Simple Efficient Distributed Processes on Graphs

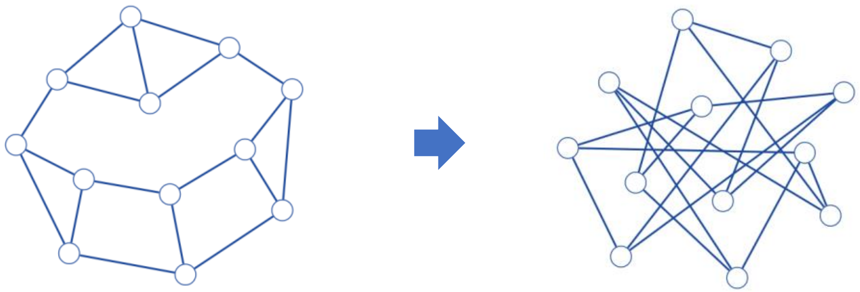
George Giakkoupis

INRIA, France

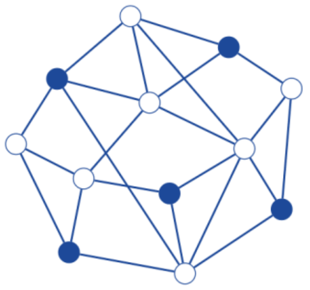
SIROCCO 2022, Paderborn, June 27-29 2022

Talk Overview

- Part 1: Transform any connected regular graph into an expander



- Part 2: Compute a maximal-independent-set of any graph

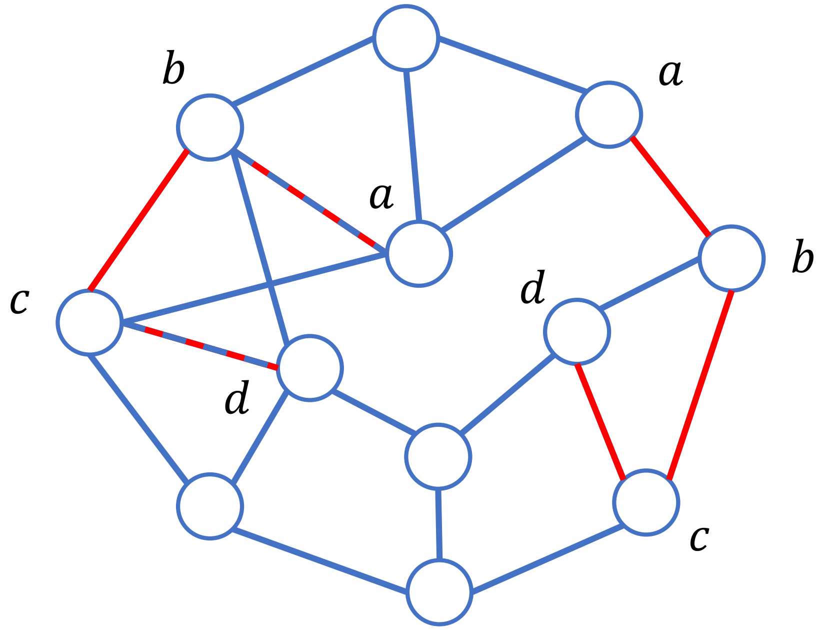


Flip Process

- Start from any connected d-regular graph
- Apply a sequence of flip operations

- Flip operation
  - Pick a random 3-path $abcd$
  - If edges ac and bd do not exist:
    replace ab and cd by ac and bd

- Maintains graph connectivity & degrees



Flip Process

[Mahlmann and Schindelhaue, 2005]

- Converges to uniform distribution over all connected d-regular graphs
- Time until an expander graph is established? / Mixing time?

- Experiments: $O(nd \log n)$ operations to have an expander w.h.p.

Motivation

- Simple local MCMC process for sampling (approximately) random connected d-regular graphs
- Easy to implement in parallel (MapReduce, Hadoop,…)

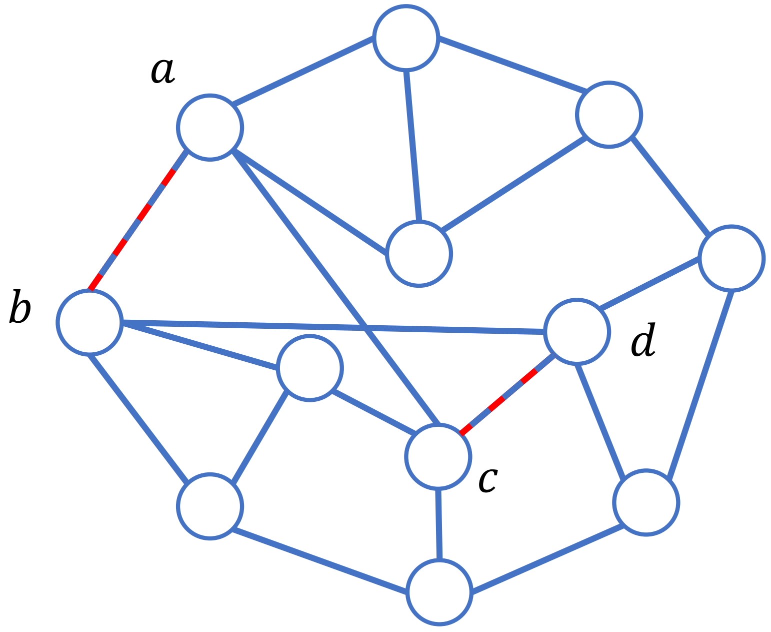
- Simple local process for generating/maintaining a d-regular expander
- Application to design of unstructured overlay (p2p) networks
  - Small diameter, low degree, good connectivity (for robustness)
- Edge flip operations already used in overlay systems in practice

Related: Switch Process

[McKay, 1981]

- Switch operation
  - Pick random non-adjacent edges ab & cd
  - If edges ac and bd do not exist:
    replace ab and cd by ac and bd

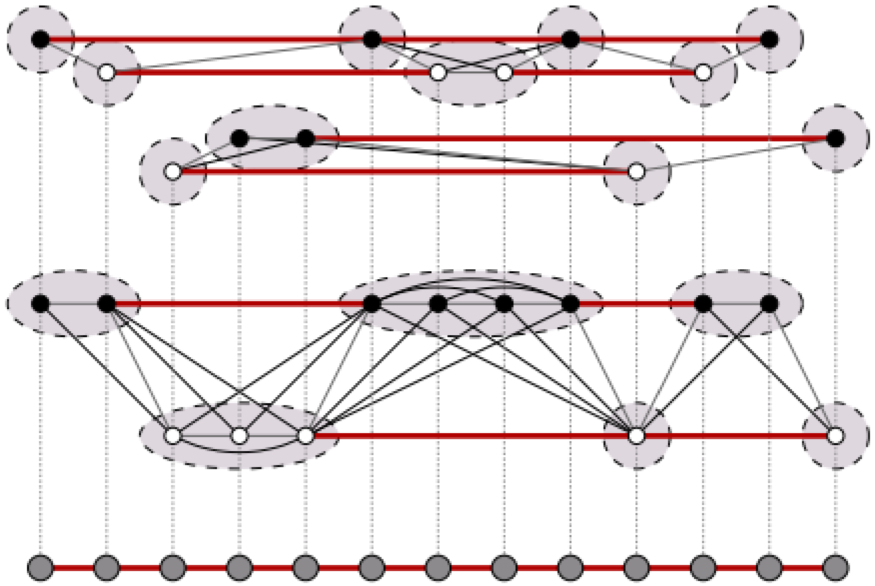
- Converges to a random d-regular graph
- But not local & may disconnect graph



Related: Expanders via “Structured” Overlay Designs

SKIP+ Graph [Jacob, Richa, Scheideler, Schmid and Täubig, 2014]

- Local, self-stabilizing
- Transforms any connected graph, to one containing a spanning constant-degree expander, in $O(\log^2 n)$ synchronous rounds
- But complex (large state/messages)



Known Bounds for Flip and Switch Processes

- For d-regular n-vertex graphs:

<table>
<tr><th colspan="2"></th><th>Mixing time</th><th>Time to expander (w.h.p.)</th></tr>
<tr><td rowspan="2">Switch process</td><td colspan="2">$O(n^9 d^{24} \log n)$
[Cooper, Dyer and Greenhill, 2007+2012]</td><td rowspan="2">$O(nd)$
[Allen-Zhu, Bhaskara, Lattanzi, Mirrokni and Orecchia, 2016]</td></tr>
<tr><td>Bipartite</td><td>$O(n^2 d^2 \log n)$, $O(n \log^2 n)$ if $d = O(1)$
[Tikhomirov and Youssef, 2020],
[Kannan, Tetali and Vempala, 1999]</td></tr>
<tr><td>Flip process</td><td colspan="2">$O(n^{16} d^{36} \log n)$
[Cooper, Dyer, Greenhill and Handley, 2019],
[Feder, Guetz, Mihail, and Saberi, 2006]</td><td>$O\left(n^2 d^2 \sqrt{\log n}\right)$
[Allen-Zhu *et al*, 2016]</td></tr>
</table>

- Techniques: canonical path, Markov Chain comparison, spectral /algebraic
- Also results for non-regular/directed graphs

New Bound for the Flip Process

[Giakkoupis 2022]

For any n and $d = \Omega(\log^2 n)$, there exists $t = O(nd \log^2 n)$, such that, applying t flip operations to any connected d-regular n-vertex graph, results in an expander graph w.h.p.

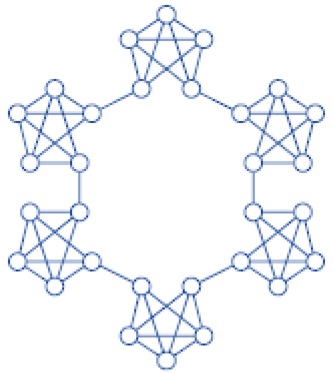
- $O(t/n) = O(d \log^2 n)$ operations per vertex
  - Previous best was $O\left(nd^2\sqrt{\log n}\right)$ [Allen-Zhu *et al*, 2016]
  - Justifies use of flip-like operations in overlay networks
- Comparable to bounds for (non-local) switch process, and (more complex) SKIP+ graph

New Bound for the Flip Process

[Giakkoupis 2022]

For any n and $d = \Omega(\log^2 n)$, there exists $t = O(nd \log^2 n)$, such that, applying t flip operations to any connected d-regular n-vertex graph, results in an expander graph w.h.p.

- Almost tight
  - $\Omega(nd \log(n/d))$ operations for "ring-of-cliques"



- Probably, a refinement of our analysis could improve result to $d = \Omega(\log n)$ and $t = O(nd \log n)$

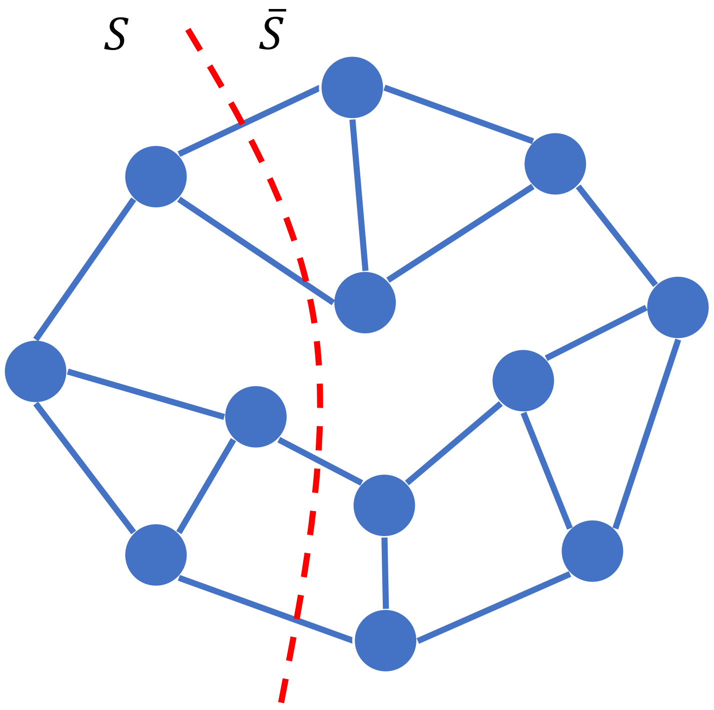
Analysis of Flip Process

Some Standard Definitions

- Cut $(S, \bar{S})$

- Cut size = number of crossing edges
- Graph (edge-)connectivity = min cut size

- Cut (edge-)expansion = cut size / $|S|$
- Graph expansion = min cut expansion



- A d-regular graph is an expander if the expansion is $\Omega(d)$

Proof Overview

Part I: Edge Connectivity Analysis

- Edge connectivity $\geq d/2$ achieved in $O(nd \log^2 n)$ operations, and maintained for $\mathrm{poly}(n)$ operations thereafter
- Requires $d = \Omega(\log^2 n)$

Part II: Expansion Analysis

- Assumes edge connectivity $\geq d/2$ throughout
- Expansion $\Omega(d)$ achieved in $O(nd \log n)$ operations, and maintained for $\mathrm{poly}(n)$ operations thereafter
- Requires $d = \Omega(\log n)$

Edge Connectivity Analysis

- Analyze a single cut $(S, \bar{S})$
  - Analyze cut size $c(S)$
  - $c(S) \geq d/2$ after t ops $\forall t = \Theta(nd \log n) \dots \text{poly}(n)$, w.pr. $1 - n^{-c}$
- Argue about all cuts using "smart" union bounds
  - UB over all S with $\ell < |S| \leq 2\ell$, after establishing the fact $\forall S$ with $|S| \leq \ell$
  - Key Lemma: If $c(S) \geq k$ $\forall S$ with $|S| \leq \ell$, then there are $O(n)$ many sets S with $\ell < |S| \leq 2\ell$ and $c(S) < k$

Expansion Analysis

cut $(S, \bar{S})$ is ℓ-expanding if $c(S) = \Omega(d \min\{\ell, |S|\})$

- Analyze a single cut $(S, \bar{S})$
  - Analyze new measure of cut strain
  - As long as all cuts remain ℓ-expanding, $(S, \bar{S})$ is 2ℓ-expanding after t ops $\forall t = \Theta(nd) \ldots \mathrm{poly}(n)$, w.pr. $1 - e^{-\Omega(\ell d)}$
- Argue about all cuts using "smart" union bounds
  - Show 2ℓ-expansion for all cuts, after establishing ℓ-expansion
  - By Karger's bound and assumption that edge connectivity $\geq d/2$, there are $n^{O(\ell)} = e^{O(\ell \log n)} < e^{\Omega(\ell d)}$ cuts of size $O(\ell d)$

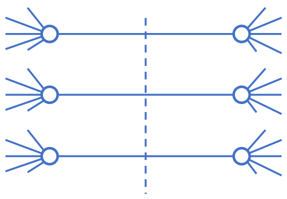
Cut Strain

- Let $a_v(S) \in \left\{0, \frac{1}{d}, \frac{2}{d}, \dots, 1\right\}$: fraction of vertex v's neighbors in set S
- Strain of cut $(S, \bar{S})$

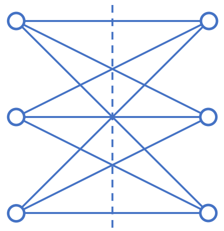
$$\sigma(S) = \sum_v a_v(S) \cdot a_v(\bar{S})$$

- $\sigma(S) \leq \sum_{v \in S} a_v(\bar{S}) + \sum_{v \in \bar{S}} a_v(S) = \frac{2c(S)}{d}$
- But, possibly, $\sigma(S) \ll \frac{2c(S)}{d}$

$\sigma(S) \approx \frac{2c(S)}{d}$



$\sigma(S) = 0$



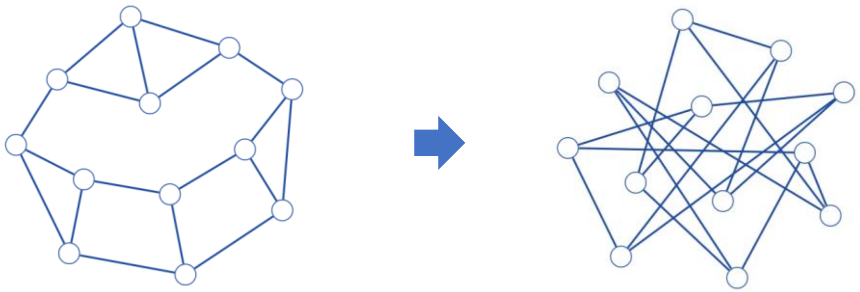
Conclusion of Part 1 of the Talk

The local flip process transforms any connected d-regular graph, with $d = \Omega(\log^2 n)$, to an expander after $O(nd \log^2 n)$ operations w.h.p.

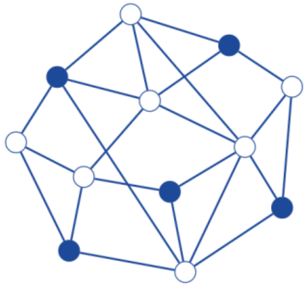
- Get rid of extra logarithmic factor ?
- Analysis for sub-logarithmic/constant degree d ?
- Bounds for vertex expansion ?
- Analysis of similar dynamic for bipartite graphs ?
- Improve existing bounds on the mixing time ?

Talk Overview

- Part 1: Transform any connected regular graph into an expander



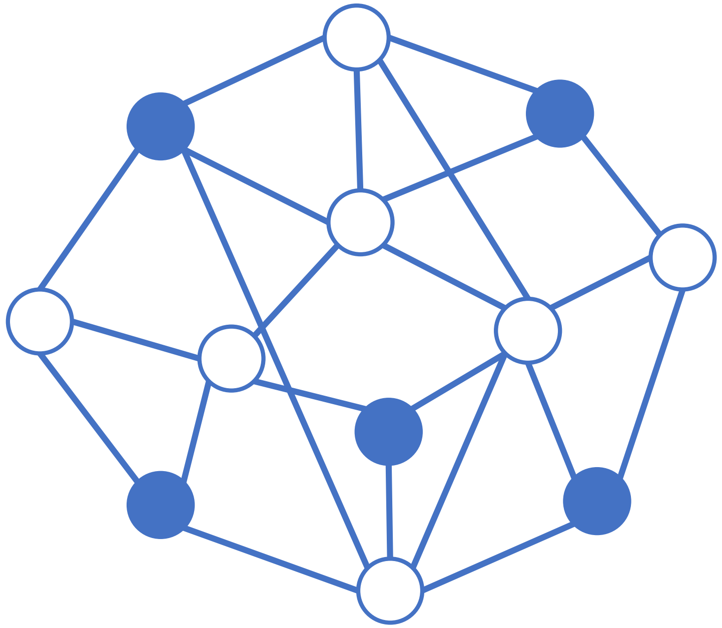
- Part 2: Compute a maximal-independent-set of any graph



Maximal Independent Set (MIS)

- Graph $G = (V, E)$

- $B \subseteq V$ is an MIS
  1. $u \in B \Rightarrow \nexists v \in B, v \sim u$
  2. $u \notin B \Rightarrow \exists v \in B, v \sim u$



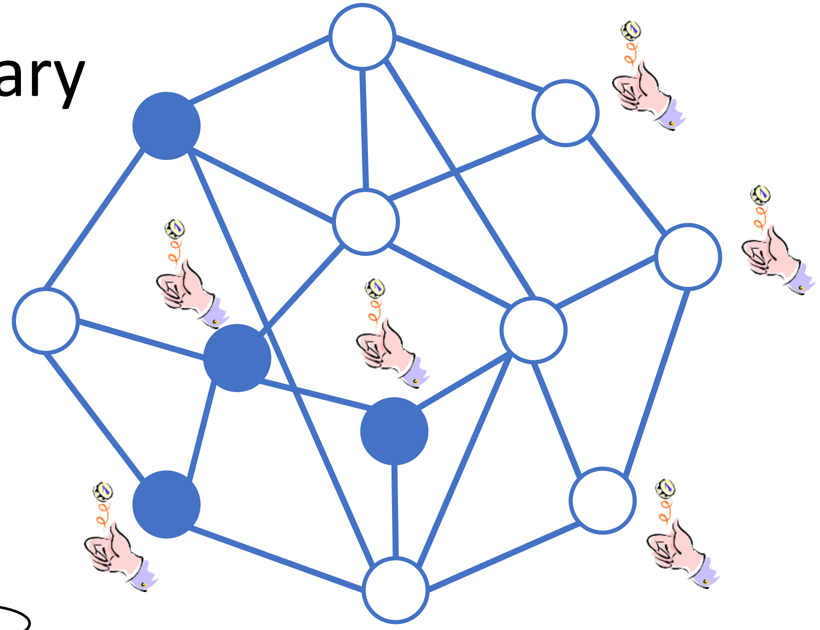
A simplest process for the MIS problem

- Arbitrary G
- Each u has state $s(u) \in \{0,1\}$, initially arbitrary
- All states updated in parallel rounds
- Update rule:

blue-blue error

If $(\ s(u) = 1\ \ \&\ \ \exists v \sim u, s(v) = 1\ )$
or $(\ s(u) = 0\ \ \&\ \ \nexists v \sim u, s(v) = 1\ )$ **then**
$s(u) \leftarrow \text{coin–flip}$

all-white error

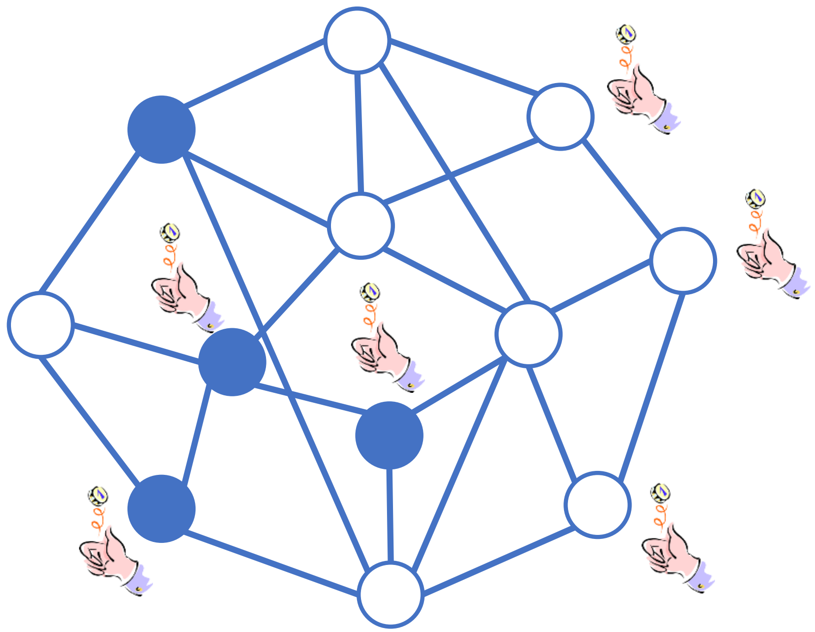


A simplest process for the MIS problem

- A vertex stabilizes if
  - is blue all its neighbors are white
  - is white and has a blue stabilized neighbor

- $B = \{u: s(u) = 1\}$
- Eventually, B becomes an MIS
- And at that point stabilizes

- Time until stabilization?



Properties

- Minimal state space: 2 states per vertex
- Minimal communication:
  - beeping model with sender collision detection (SCD)
  - 3-state variant for stone age model (w/o collision detection)

to see later

- Minimal computation: for stone age model
- Self-stabilizing (SS): works for any initial configuration

- And yet it has hardly ever been considered in literature !?!

Related: Sequential Version

- Folklore, also [Shukla, Rosenkrantz, and Ravi, 1995], [Hedetniemi, Hedetniemi, Jacobs, and Srimani, 2003]
- One vertex u updated per step / no randomization

If $(\ s(u) = 1\ \ \&\ \exists v \sim u, s(v) = 1\ )$
  or $(\ s(u) = 0\ \ \&\ \nexists v \sim u, s(v) = 1\ )$ **then**
    $s(u) \leftarrow 1 - s(u)$

- No blue-blue errors left after each u takes one step
- No all-white errors left after each u takes one additional step
- Stochastic scheduler: stabilization in $O(n \log n)$ steps, w.h.p.

Related: From Sequential to Parallel

[Shukla, Rosenkrantz, and Ravi, 1995]

- Adding randomization to updates yield a parallel algorithm that stabilizes (in at most exponential time)

[Turau and Weyer, 2006]

- Similar observations for any sequential self-stabilizing MIS algorithm

Other Related Work (Randomized + SS)

Algorithm	States	Communication	Knowledge	Runtime
New	2	Beeping-SCD	-	
New	3	Stone-age	-	
[Afek et al 2013]	$\text{poly}(\log n)$	Beeping	n	$O(\log^3 n)$
[Ghaffari 2017], [Jeavons et al 2016]	$O(\log n)$	Beeping-SCD	n	$O(\log n)$
[Emek and Keren 2021]	$\text{poly}(D)$	Stone-age	D	$O(D \log n)$
[Turau 2019]	$O(d_u)$	State-reading	-	$O(\log n)$

[Emek and Wattenhofer 2013]	$O(1)$	Stone-age	Non-SS	$O(\log^2 n)$

Other Related Work (Deterministic + SS)

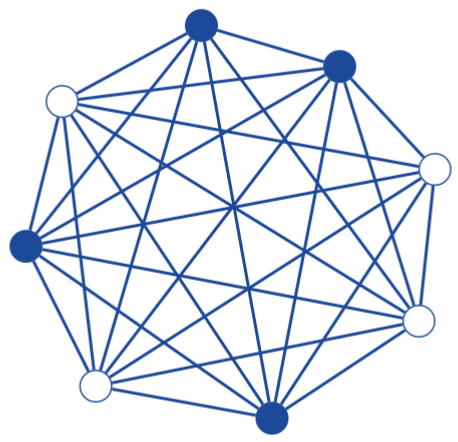
Algorithm	States	Communication	Knowledge	Runtime
[Ikeda et al 2002], [Goddard et al 2003], [Turau 2007]	ID + 2 or 3	State-reading	ID	$O(n)$
[Barenboim et al 2018]	$\text{poly}(n)$	Local	ID, n, Δ	$O(\Delta + \log^* n)$

Stabilization Time Bounds

Simple Bounds: Complete Graph

- In each step, half of the blue vertices become white on average
- All blue vertices become white before only one left, with constant probability < 0.61

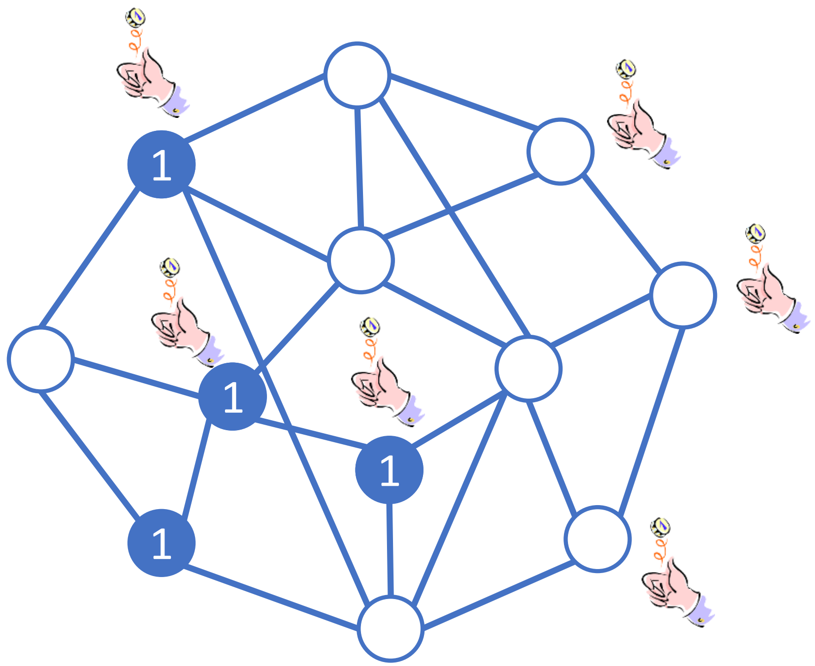
- Stabilization time
  - $O(\log n)$ expected
  - $O(\log^2 n)$ w.h.p.



3-State Process

- Each u has state $s(u) \in \{0,1,2\}$
- All states updated in parallel rounds
- Update rule:

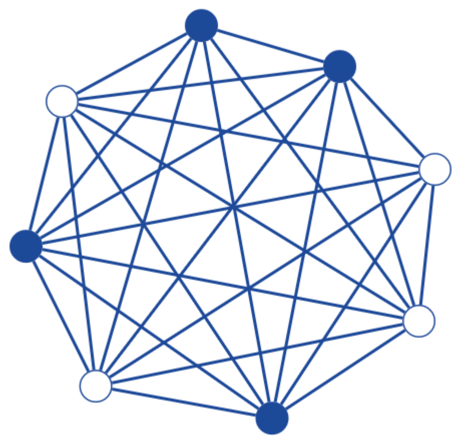
If $(\ s(u) = 0\ \ \&\ \nexists v \sim u, s(v) > s(u)\ )$ **then**
  $c_1 \leftarrow \text{coin-flip};\ c_2 \leftarrow \text{coin-flip}$
  $s(u) \leftarrow c_1 \cdot (1 + c_2)$
Elseif $(\ s(u) > 0\ \ \&\ \nexists v \sim u, s(v) > s(u)\ )$ **then**
  $s(u) \leftarrow (1 + \text{coin-flip})$
Elseif $(\ s(u) > 0\ \ \&\ \exists v \sim u, s(v) > s(u)\ )$ **then**
  $s(u) \leftarrow 0$



3-State Process: Complete Graph

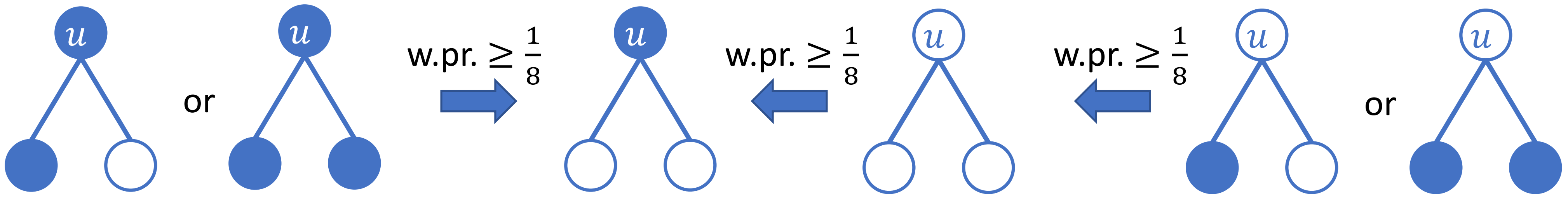
- In each step, ~half of the blue vertices become white on average
- ~~All blue vertices become white before only one left, with constant probability < 0.61~~

- Stabilization time
  - $O(\log n)$ expected and w.h.p.



Simple Bounds: Trees

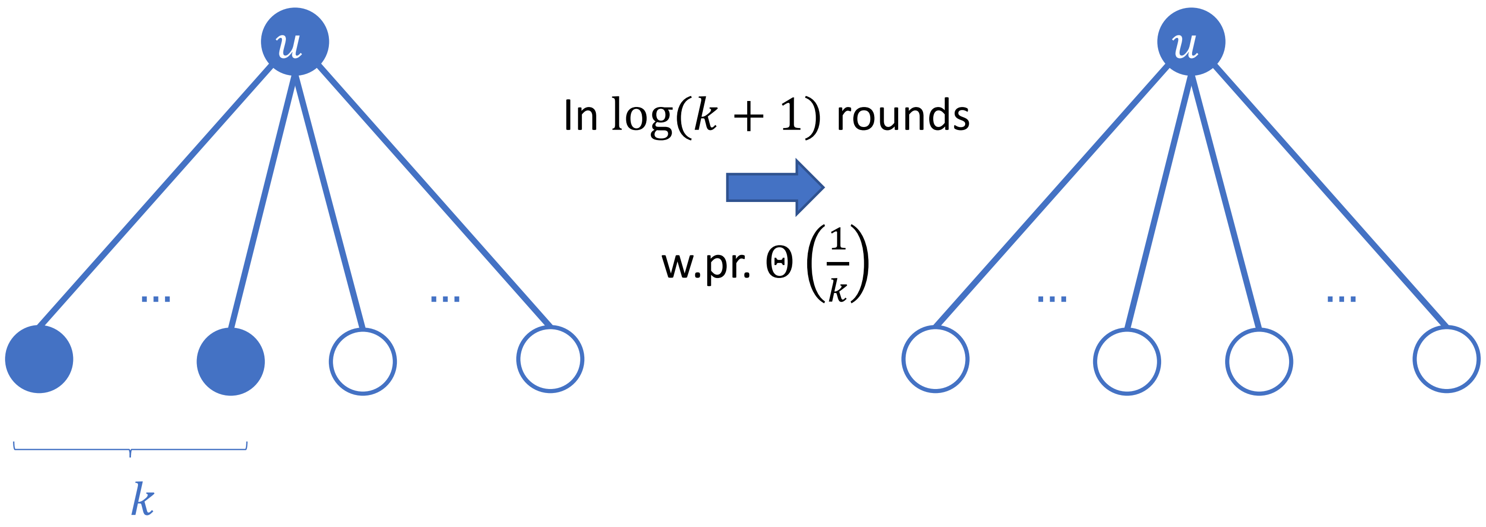
- Ignore stabilized vertices
- At least half of remaining vertices u have degree ≤ 2



- At least a cons fraction of vertices stabilize in two rounds on average
- Stabilization time $O(\log n)$ w.h.p.

Simple Bounds: General Graphs

- Lemma:



- Proof: The probability u is still blue after $r = \log(k+1)$ rounds and none of its k blue neighbors is, is

$$2^{-r}(1-2^{-r})^k \geq 2^{-r}4^{-k2^{-r}} = \frac{1}{4(k+1)}$$

Simple Bounds: General Graphs

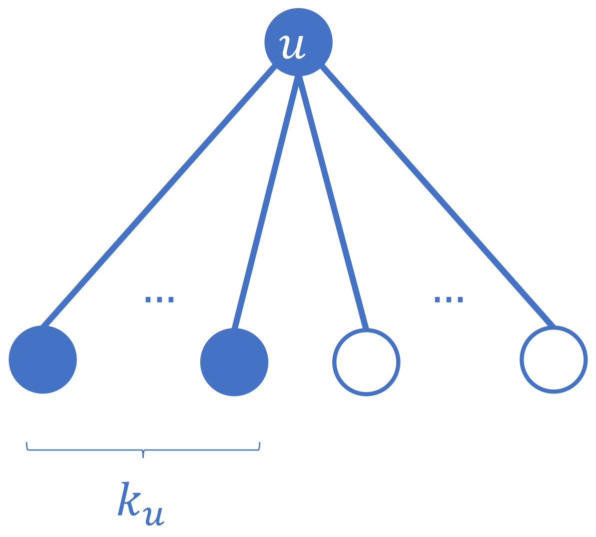
- Δ: maximum degree
- In $O(\log \Delta)$ rounds, u or a neighbor permanently joins B w.pr. $\Omega\left(\frac{1}{\Delta}\right)$
- Thus u stabilizes in
  - $O(\log \Delta)$ rounds w.pr. $\Omega\left(\frac{1}{\Delta}\right)$
  - $O(\Delta \sout{\log \Delta} \log n)$ rounds w.h.p.

- Stabilization time $O(\Delta \log n)$ w.h.p.

A More Refined Lemma

- k_u: number of blue neighbors of u
- $B_u = B \cap (\Gamma_u \cup \{u\})$
- "Goodness" of u

$$g_u = \sum_{v \in B_u} \frac{1}{k_v + 1}$$



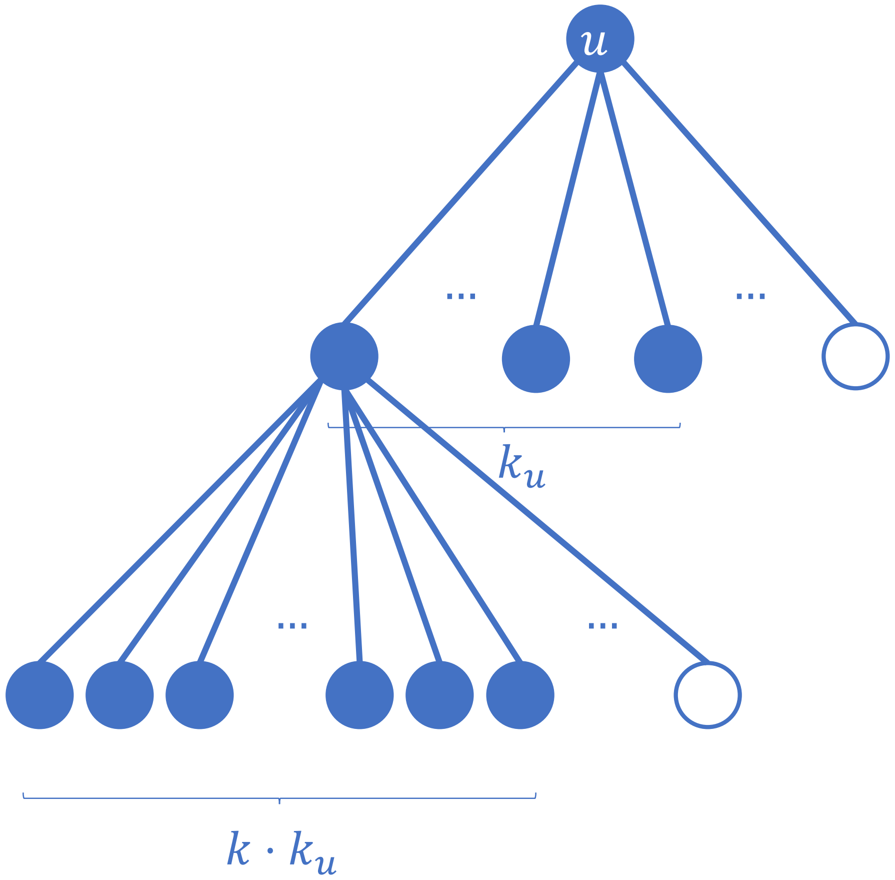
- Lemma: The prob. u stabilizes in $O(\log \Delta)$ rounds is $\Omega(\min\{1, g_u\})$

Low Goodness

If $g_u \leq \frac{1}{k}$ then

- $\sum_{v \in B \cap \Gamma_u} \frac{1}{k_v+1} = O\left(\frac{1}{k}\right)$, i.e., the harmonic mean of $k_v + 1$ is $\Omega(k \cdot k_u)$
- and $k_u = \Omega(k)$, if $u \in B$

- In words: u's blue neighbors have many more blue neighbors than u



Erdős-Rényi Random Graph $G_{n,p}$

- n vertices
- Each edge is present independently w.pr. $p \in [0,1]$
- Average degree $p \cdot n$

- [Giakkoupis and Ziccardi, in prep.]

For $0 \leq p \leq n^{-\epsilon}$ and any $\epsilon > 0$, the stabilization time is $\text{poly}(\log n)$ w.h.p.

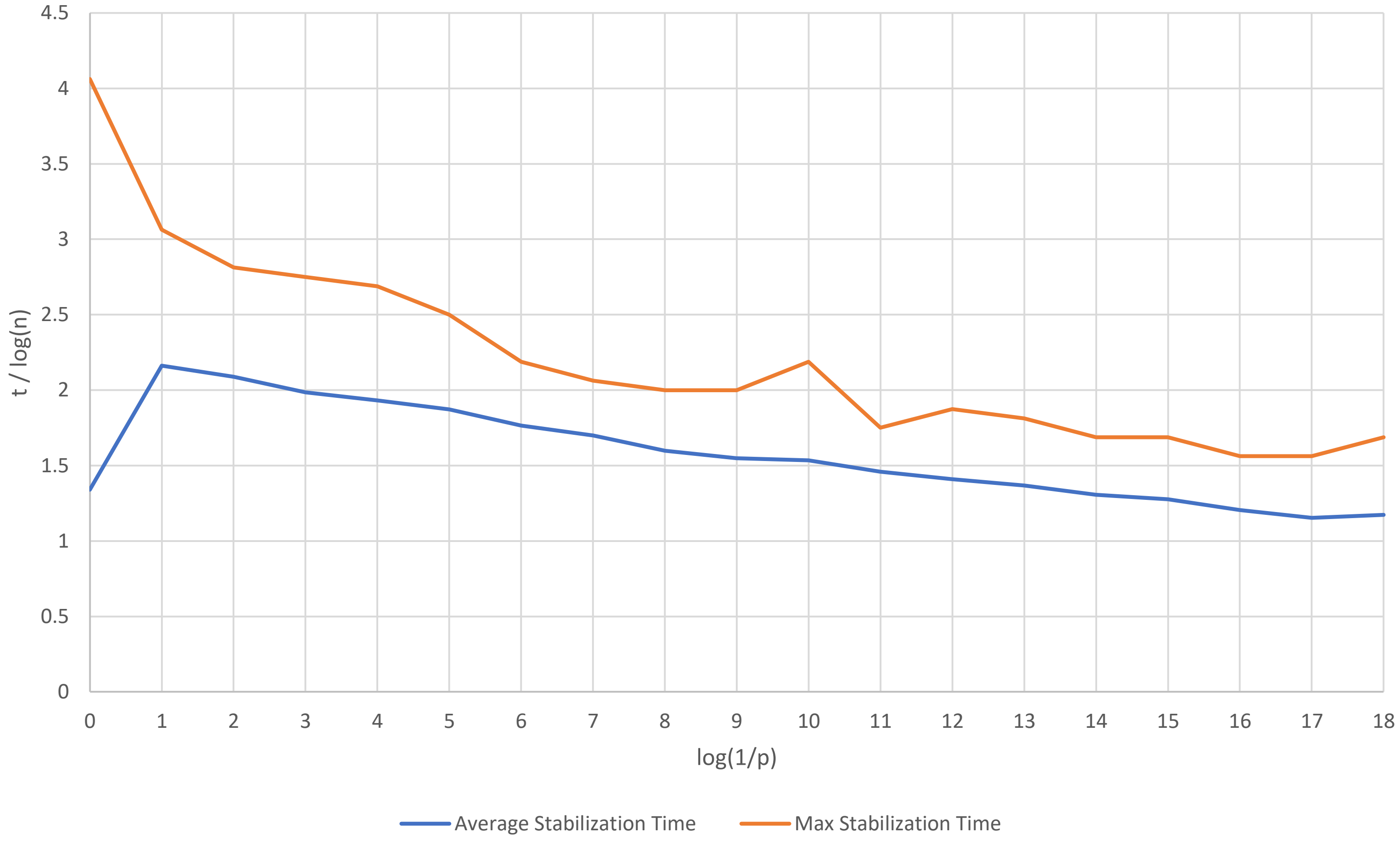
Avg degree $0 \ldots n^{1-\epsilon}$

Analysis Overview

- u is "good" if $g_u \geq \frac{1}{\log n}$
- W.h.p. (on the randomness of the graph), for any given configuration (or for the conf. one round later), a large enough fraction of the non-stabilized vertices are good
- … except if the #of non-stabilized vertices is small $\left(\leq \frac{\mathrm{poly}(\log n)}{p}\right)$
- In this case, progress is slow only if there exist vertices which switch state frequently, over a long period
- We show this to be highly unlikely unless p is too large

Simulation: Stabilization Time on $G_{n,p}$:

- $n = 2^{16}$
- $p = 2^{-i}, i = 0, \dots, 18$
- 100 iterations for each p
- All-1s initially



Conclusion of Part 2 of the Talk

- Is this simplest MIS process fast for all/most families of graphs?

Algorithm	States	Communication	Knowledge	Runtime
New	2	Beeping-SCD	-	$O(\log^2 n)$?
New	3	Stone-age	-	$O(\log n)$?
[Afek et al 2013]	$\mathrm{poly}(\log n)$	Beeping	n	$O(\log^3 n)$
[Jeavons et al 2016], [Ghaffari 2017]	$O(\log n)$	Beeping-SCD	n	$O(\log n)$
[Emek and Keren 2021]	$\mathrm{poly}(D)$	Stone-age	D	$O(D \log n)$
[Turau 2019]	$O(d_u)$	State-reading	-	$O(\log n)$

$\mathrm{poly}(\log n)$? (first two rows)

Thank you!