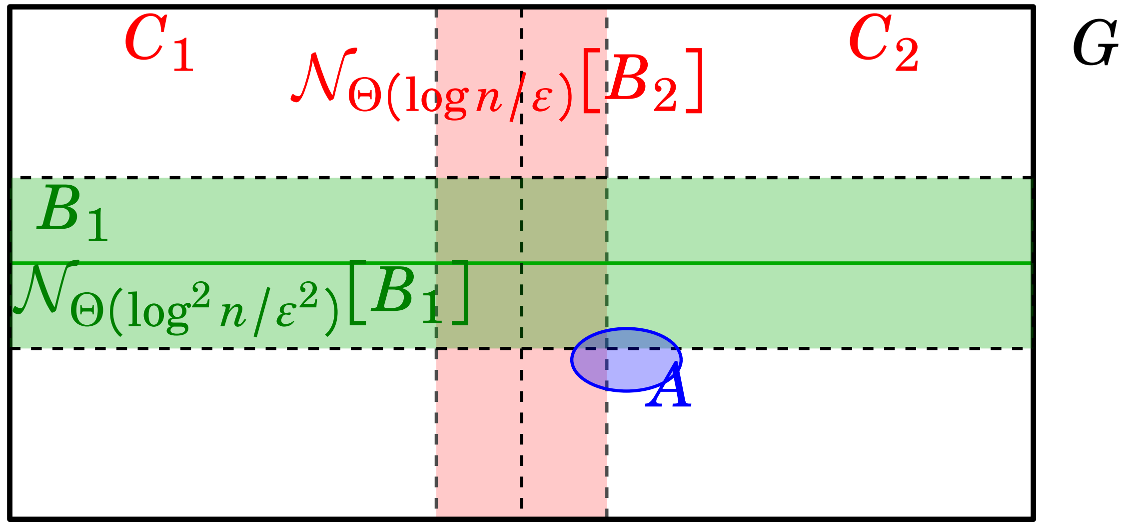
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000\log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\leq i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained
- Maximality of the λ_1-improving sequence! Contradiction



- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\leq i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_i$ is not contained
- A must be inside $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i]$ otherwise we break maximality
- $\exists j \leq i-1$ such that A is not inside $\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j] \implies A$ is fully within some cluster C at phase j
- consider all the min. imp. sets flipped after phase j: sequence of λ_{j+1}-improving sets

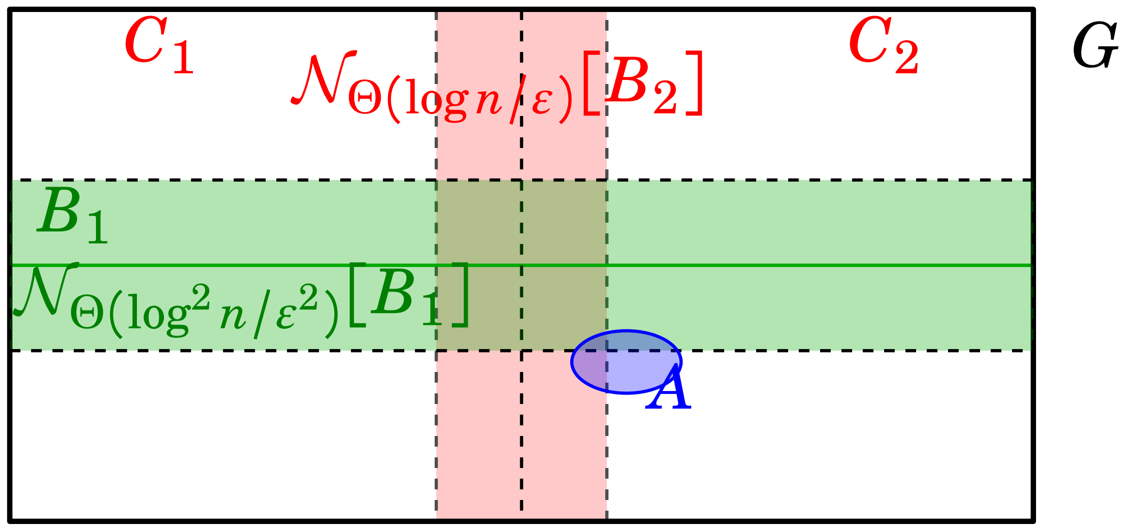
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000\log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR ≥ λ_i of diameter $O(\log n/\varepsilon)$ lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\le i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$

Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR ≥ λ_1 is not contained
- Maximality of the λ_1-improving sequence! Contradiction



- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\le i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$
- By contradiction, suppose min. imp. set A with IR ≥ λ_i is not contained
- A must be inside $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i]$ otherwise we break maximality
- $\exists j \le i-1$ such that A is not inside $\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j] \implies A$ is fully within some cluster C at phase j
- consider all the min. imp. sets flipped after phase j: sequence of λ_{j+1}-improving sets
- Property 2: $\exists$ min. imp. set A' inside $\mathcal{N}_{O(\log^2 n/\varepsilon^2)}[A]$ with IR$(A') \ge \lambda_{j+1} - \varepsilon \ge \lambda_j$

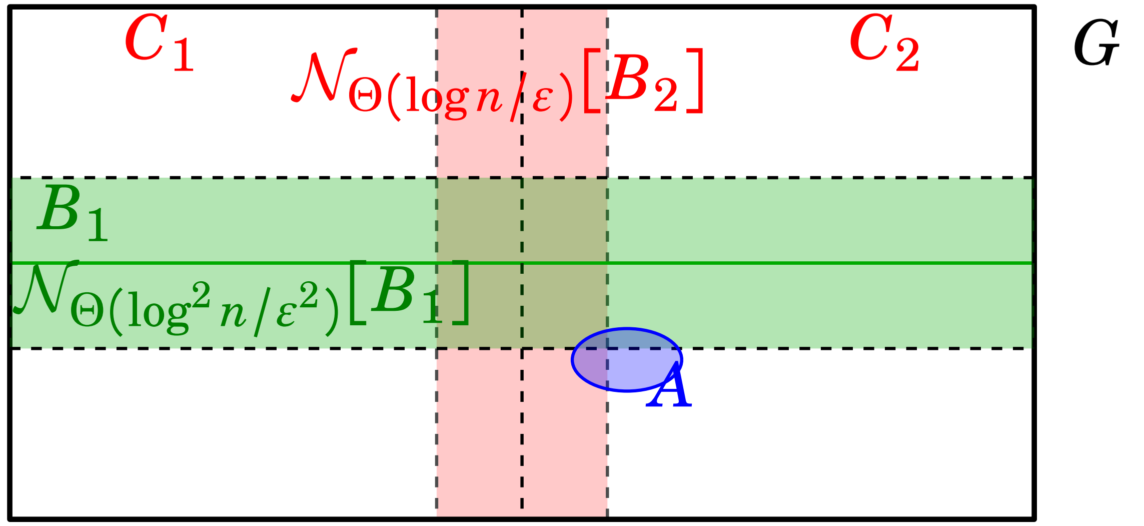
Proof of Claim 2

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

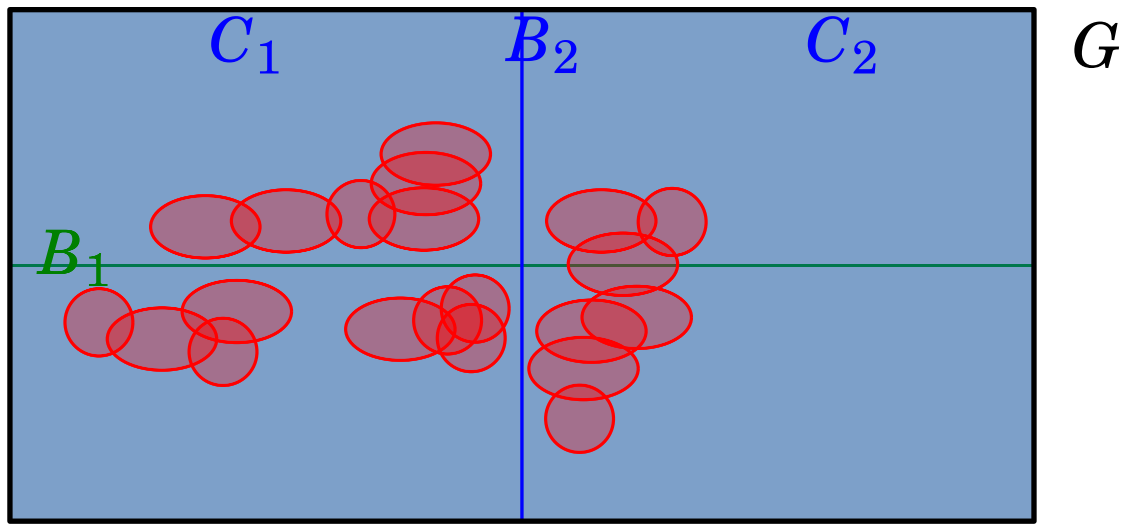
Proof 2: By induction on the phases.

- Base case phase $i = 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_1]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_1$ is not contained
- Maximality of the λ_1-improving sequence! Contradiction



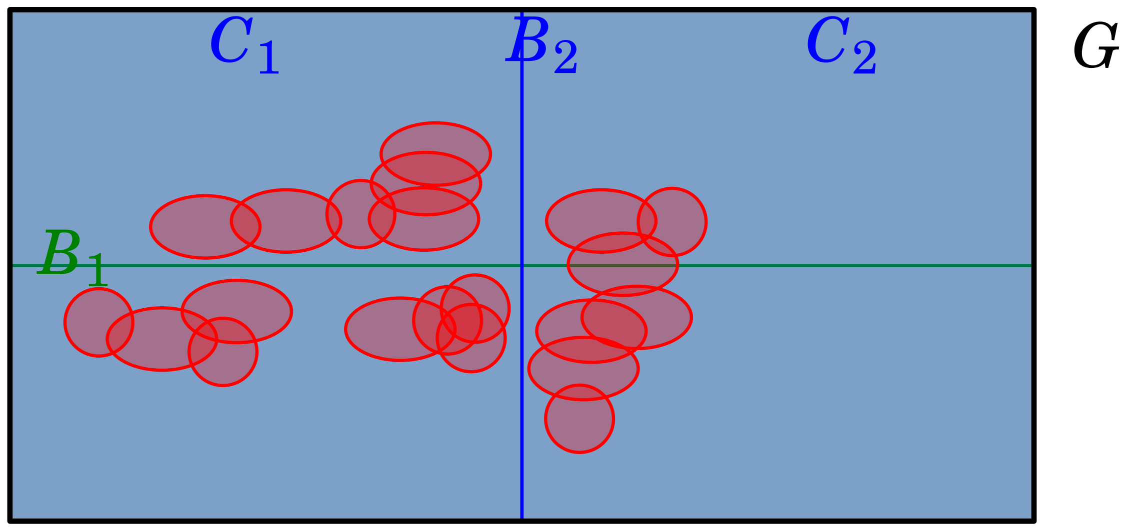
- Suppose $i > 1$, set $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$
- By contradiction, suppose min. imp. set A with IR $\geq \lambda_i$ is not contained
- A must be inside $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i]$ otherwise we break maximality
- $\exists j \leq i-1$ such that A is not inside $\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j] \implies A$ is fully within some cluster C at phase j
- consider all the min. imp. sets flipped after phase j: sequence of λ_{j+1}-improving sets
- Property 2: $\exists$ min. imp. set A' inside $\mathcal{N}_{O(\log^2 n/\varepsilon^2)}[A]$ with IR$(A') \geq \lambda_{j+1} - \varepsilon \geq \lambda_j \implies$ broken maximality in Phase j

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$



- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

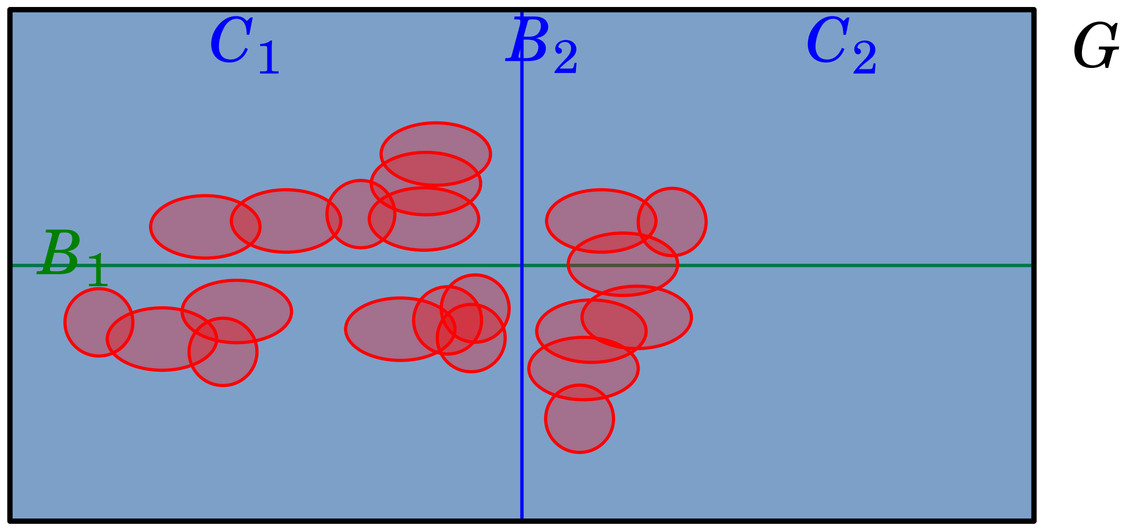
From $\mathcal{MPX}$: *For each* $v \in V$, *with probability* $\geq 1/2$ *there exists a cluster* C *such that* $\mathcal{N}_{\Theta(1/\alpha)}[v] \subseteq C$.



- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

From $\mathcal{MPX}$: *For each* $v \in V$, *with probability* $\geq 1/2$ *there exists a cluster* C *such that* $\mathcal{N}_{\Theta(1/\alpha)}[v] \subseteq C$.

Claim 2: *After phase* i, *any MIS with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$



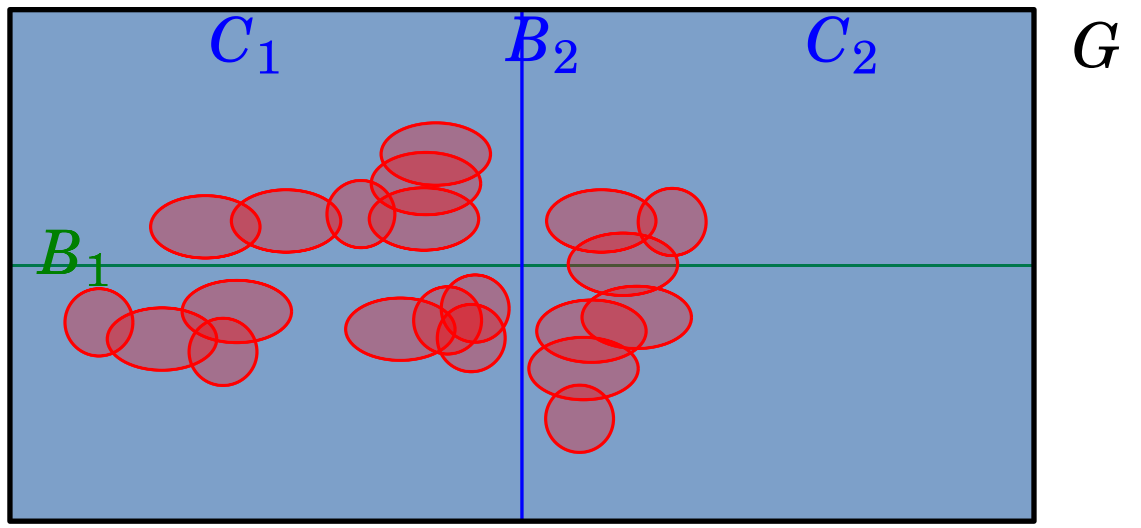
Proof of Claim 3

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000\log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

From $\mathcal{MPX}$: *For each* $v \in V$, *with probability* $\geq 1/2$ *there exists a cluster* C *such that* $\mathcal{N}_{\Theta(1/\alpha)}[v] \subseteq C$.

Claim 2: *After phase* i, *any MIS with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j\leq i-1}\mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$

Claim 3: *With probability* $\geq 1 - 1/n^{10}$, *there is no error after* $100\log n$ *phases*



Proof of Claim 3

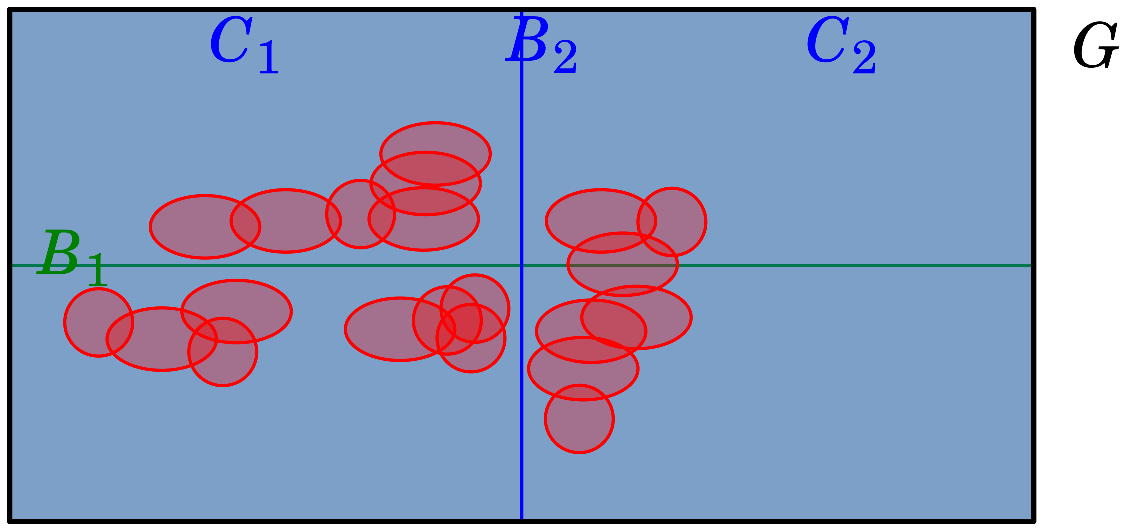
- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

From $\mathcal{MPX}$: *For each* $v \in V$, *with probability* $\geq 1/2$ *there exists a cluster* C *such that* $\mathcal{N}_{\Theta(1/\alpha)}[v] \subseteq C$.

Claim 2: *After phase* i, *any MIS with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$

Claim 3: *With probability* $\geq 1 - 1/n^{10}$, *there is no error after* $100 \log n$ *phases*

Proof 3: By contradiction, there is an error at the end of phase $100 \log n$. Note that $\lambda_{100 \log n} \leq 3/4$.



Proof of Claim 3

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

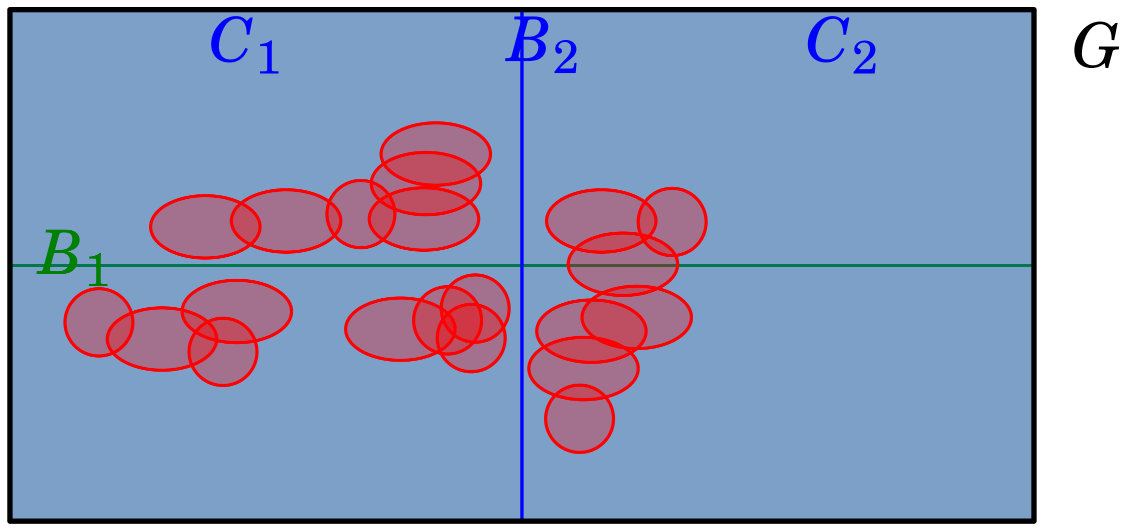
From $\mathcal{MPX}$: *For each $v \in V$, with probability $\geq 1/2$ there exists a cluster C such that $\mathcal{N}_{\Theta(1/\alpha)}[v] \subseteq C$.*

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

Claim 3: *With probability $\geq 1 - 1/n^{10}$, there is no error after $100 \log n$ phases*

Proof 3: By contradiction, there is an error at the end of phase $100 \log n$. Note that $\lambda_{100 \log n} \leq 3/4$.

- Node v colored with green with $> \deg(v)/2$ neighbors of color green



Proof of Claim 3

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

From $\mathcal{MPX}$: *For each* $v \in V$, *with probability* $\geq 1/2$ *there exists a cluster* C *such that* $\mathcal{N}_{\Theta(1/\alpha)}[v] \subseteq C$.

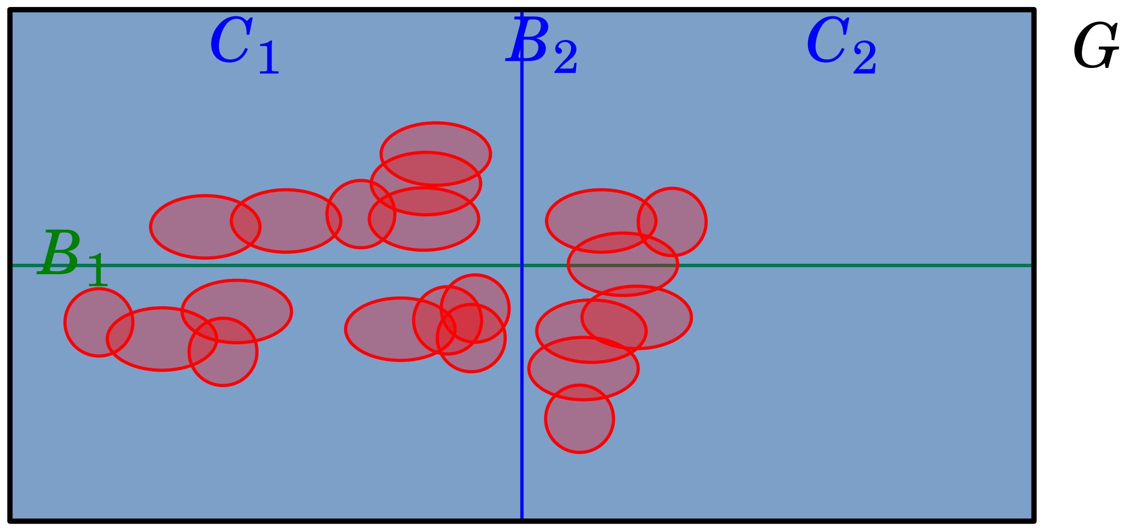
Claim 2: *After phase* i, *any MIS with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap \left[\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]\right]$

Claim 3: *With probability* $\geq 1 - 1/n^{10}$, *there is no error after* $100 \log n$ *phases*

Proof 3: By contradiction, there is an error at the end of phase $100 \log n$. Note that $\lambda_{100 \log n} \leq 3/4$.

- Node v colored with green with $> \deg(v)/2$ neighbors of color green

- $\{v\}$ is an improving set, and IR$(\{v\}) \geq 1$



Proof of Claim 3

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000 \log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

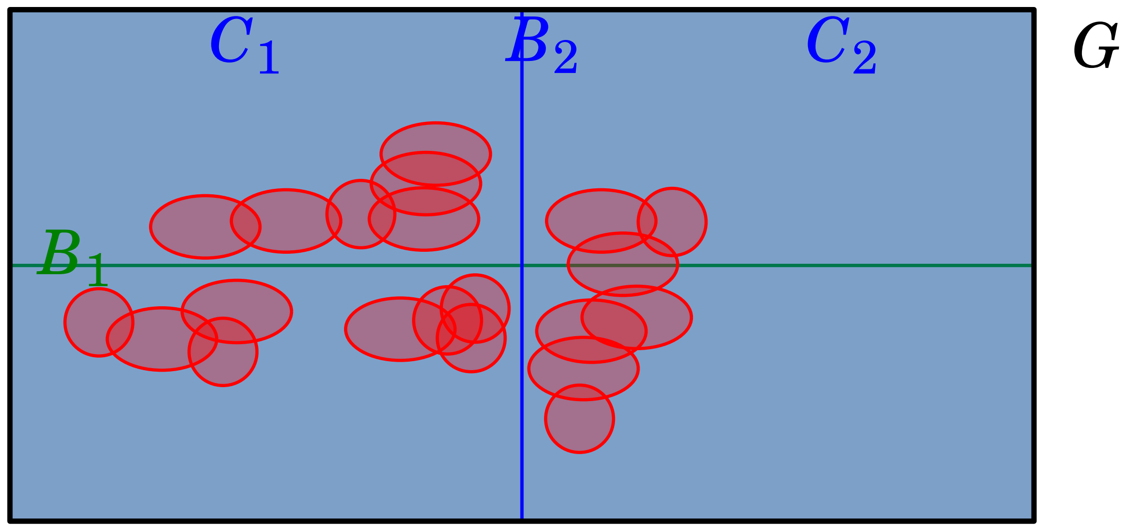
From $\mathcal{MPX}$: *For each $v \in V$, with probability $\geq 1/2$ there exists a cluster C such that $\mathcal{N}_{\Theta(1/\alpha)}[v] \subseteq C$.*

Claim 2: *After phase i, any MIS with IR $\geq \lambda_i$ of diameter $O(\log n/\varepsilon)$ lies in $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap [\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]]$*

Claim 3: *With probability $\geq 1 - 1/n^{10}$, there is no error after $100 \log n$ phases*

Proof 3: By contradiction, there is an error at the end of phase $100 \log n$. Note that $\lambda_{100 \log n} \leq 3/4$.

- Node v colored with green with $> \deg(v)/2$ neighbors of color green
- $\{v\}$ is an improving set, and IR$(\{v\}) \geq 1$
- By $\mathcal{MPX}$ guarantees, $\exists i \leq 100 \log n$ such that $\mathcal{N}_{\Theta(1/\alpha)}[v]$ is contained in some cluster in phase i w.p. $1 - 1/2^{100 \log n - 1} \geq 1 - 1/n^{99}$



Proof of Claim 3

- $\lambda_1 = 1/4$, $\varepsilon = \lambda/(2000\log n)$, $\alpha = \Theta(\varepsilon^2/\log^2 n)$

From $\mathcal{MPX}$: *For each* $v \in V$*, with probability* $\geq 1/2$ *there exists a cluster* C *such that* $\mathcal{N}_{\Theta(1/\alpha)}[v] \subseteq C$.

Claim 2: *After phase* i*, any MIS with IR* $\geq \lambda_i$ *of diameter* $O(\log n/\varepsilon)$ *lies in* $\mathcal{N}_{\Theta(\log n/\varepsilon)}[B_i] \cap \left[\cap_{j \leq i-1} \mathcal{N}_{\Theta(\log^2 n/\varepsilon^2)}[B_j]\right]$

Claim 3: *With probability* $\geq 1 - 1/n^{10}$*, there is no error after* $100\log n$ *phases*

Proof 3: By contradiction, there is an error at the end of phase $100\log n$. Note that $\lambda_{100\log n} \leq 3/4$.

- Node v colored with green with $> \deg(v)/2$ neighbors of color green
- $\{v\}$ is an improving set, and IR$(\{v\}) \geq 1$
- By $\mathcal{MPX}$ guarantees, $\exists i \leq 100\log n$ such that $\mathcal{N}_{\Theta(1/\alpha)}[v]$ is contained in some cluster in phase i w.p. $1 - 1/2^{100\log n - 1} \geq 1 - 1/n^{99}$
- $\alpha = \Theta(\varepsilon^2/\log^2 n)$ is chosen large enough so that Claim 2 is contradicted