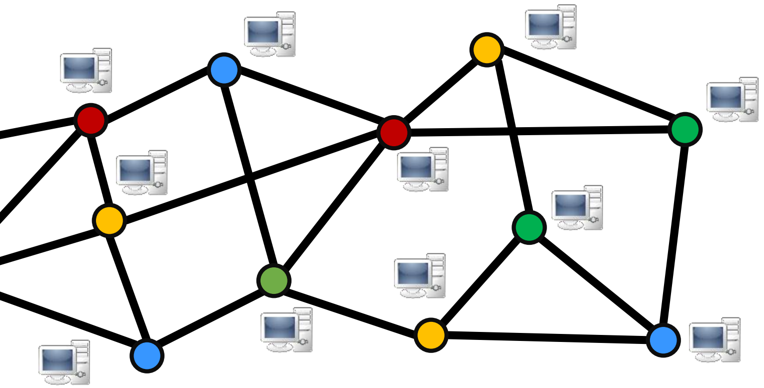
Coloring

Distributed Algorithms **Below** the **Greedy Regime**

Yannic Maus

TECHNION
Israel Institute
of Technology

Mohsen Ghaffari , Juho Hirvonen, Fabian Kuhn, Jara Uitto

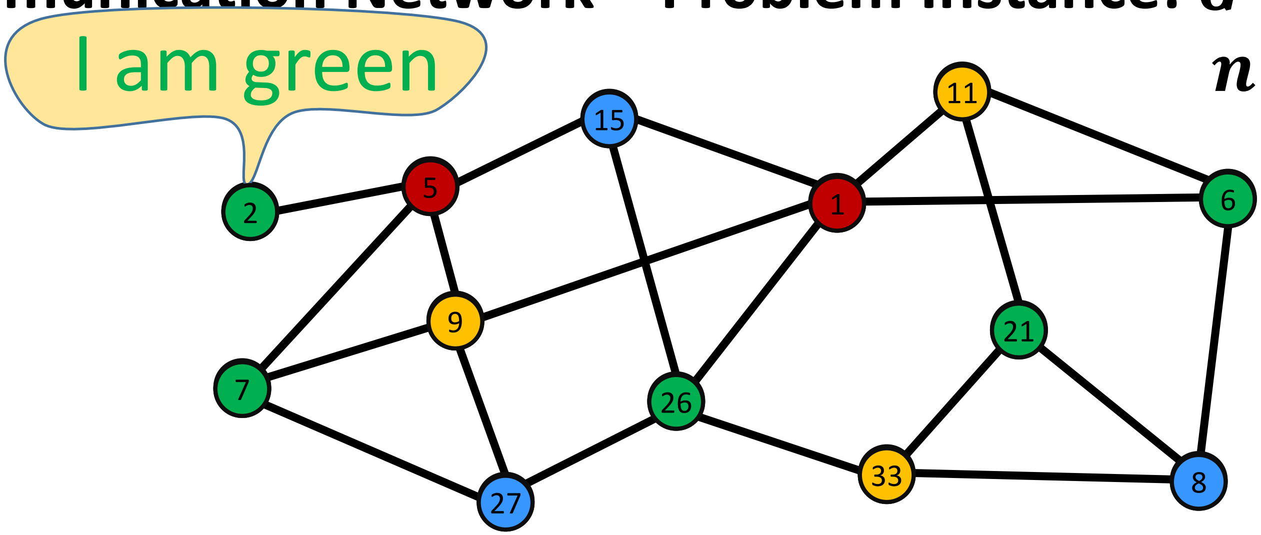


This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 755839.

LOCAL Model [Linial; FOCS ’87]

Communication Network = Problem Instance: $G = (V, E)$,

$n = |V|$



Discrete synchronous rounds:

- local computations
- exchange messages with <u>all</u> neighbors

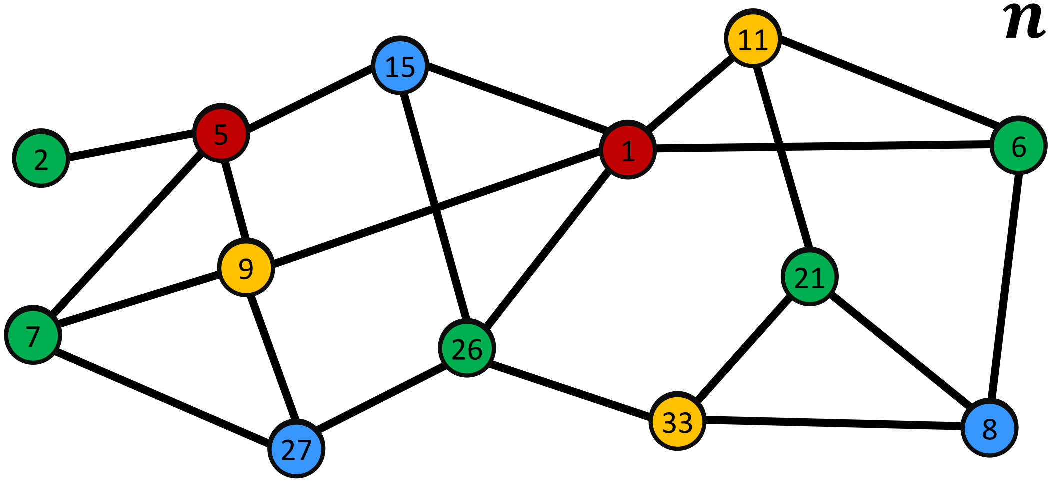
(computations <u>unbounded</u>, message sizes are <u>unbounded</u>)

---

time complexity = number of rounds

CONGEST MODEL

Communication Network = Problem Instance: $G = (V, E)$,
$n = |V|$



Discrete synchronous rounds:

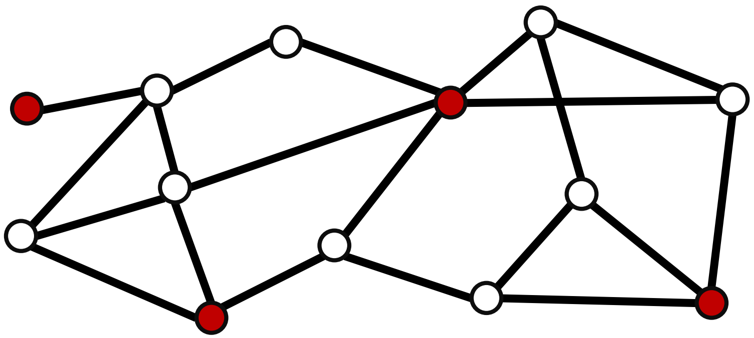
- local computations
- exchange messages with all neighbors

(computations unbounded, message sizes are $O(\log n)$ bits)

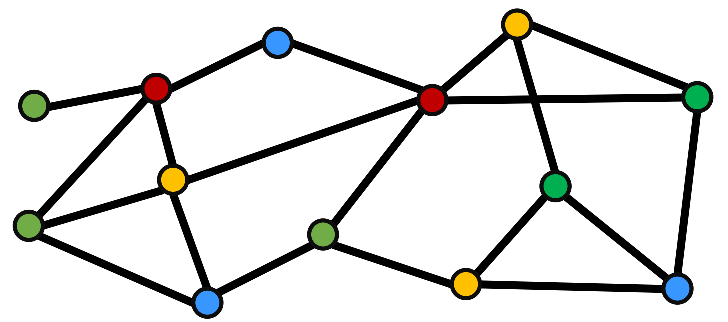
time complexity = number of rounds

Classic Big Four (Greedy Regime)

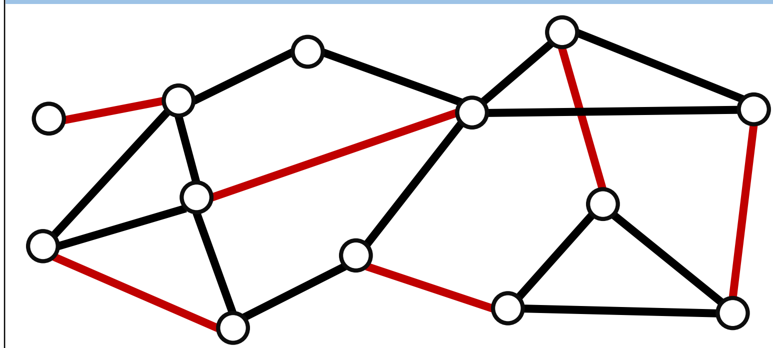
Maximal Ind. Set (MIS)



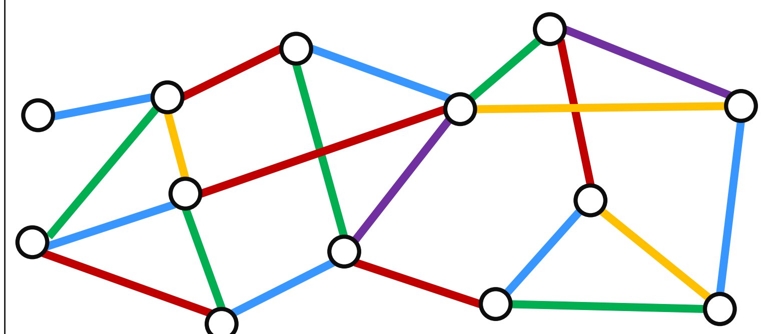
$(\Delta + 1)$-Vertex Coloring



Maximal Matching



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



(Δ: maximum degree of G)

In the LOCAL Model …

Greedy		Below Greedy	
Maximal IS	$2^{O(\sqrt{\log n})}$	$2^{O(\sqrt{\log n})}$	Maximum IS, $(1-\epsilon)$-approx.
vertex cover, **2**-approx.	$\text{poly}\log n$	$2^{O(\sqrt{\log n})}$	vertex cover, $(1+\epsilon)$-approx.
min. dominating set, $(1+\epsilon)\log\Delta$-approx.	$2^{O(\sqrt{\log n})}$	$2^{O(\sqrt{\log n})}$	min dominating set, $(1+\epsilon)$-approx.
hypergraph vertex cover, **rank**-approx.	$2^{O(\sqrt{\log n})}$	$2^{O(\sqrt{\log n})}$	hypergraph vertex cover, $(1+\epsilon)$-approx.

$(\Delta+1)$-vertex coloring	$2^{O(\sqrt{\log n})}$	$2^{O(\sqrt{\log n})}$	Δ-vertex coloring

$(2\Delta-1)$-edge coloring	$\text{poly}\log n$	$\text{poly}\log n$	$(1+\epsilon)\Delta$-edge coloring
maximal matching	$\text{poly}\log n$	$\text{poly}\log n$	Maximum Matching, $(1+\epsilon)$-approx.

CONGEST ✔

In the LOCAL Model …

Greedy	Below Greedy
*"Problems that do have **easy sequential greedy algorithms.**"*	*"**Problems** that do **not** have **easy sequential greedy algorithms.**"*

CONGEST ✔ (Greedy)

Outline

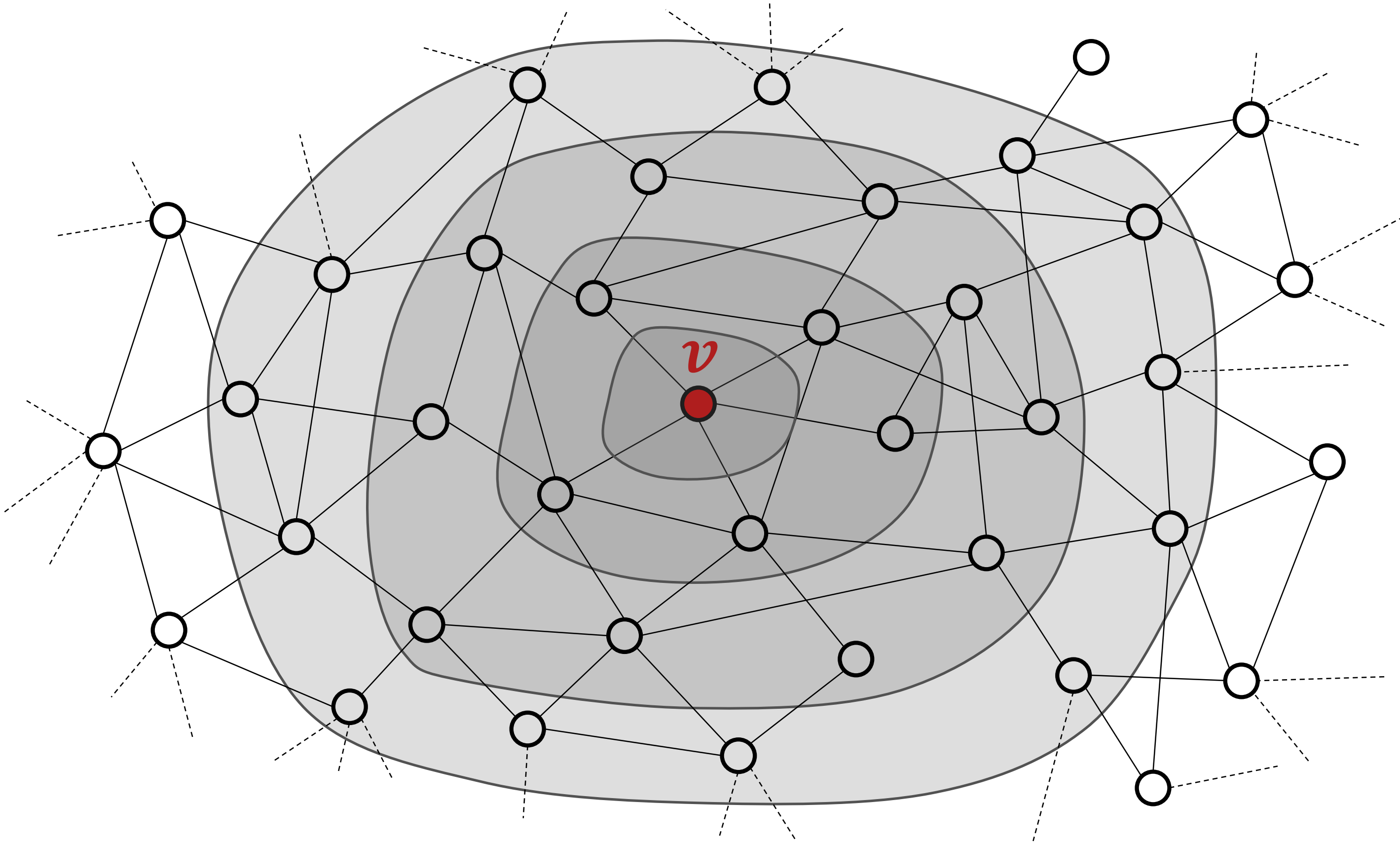
This Talk: How do we use LOCAL?	Below Greedy	
Technique 1: Ball growing [Ghaffari, Kuhn, **Maus; STOC ’17**]	$2^{O(\sqrt{\log n})}$	Maximum IS, $(\mathbf{1}-\boldsymbol{\epsilon})$-approx.
	$2^{O(\sqrt{\log n})}$	vertex cover, $(\mathbf{1}+\boldsymbol{\epsilon})$-approx.
	$2^{O(\sqrt{\log n})}$	min dominating set, $(\mathbf{1}+\boldsymbol{\epsilon})$-approx.
	$2^{O(\sqrt{\log n})}$	hypergraph vertex cover, $(\mathbf{1}+\boldsymbol{\epsilon})$-approx.
Technique 2: Local filling [Ghaffari, Hirvonen, Kuhn, **Maus; PODC ’18**]	$2^{O(\sqrt{\log n})}$	$\boldsymbol{\Delta}$-vertex coloring
Technique 3: Aug. paths [Ghaffari, Kuhn, **Maus**, Uitto; **STOC ’18**]	$\text{poly} \log n$	$(\mathbf{1}+\boldsymbol{\epsilon})\boldsymbol{\Delta}$-edge coloring
	$\text{poly} \log n$	Maximum Matching, $(\mathbf{1}-\boldsymbol{\epsilon})$-approx.

CONGEST

Technique 1: Ball Growing

This Talk: How do we use LOCAL?	Below Greedy	
Technique 1: Ball growing	$2^{O(\sqrt{\log n})}$	Maximum IS, $(\mathbf{1}-\boldsymbol{\epsilon})$-approx.
	$2^{O(\sqrt{\log n})}$	vertex cover, $(\mathbf{1}+\boldsymbol{\epsilon})$-approx.
	$2^{O(\sqrt{\log n})}$	min dominating set, $(\mathbf{1}+\boldsymbol{\epsilon})$-approx.
	$2^{O(\sqrt{\log n})}$	hypergraph vertex cover, $(\mathbf{1}+\boldsymbol{\epsilon})$-approx.
Technique 2: Local filling	$2^{O(\sqrt{\log n})}$	$\boldsymbol{\Delta}$-vertex coloring
Technique 3: Aug. paths	$\text{poly} \log n$	$(\mathbf{1}+\boldsymbol{\epsilon})\boldsymbol{\Delta}$-edge coloring
	$\text{poly} \log n$	Maximum Matching, $(\mathbf{1}-\boldsymbol{\epsilon})$-approx.

Sequential Ball Growing (MaxIS)

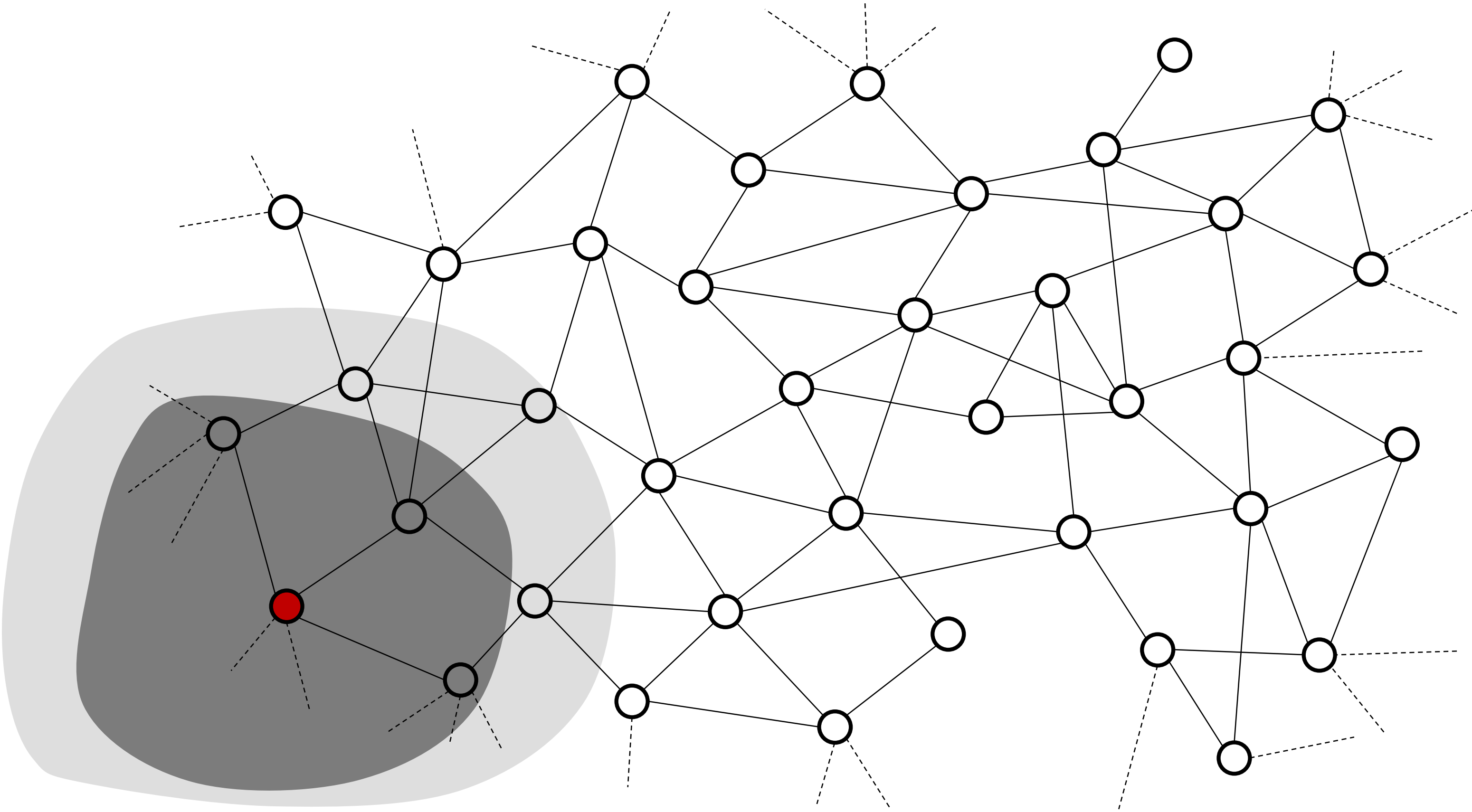


Safe Ball $\boldsymbol{B_r(v)}$: $|MaxIS(B_{r+1})| < (1+\epsilon) \cdot |MaxIS(B_r)|$

Find safe ball: Set $r = 0$ and increase r until ball B_r is safe.

Terminates with small radius $r = O(\epsilon^{-1} \log n)$.

Sequential Ball Growing (MaxIS)

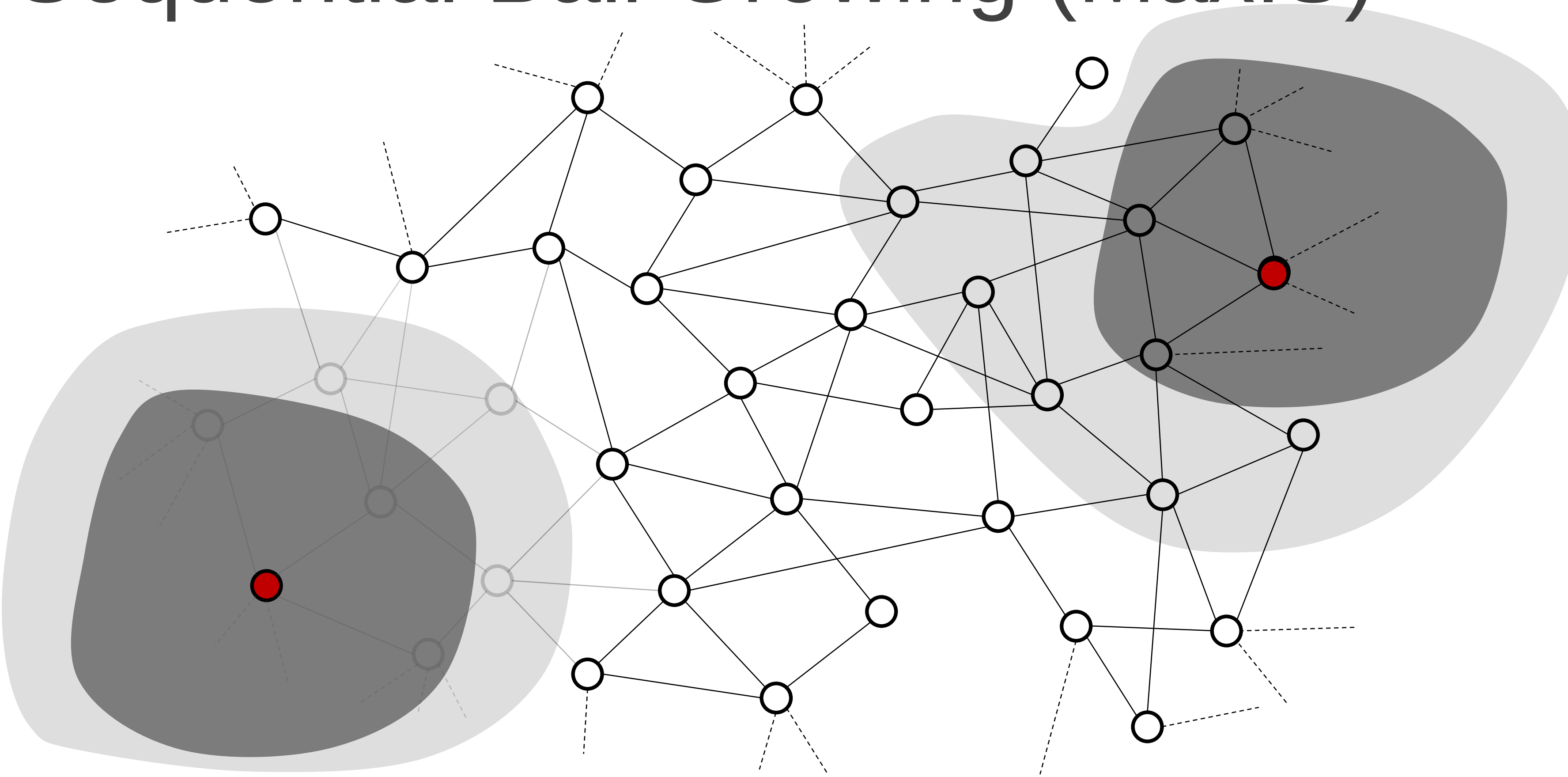


Safe Ball $\boldsymbol{B_r(v)}$: $|MaxIS(B_{r+1})| < (1+\epsilon) \cdot |MaxIS(B_r)|$

Find safe ball: Set $r = 0$ and increase r until ball B_r is safe.

Terminates with small radius $r = O(\epsilon^{-1} \log n)$.

Sequential Ball Growing (MaxIS)

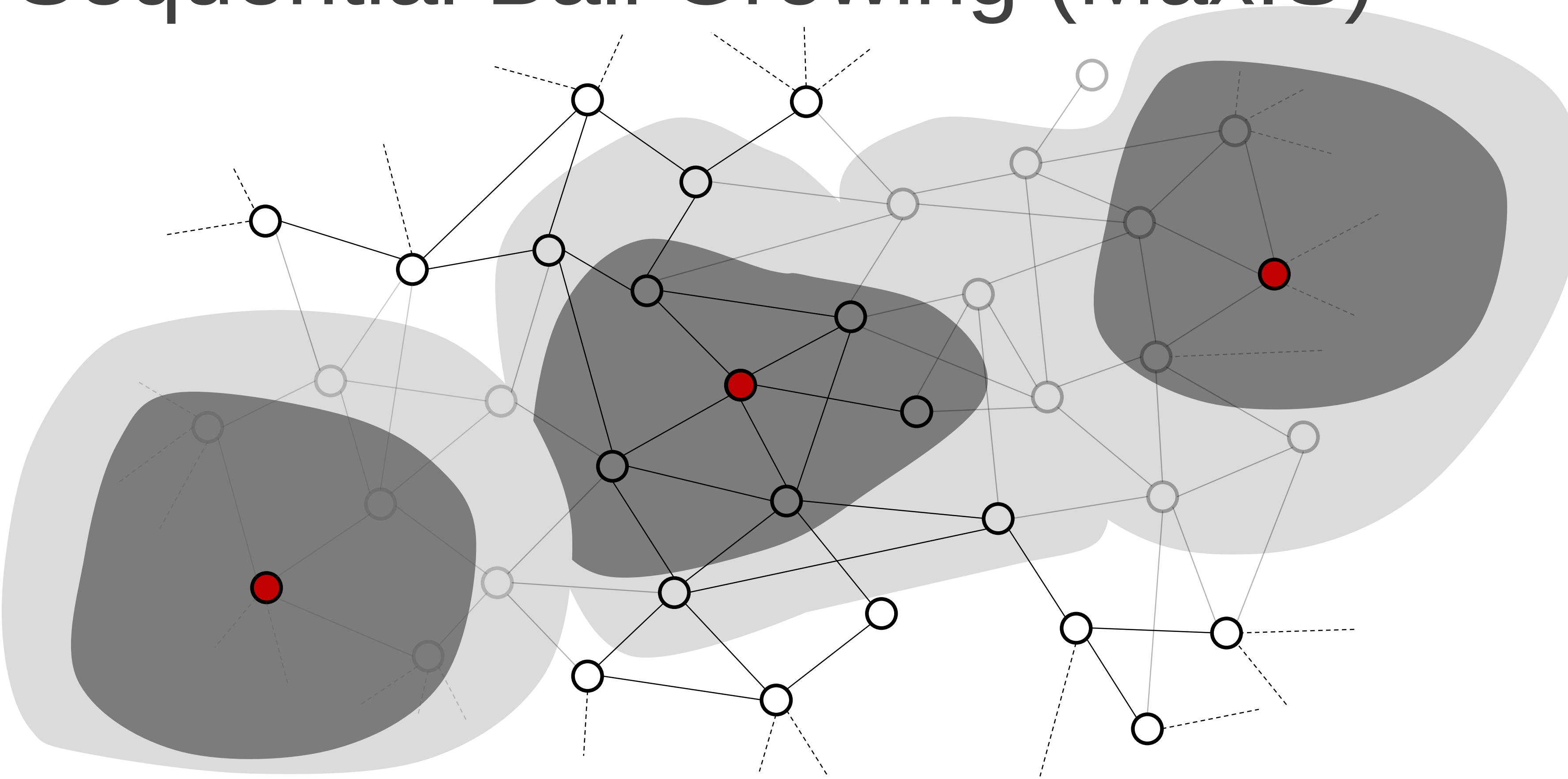


Safe Ball $\boldsymbol{B_r(v)}$: $|MaxIS(B_{r+1})| < (1+\epsilon) \cdot |MaxIS(B_r)|$

Find safe ball: Set $r = 0$ and increase r until ball B_r is safe.

Terminates with small radius $r = O(\epsilon^{-1} \log n)$.

Sequential Ball Growing (MaxIS)

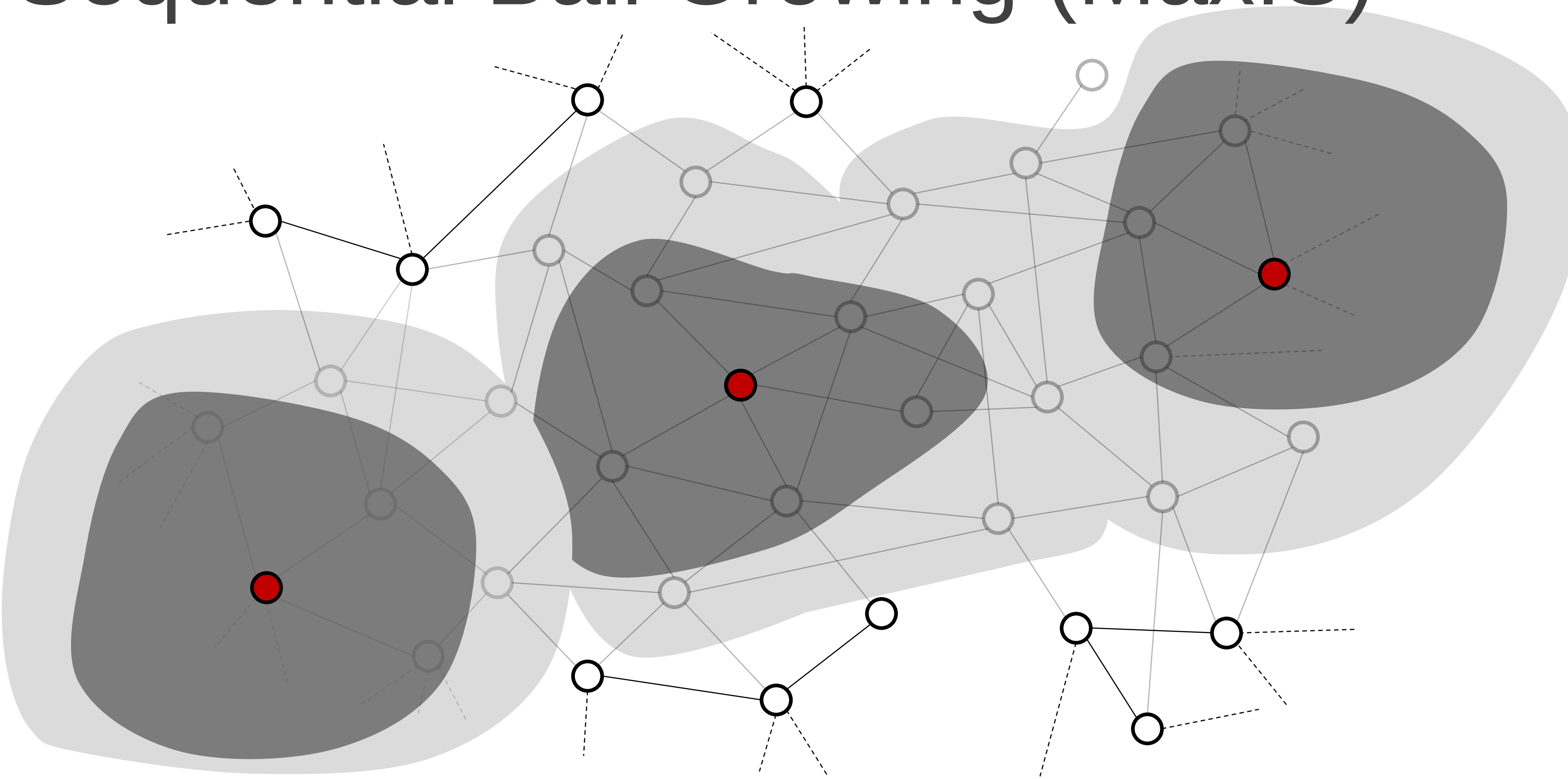


Safe Ball $\boldsymbol{B_r(v)}$: $|MaxIS(B_{r+1})| < (1+\epsilon) \cdot |MaxIS(B_r)|$

Find safe ball: Set $r = 0$ and increase r until ball B_r is safe.

Terminates with small radius $r = O(\epsilon^{-1} \log n)$.

Sequential Ball Growing (MaxIS)

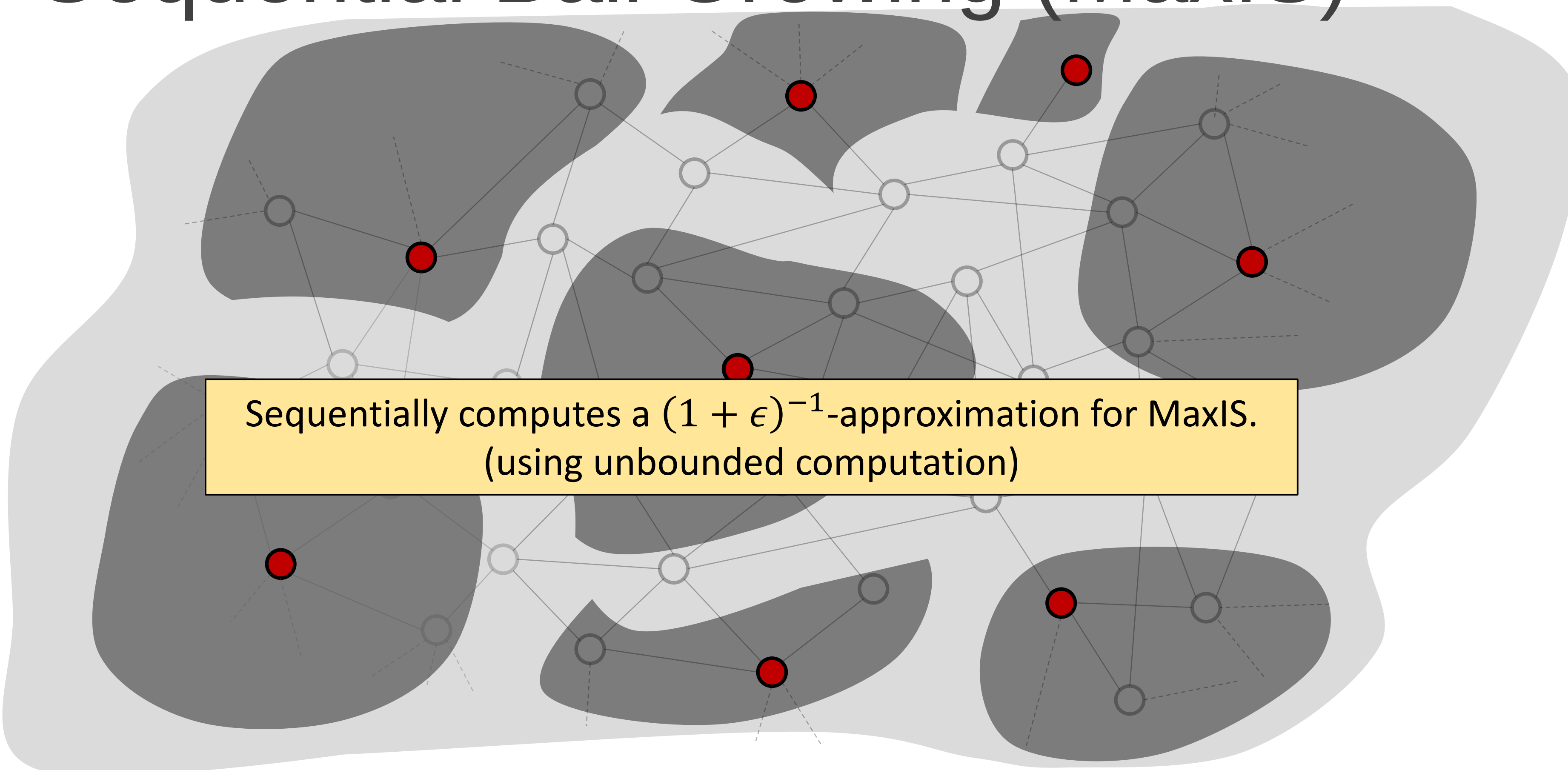


Safe Ball $\boldsymbol{B_r(v)}$: $|MaxIS(B_{r+1})| < (1+\epsilon) \cdot |MaxIS(B_r)|$

Find safe ball: Set $r = 0$ and increase r until ball B_r is safe.

Terminates with small radius $r = O(\epsilon^{-1} \log n)$.

Sequential Ball Growing (MaxIS)



Sequentially computes a $(1+\epsilon)^{-1}$-approximation for MaxIS.
(using unbounded computation)

Safe Ball $\boldsymbol{B_r(v)}$: $|MaxIS(B_{r+1})| < (1+\epsilon)\cdot|MaxIS(B_r)|$

Find safe ball: Set $r = 0$ and increase r until ball B_r is safe.

Terminates with small radius $r = O(\epsilon^{-1}\log n)$.

Parallel Ball Growing

Theorem
Using $(\text{poly} \log n, \text{poly} \log n)$-network decompositions "sequentially ball growing" can be **"done in parallel"** in LOCAL.

[**STOC '17,** Ghaffari, Kuhn, **Maus**]

Corollary
There are $\text{poly} \log n$ randomized and $2^{O(\sqrt{\log n})}$ deterministic $(1+\epsilon)$-approximation algorithms for ***covering and packing integer linear programs***.

This includes maximum independent set, minimum dominating set, vertex cover, … .

[**STOC '17,** Ghaffari, Kuhn, **Maus**]

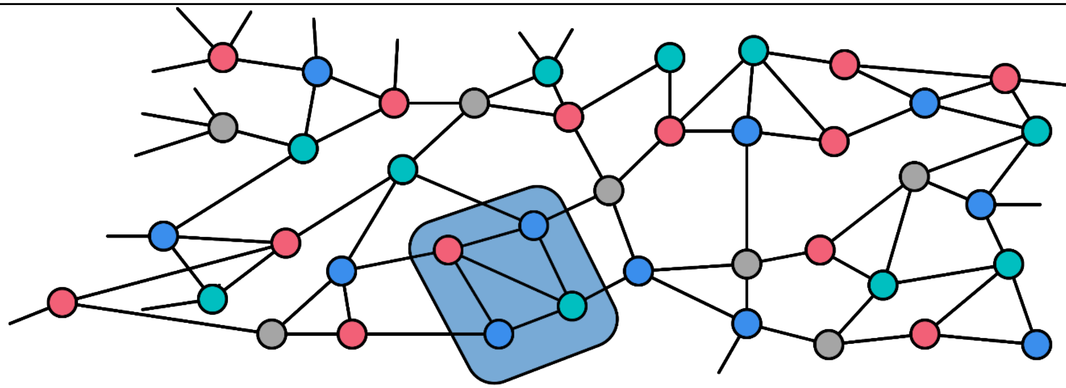
Technique 2: Local Filling

This Talk: How do we use LOCAL?	Below Greedy	
Technique 1: Ball growing	$2^{O(\sqrt{\log n})}$	Maximum IS, $(1-\epsilon)$-approx.
	$2^{O(\sqrt{\log n})}$	vertex cover, $(1+\epsilon)$-approx.
	$2^{O(\sqrt{\log n})}$	min dominating set, $(1+\epsilon)$-approx.
	$2^{O(\sqrt{\log n})}$	hypergraph vertex cover, $(1+\epsilon)$-approx.
Technique 2: Local filling	$2^{O(\sqrt{\log n})}$	Δ-vertex coloring
Technique 3: Aug. paths	$\text{poly} \log n$	$(1+\epsilon)\Delta$-edge coloring
	$\text{poly} \log n$	Maximum Matching, $(1-\epsilon)$-approx.

Δ-Coloring

Previous Work: [Panconesi, Srinivasan; STOC ’93]

Definition: An induced subgraph $H \subseteq G$ is called an ***easy component*** if any Δ_G*-coloring of* $G \setminus H$ can be extended to a Δ_G*-coloring of* G *without changing the coloring on* $G \setminus H$.



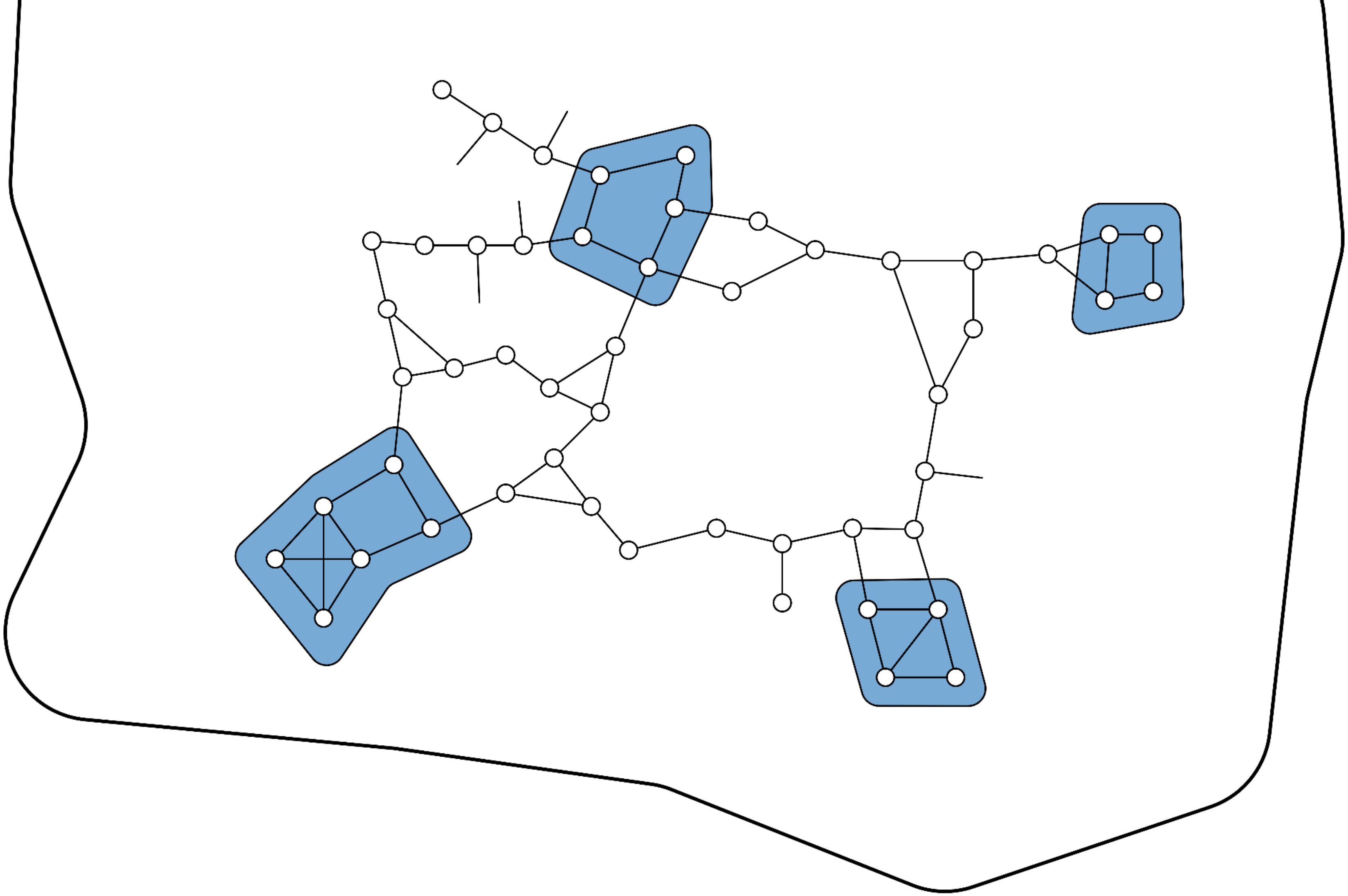
Well studied under the name ***degree chosable components.***

[Erdős et al. ’79, Vizing ‘76]

“ **Theorem:** Let G be a graph (≠clique) with max. degree $\Delta \geq 3$. Every node of G has a **small diameter easy component** in **distance at most O(log n)**. “

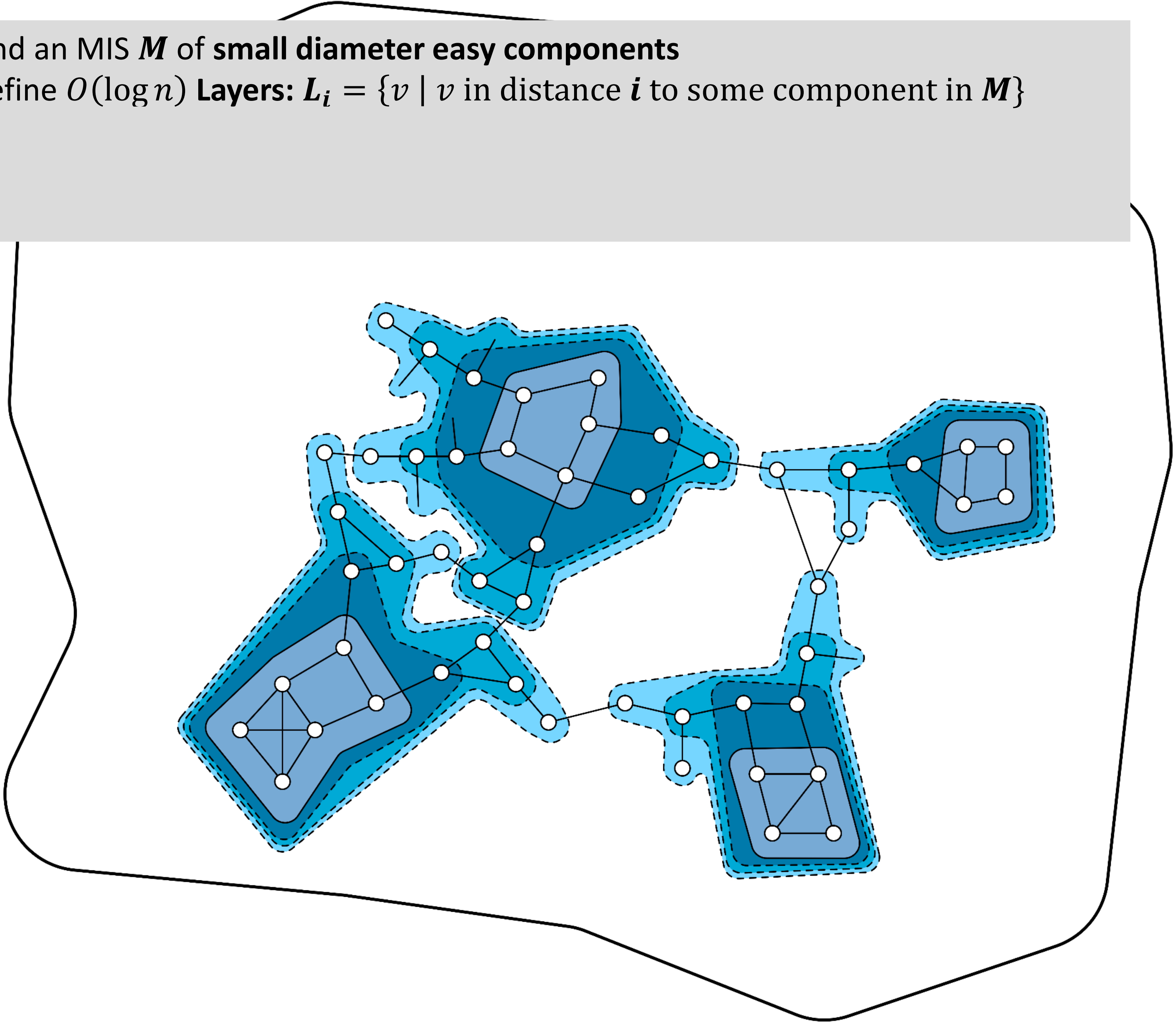
[**PODC ’18;** Ghaffari, Hirvonen, Kuhn, **Maus**]

Find an MIS $\boldsymbol{M}$ of **small diameter easy components**



Find an MIS $\boldsymbol{M}$ of **small diameter easy components**

Define $O(\log n)$ **Layers:** $\boldsymbol{L_i} = \{v \mid v \text{ in distance } \boldsymbol{i} \text{ to some component in } \boldsymbol{M}\}$



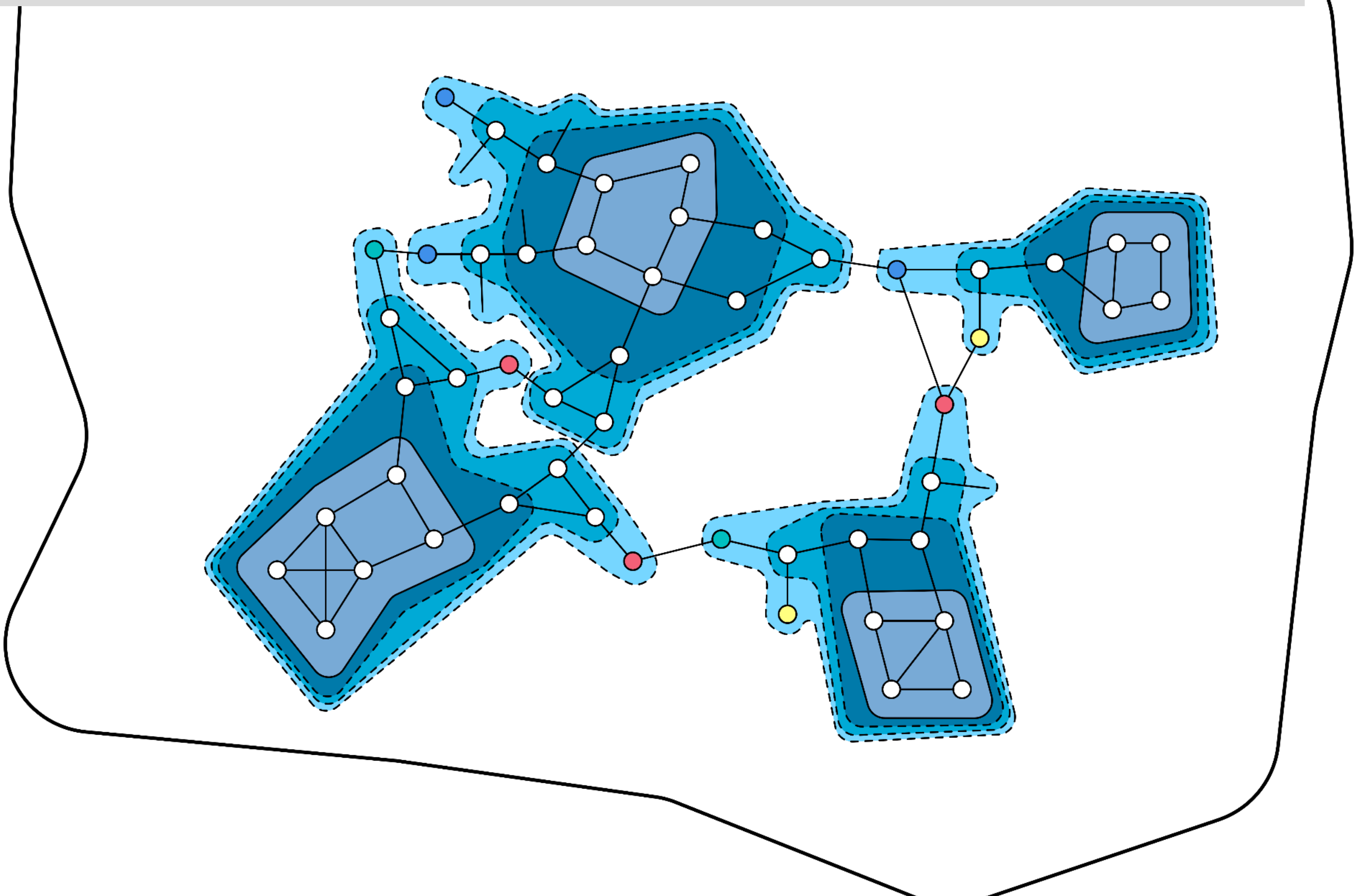
Find an MIS $\boldsymbol{M}$ of **small diameter easy components**

Define $O(\log n)$ **Layers:** $\boldsymbol{L_i} = \{v \mid v \text{ in distance } \boldsymbol{i} \text{ to}$ … $\}$

For $i = O(\log n)$ to 1

color nodes in L_i through solving a **(deg+1)-list coloring**

Greedy regime



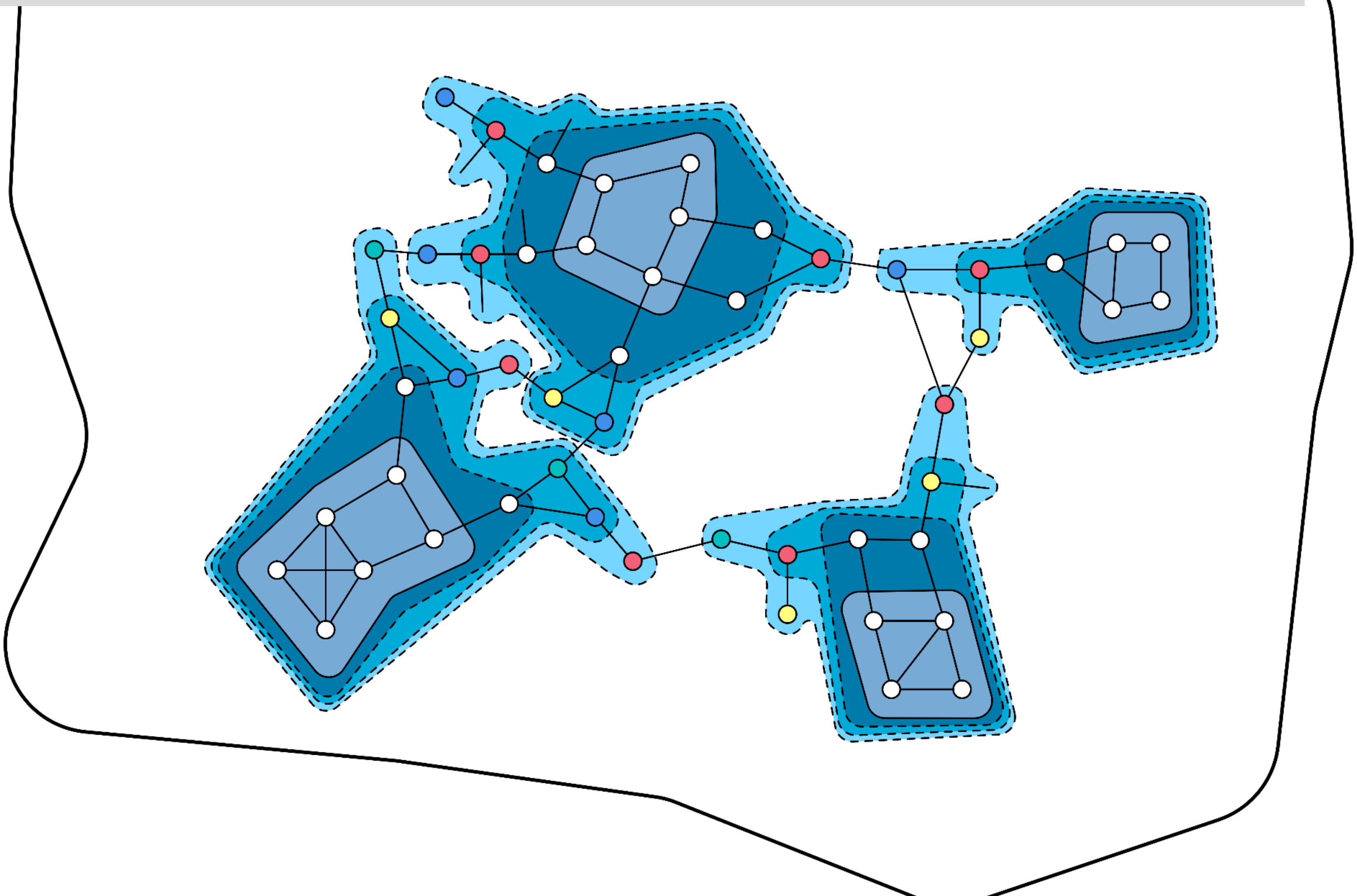
Find an MIS $\boldsymbol{M}$ of **small diameter easy components**

Define $O(\log n)$ **Layers:** $\boldsymbol{L_i} = \{v \mid v \text{ in distance } \boldsymbol{i} \text{ to}$ … $\}$

For $i = O(\log n)$ to 1

color nodes in L_i through solving a **(deg+1)-list coloring**

Greedy regime



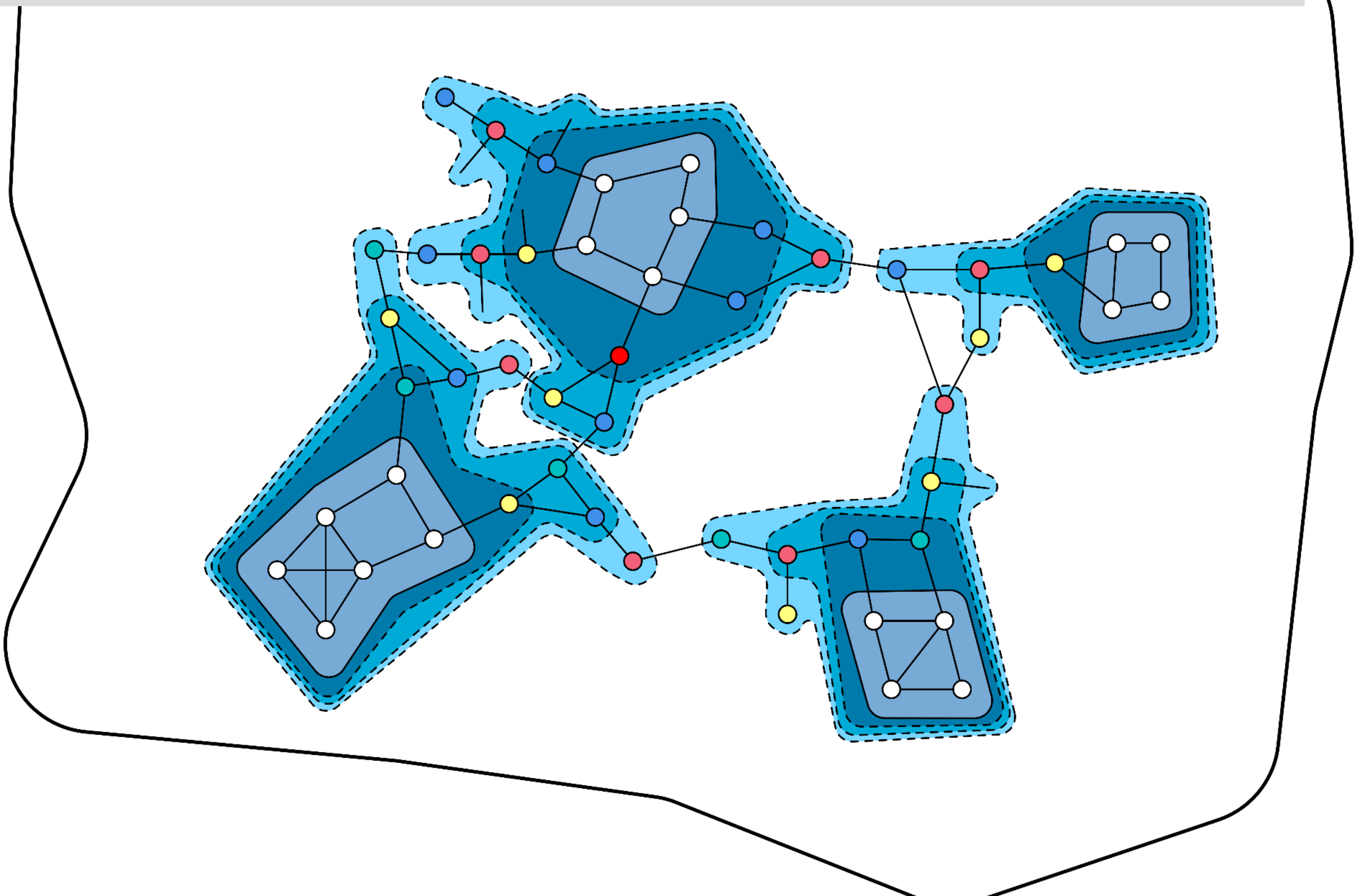
Find an MIS $\boldsymbol{M}$ of **small diameter easy components**

Define $O(\log n)$ **Layers:** $\boldsymbol{L_i} = \{v \mid v \text{ in distance } \boldsymbol{i} \text{ to}$ … $\}$

For $i = O(\log n)$ to 1

color nodes in L_i through solving a **<u>(deg+1)-list coloring</u>**

Greedy regime



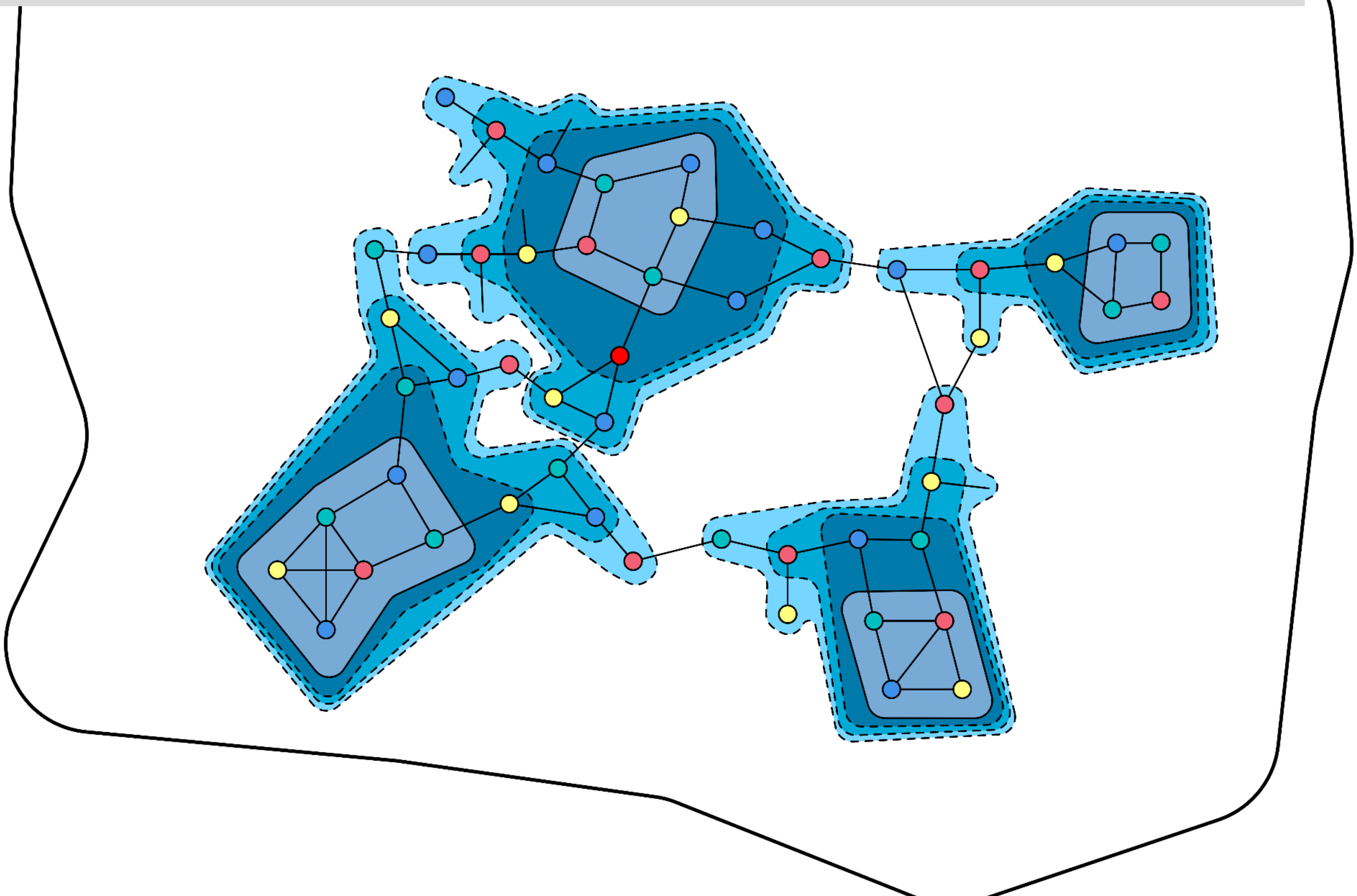
Find an MIS $\boldsymbol{M}$ of **small diameter easy components**

Define $O(\log n)$ **Layers:** $\boldsymbol{L_i} = \{v \mid v \text{ in distance } \boldsymbol{i} \text{ to}$ … $\}$

For $i = O(\log n)$ to 1

color nodes in L_i through solving a **(deg+1)-list coloring**

Color easy components in M



Technique 3: Augmenting Paths

This Talk: How do we use LOCAL?	Below Greedy	
Technique 1: Ball growing	$2^{O(\sqrt{\log n})}$	Maximum IS, $(\mathbf{1-\epsilon})$-approx.
	$2^{O(\sqrt{\log n})}$	vertex cover, $(\mathbf{1+\epsilon})$-approx.
	$2^{O(\sqrt{\log n})}$	min dominating set, $(\mathbf{1+\epsilon})$-approx.
	$2^{O(\sqrt{\log n})}$	hypergraph vertex cover, $(\mathbf{1+\epsilon})$-approx.
Technique 2: Local filling	$2^{O(\sqrt{\log n})}$	$\mathbf{\Delta}$-vertex coloring
Technique 3: Aug. paths	$\text{poly} \log n$	$(\mathbf{1+\epsilon})\mathbf{\Delta}$-edge coloring
	$\text{poly} \log n$	Maximum Matching, $(\mathbf{1-\epsilon})$-approx.

$(1+\epsilon)\Delta$-Edge Coloring

Theorem
$(1+\epsilon)\Delta$-edge coloring can be **efficiently** reduced to the computation of **weighted maximum matching approximations** .

The **reduction** can be executed in the **CONGEST model**.

[Ghaffari, Kuhn, **Maus**, Uitto; **STOC '18**]

For $i = 1$ to $2\Delta - 1$
  compute a maximal *matching* M of G
  color edges of M with color i
  remove M from G
Next

Well known: $2\Delta - 1$ *iterations suffice to color all edges.*

$(1+\epsilon)\Delta$-Edge Coloring

Theorem
$(1+\epsilon)\Delta$-edge coloring can be **efficiently** reduced to the computation of **weighted maximum matching approximations** .

The **reduction** can be executed in the **CONGEST model**.

[Ghaffari, Kuhn, **Maus**, Uitto; **STOC '18**]

For $i = 1$ to $(1+\epsilon)\Delta$
  compute a *good matching* M_{good} of G
  color edges of M_{good} with color i
  remove M_{good} from G
Next

$(1+\epsilon)\Delta$**-edge coloring** through ***good*** matchings:
  Reduce the max degree (amortized) at a rate of $(1-\epsilon)$.

Summary Techniques (LOCAL)

Technique 1: Sequential ball growing

Problems: Approx. for MaxIS, MinDS, MinVC and many more …

How do we (ab)use LOCAL?

*Compute **optimal solutions** in **small diameter** graphs*

Technique 2: Local filling

Problems: Δ-Coloring, ?

How do we (ab)use LOCAL?

"Existence" *+* ***small diameter*** *is enough to obtain a solution*

Technique 3: Augmenting paths

Problems: $(1+\epsilon)\Delta$ Edge Coloring, Maximum Matching Approx.

How do we ~~(ab)~~use LOCAL?

Finding a maximal set of augmenting paths

CONGEST

Lower Bounds in CONGEST

Lower Bounds in CONGEST		**Below Greedy**	
Maximum IS **7/8**-approx.	$\widetilde{\Omega}(n^2)$	$2^{O(\sqrt{\log n})}$	Maximum IS, $(\mathbf{1}-\boldsymbol{\epsilon})$-approx.
vertex cover, **exact**	$\widetilde{\Omega}(n^2)$	$2^{O(\sqrt{\log n})}$	vertex cover, $(\mathbf{1}+\boldsymbol{\epsilon})$-approx.
min. dominating set, **exact**	$\widetilde{\Omega}(n^2)$	$2^{O(\sqrt{\log n})}$	min dominating set, $(\mathbf{1}+\boldsymbol{\epsilon})$-approx.
hypergraph vertex cover, **exact**	$\widetilde{\Omega}(n^2)$	$2^{O(\sqrt{\log n})}$	hypergraph vertex cover, $(\mathbf{1}+\boldsymbol{\epsilon})$-approx.
$\boldsymbol{\chi(G)}$-vertex coloring	$\widetilde{\Omega}(n^2)$	$2^{O(\sqrt{\log n})}$	$\boldsymbol{\Delta}$-vertex coloring
?-edge coloring	?	$\text{poly}\log n$	$(\mathbf{1}+\boldsymbol{\epsilon})\boldsymbol{\Delta}$-edge coloring
Maximum matching **almost exact**	$\Omega(\sqrt{n}$	$\text{poly}\log n$	Maximum Matching, $(\mathbf{1}-\boldsymbol{\epsilon})$-approx.

[BCDELP ‘19], [AKO ’18], [ACK ’16]

CONGEST 

Let’s Discuss …

A lot was spared in this talk (randomized!)

- $(1-\epsilon)$ max cut approximation: [Zelke ‘09]
  Similar to technique 1: Subsampling + solving optimally
- spanners, e.g., [Censor-Hillel, Dory; PODC ‘18]
- randomized edge coloring below the greedy regime, e.g., [Elkin, Pettie, Su; SODA ’15], [Chang, He, Li, Pettie, Uitto; SODA ‘18], [Su, Vu; STOC ‘19]
- MPC: Maximum Matching approx. in time $O(\log\log n)$ [Behnezhad, Hajiaghayi, Harris; FOCS ‘19]
- lots more …

*What **can** or **cannot** be done **below the greedy regime** in distributed models with **limited communication** (CONGEST, CONGESTED CLIQUE, MPC, …)?*

Thank you