



HAL
open science

Distributed Self-Stabilizing MIS with Few States and Weak Communication

George Giakkoupis, Isabella Ziccardi

► **To cite this version:**

George Giakkoupis, Isabella Ziccardi. Distributed Self-Stabilizing MIS with Few States and Weak Communication. PODC 2023 - ACM Symposium on Principles of Distributed Computing, Jun 2023, Orlando (FL), United States. pp.310-320, 10.1145/3583668.3594581 . hal-04393730

HAL Id: hal-04393730

<https://inria.hal.science/hal-04393730>

Submitted on 15 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Distributed Self-Stabilizing MIS with Few States and Weak Communication

George Giakkoupis
Inria, Univ Rennes, CNRS, IRISA
Rennes, France
george.giakkoupis@inria.fr

Isabella Ziccardi
Bocconi University
Milan, Italy
isabella.ziccardi@unibocconi.it

ABSTRACT

We study a simple random process that computes a maximal independent set (MIS) on a general n -vertex graph. Each vertex has a binary state, black or white, where black indicates inclusion into the MIS. The vertex states are arbitrary initially, and are updated in parallel: In each round, every vertex whose state is “inconsistent” with its neighbors, i.e., it is black and has a black neighbor, or it is white and all neighbors are white, changes its state with probability $1/2$. The process stabilizes with probability 1 on any graph, and the resulting set of black vertices is an MIS. We show that the expected stabilization time is $O(\log n)$ on certain graph families, such as cliques and graphs of bounded arboricity.

Our main result is that the process stabilizes in $\text{poly}(\log n)$ rounds w.h.p. on $G_{n,p}$ random graphs, for $0 \leq p \leq \text{poly}(\log n) \cdot n^{-1/2}$ or $p \geq 1/\text{poly}(\log n)$. Further, we propose an extension of this process, with larger but still constant vertex state space, which stabilizes in $\text{poly}(\log n)$ rounds on $G_{n,p}$ w.h.p., for all $1 \leq p \leq 1$. Both processes readily translate into distributed/parallel MIS algorithms, which are self-stabilizing, use constant space (and constant random bits per round), and assume restricted communication as in the beeping or the synchronous stone age models. To the best of our knowledge, no previously known MIS algorithm is self-stabilizing, uses constant space and constant randomness, and stabilizes in $\text{poly}(\log n)$ rounds on $G_{n,p}$ random graphs.

CCS CONCEPTS

• **Theory of computation** → *Distributed algorithms; Graph algorithms analysis*; • **Mathematics of computing** → *Graph algorithms*.

KEYWORDS

maximal independent set, parallel/distributed algorithms, self-stabilizing algorithms, beeping model

ACM Reference Format:

George Giakkoupis and Isabella Ziccardi. 2023. Distributed Self-Stabilizing MIS with Few States and Weak Communication. In *ACM Symposium on Principles of Distributed Computing (PODC '23)*, June 19–23, 2023, Orlando, FL, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

PODC '23, June 19–23, 2023, Orlando, FL, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0121-4/23/06...\$15.00
<https://doi.org/10.1145/3583668.3594581>

USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3583668.3594581>

1 INTRODUCTION

Finding a maximal independent set (MIS) is a fundamental problem in parallel and distributed computing. Given a graph $G = (V, E)$, the objective is to identify a set of vertices $S \subseteq V$ such that no two vertices $u, v \in S$ are adjacent to each other (*independence* property), and no vertex $u \in V \setminus S$ can be added to S without violating independence (*maximality* property). The significance of the problem in parallel computing was first recognized in the early 80s [6, 25], due to its various applications in symmetry breaking [19], and it has been studied extensively ever since (see [5] for a review of work until 2015, and [3, 13] for state of the art results).

In this paper, we explore simple distributed random processes on graphs that find an MIS starting from arbitrary initial states of the vertices. These processes immediately translate into self-stabilizing [8, 9] synchronous distributed algorithms for network systems with severely restricted computation and communication capabilities, such as wireless sensor networks. The processes we consider are also relevant to certain biological cellular networks. For example, it is known that a biological process occurring during the development of the nervous system of a fly is equivalent to computing an MIS [2, 18].

The main random process we consider, which we call the *2-state MIS process*, is as follows. Each vertex has a binary state, *black* or *white*, where black indicates inclusion into the MIS. The vertex states are arbitrary initially and are updated in synchronous rounds. In each round, every vertex u whose state violates the independence or maximality properties, i.e., u is black and has a black neighbor, or it is white and has no black neighbor, changes its state to the opposite state with probability $1/2$. It is easy to see that the state of a vertex stabilizes as soon as it is black and has no black neighbors, or it is white and has a *stabilized black* neighbor; and when all vertices have stabilized, the set of black vertices is an MIS. It is also immediate that, on any graph G , the process stabilizes eventually with probability 1, due to the randomization.¹

The 2-state MIS process can be viewed as a natural parallelization (with the addition of randomness) of a simple self-stabilizing sequential deterministic algorithm, proposed in [16, 21], where in each step a single vertex updates its state (from black to white if the vertex has a black neighbor, and from white to black if it has no black neighbors). [21] also observed that by randomizing the transitions of the sequential algorithm we obtain an algorithm that stabilizes with probability 1 on a general adversarial scheduler

¹We could have defined the process so that the transition from white to black (when the white vertex has no black neighbors) occurs with probability 1, but we opted for a randomized transition because it simplifies our analysis.

model, which includes the synchronous model. A similar observation follows from a general transformation framework proposed in [24]. The sequential algorithm is known to stabilize after each process has taken at most 2 steps, regardless of the scheduling order. However, analyzing the stabilization time of the parallel process seems a much more challenging problem, and has not been studied before.

The 2-state MIS process directly translates into a self-stabilizing MIS algorithm for the harsh beeping communication model [7]. In this model, in every synchronous round, each vertex either beeps or listens, and a listening vertex can only differentiate between none of its neighbors beeping, or at least one beeping. In our case, we can let black vertices beep in each round, while white vertices listen. Black vertices must be able to detect collisions, otherwise they cannot tell if they have a black neighbor, thus we assume the beeping model version with *sender collision detection* (a.k.a. full-duplex model) [1, 12].

We also propose a simple variant of the 2-state MIS process, called the *3-state MIS process*, which has an additional state and does not require collision detection (see Definition 5). This variant is suitable for the synchronous stone age model [10, 11]. The synchronous stone age model can be viewed as an extension of the beeping model over a constant number of channels (without collision detection): each vertex beeps in at most one channel and listens to the other channels.

The algorithms obtained from the 2-state and 3-state MIS processes have several attractive properties: they use a constant number of states (2 or 3) and one random bit per round, they do not require vertex IDs or any global graph information (such as the number of vertices n or the maximum degree Δ), assume very weak communication (the beeping or stone age models), they are self-stabilizing, and are extremely simple. We will prove that, on some families of graphs, the 2-state algorithm is also *fast*, i.e., it stabilizes (from an arbitrary initial state) in a number of rounds that is poly-logarithmic in n , w.h.p. In this paper, we do not analyze the 3-state MIS process, but we expect that it behaves similarly to the 2-state MIS process.

Several self-stabilizing distributed MIS algorithms have been proposed in the literature, but as far as we know, none possesses all the above properties. Known fast self-stabilizing MIS algorithms for the beeping model require (approximate) knowledge of n , use space that is a super-constant function of n , and require a super-constant number of random bits [1, 12, 18]. In the stone age model, an MIS algorithm proposed in [11] has similar properties as our algorithms, and is provably fast for all graphs, but is not self-stabilizing; while a self-stabilizing algorithm for the model proposed recently in [10] is fast only on graphs whose diameter is bounded by a known constant. A fast self-stabilizing MIS algorithm proposed in [23] requires super constant state and communication. All the above algorithms are randomized. Known deterministic self-stabilizing MIS algorithms require distinct vertex IDs, super constant state and communication, and are in general much slower than the randomized algorithms, stabilizing in time linear in n or in the maximum degree Δ [4, 15, 17, 22].

1.1 Our Contribution

We first analyze the stabilization time of the 2-state MIS process on complete graphs and on graphs with bounded arboricity.² We also provide an upper bound in terms of the maximum degree for general graphs. The proof of these results is mostly straightforward.

THEOREM 1. *The stabilization time of the 2-state MIS process on an n -vertex graph G is*

- $O(\log n)$ in expectation and $O(\log^2 n)$ w.h.p., if G is the complete graph K_n .
- $O(\log n)$ w.h.p., if G has bounded arboricity.
- at most $O(\Delta \log n)$ w.h.p., if the maximum degree of G is Δ .

A main technical contribution of the paper is the analysis of the 2-state MIS process on Erdős-Rényi $G_{n,p}$ random graphs. We show a poly-logarithmic upper bound for $G_{n,p}$ random graphs when the average degree np is at most $\text{poly}(\log n) \cdot \sqrt{n}$. The same bound is easily obtained also when the average degree is at least $n/\text{poly}(\log n)$.

THEOREM 2. *The stabilization time of the 2-state MIS process on a $G_{n,p}$ random graph, such that $0 \leq p \leq \text{poly}(\log n) \cdot n^{-1/2}$ or $p \geq 1/\text{poly}(\log n)$, is at most $\text{poly}(\log n)$ w.h.p.*

Our proof techniques do not suffice to obtain a poly-logarithmic upper bound for the 2-state MIS process on $G_{n,p}$ for the complete range of p . Our second technical contribution is an extension of the 2-state MIS process, which provably stabilizes in poly-logarithmic time w.h.p. on $G_{n,p}$ for all $0 \leq p \leq 1$. The extended process has similar attractive properties as the 2-state MIS process, in particular, the number of states is still constant. It uses as a sub-process a phase clock process proposed in [10]. Interestingly, unlike [10], we do not use the phase clock for synchronization, but rather as a local non-synchronized counter (see Section 1.2 for a more detailed discussion).

THEOREM 3. *There is an extension of the 2-state process, with 18 states, such that the stabilization time of the process on a $G_{n,p}$ random graph, for any $0 \leq p \leq 1$, is at most $\text{poly}(\log n)$ w.h.p.*

It is an open problem whether the bound of Theorem 3 holds for the 2-state MIS process, as well. We believe that it does. In fact, we conjecture that the stabilization time of the 2-state MIS process is $\text{poly}(\log n)$ w.h.p. on *any given n -vertex graph*. We also conjecture that the same is true for the 3-state MIS process. For the 2-state process, we cannot hope to prove an $O(\log n)$ upper bound for all graphs, as the process requires $\Theta(\log^2 n)$ rounds to stabilize w.h.p. on the complete graph K_n .³ For the 3-state process, however, we have no example of a graph where the stabilization time is larger than $O(\log n)$.

1.2 Analysis Overview and Techniques

Below we give an overview of the analysis of the 2-state MIS process and its extension, on $G_{n,p}$ random graphs.

To avoid having to deal simultaneously with the randomness of the graph and the arbitrary initialization of vertex states, we deal

²The arboricity of a graph is the minimum number of forests into which we can partition its edges.

³It also requires $\Theta(\log^2 n)$ rounds in expectation to stabilize on a graph consisting of \sqrt{n} disjoint cliques $K_{\sqrt{n}}$.

with graph randomness first. We define a family of *good graphs*, containing those graphs that satisfy all structural properties that we will need for the analysis, such as bounds on the average degree of any induced subgraph, and bounds on the number of common neighbors of any two vertices (see Definition 18). We then show that a $G_{n,p}$ random graph is good w.h.p., and assume an arbitrary good graph in the analysis.

The analysis proceeds by showing that starting from any vertex states, the process makes sufficient progress after $O(\log n)$ rounds, where progress is measured by the expected number of vertices that stabilize.

In the 2-state MIS process, we call a vertex *active* if it is black and has a black neighbor, or it is white and has no black neighbors. Thus, active vertices change their state to a uniformly random state in the next step. A vertex is *k-active* if it is active and has *at most k active neighbors*.

An elementary property of the 2-state MIS process is that if a vertex is *k-active*, then it becomes stabilized black in $O(\log k)$ rounds with probability $\Omega(1/k)$. We use also a generalization of this property to sets of active vertices.⁴ These two properties, combined with structural properties of good graphs, suffice to show the desired expected progress when the number of non-stabilized vertices is large enough or the number of active vertices is large enough. Moreover, the analysis of this case does not require any constraints on the value of p .

The more difficult case is when the number of non-stabilized vertices is relatively small, namely $O(p^{-1} \log^2 n)$, and a smaller than $1/\text{poly}(\log n)$ fraction of them are active. One may expect this to be an easy case, since the induced subgraph on a *random* subset of $O(p^{-1} \log^2 n)$ vertices has maximum degree $\Delta = O(\log^2 n)$ w.h.p., and Theorem 1 gives an $O(\log^3 n)$ bound for that Δ . However, the above bound on Δ does not apply to the induced subgraph on an *arbitrary* subset of $O(p^{-1} \log^2 n)$ vertices. Nevertheless, it is true that the *average* degree of that subgraph is $O(\log^2 n)$, thus at least a constant fraction of vertices have degree $O(\log^2 n)$.

Let u be one such vertex, i.e., u has degree $d = O(\log^2 n)$ in the induced subgraph of non-stabilized vertices. To prevent u from becoming active (and thus d -active) or becoming stabilized, in each round at least one neighbor v of u must be non-stabilized black. We show that, roughly speaking, if a vertex v has probability b of becoming non-stabilized black at some point during an interval of r rounds (ignoring the first few rounds, e.g., if v is black initially), then v has probability at least $\text{poly}(b/r)$ of becoming θ -active in that interval, where θ is bounded by $O(\log(r/b))$ times the maximum number of common neighbors two vertices may have, which is $\text{poly}(\log n)$ if $p \leq \text{poly}(\log n) \cdot n^{-1/2}$ (see Section 4.1 for the relevant lemmas).

For the purposes of our analysis, it suffices to set $r = \Theta(\log \log n)$. If each of the d neighbors v of u has probability less than $1/(2d)$ of becoming non-stabilized black in the next r rounds, then u has a constant probability of becoming d -active (or stabilize). On the other hand, if there is some neighbor v that has probability $b \geq 1/(2d)$ of becoming non-stabilized black in the next r rounds, we saw above that v becomes θ -active with probability at

least $\text{poly}(b/r) = 1/\text{poly}(\log n)$, where $\theta = \text{poly}(\log n)$. We conclude that, with probability $1/\text{poly}(\log n)$, u is $\text{poly}(\log n)$ -active or has a $\text{poly}(\log n)$ -active neighbor at some point in the next $r = \Theta(\log \log n)$ rounds. It follows that u stabilizes with probability $1/\text{poly}(\log n)$ in the next $O(\log n)$ rounds.⁵

When $p > \text{poly}(\log n) \cdot n^{-1/2}$, the second case of the analysis above does not give a poly-logarithmic bound. A solution to this problem would be to somehow limit how frequently a vertex v can change its state from white to black, namely, at most once every $\Theta(\log^2 n)$ rounds. In fact, it would suffice if v can change its state from white to black at most once every $\Theta(\log n)$ rounds.⁶ We extend the 2-state MIS process by incorporating a mechanism which achieves roughly that.

We call the new process the *3-color MIS process*. It consists of two sub-processes running in parallel. The first is similar to the 2-state MIS process with the addition of a third color, *gray*: a black vertex now becomes gray instead of white, a gray vertex becomes white after a while, and other vertices treat gray vertices as white. The transition from gray to white is controlled by the second sub-process, called the *logarithmic switch*.

In the logarithmic switch, each vertex has an on/off binary variable, and a gray vertex changes to white if the switch variable of the vertex is on. We would like that the logarithmic switch satisfy two basic properties: (i) a vertex switches from off to on every $\Theta(\log n)$ rounds; and (ii) it switches from on to off every $O(1)$ rounds. However, we do not know how to implement property (i) using constant states. Instead, we observe that it suffices if property (i) is satisfied only when $p > \text{poly}(\log n) \cdot n^{-1/2}$; for smaller p , a weaker property suffices: (i') a vertex switches from off to on after *at most* $O(\log n)$ rounds. It is not immediately obvious how to implement this distinction, because we want the process to work for all $0 \leq p \leq 1$ *without knowing* p (or anything else about the graph topology).

We achieve that as follows. We exploit the fact that if $p > \text{poly}(\log n) \cdot n^{-1/2}$ then the graph has constant diameter (in fact diameter 2). The logarithmic switch process we devise is similar to the phase clock process *RandPhase* proposed in [10]. *RandPhase* assumes that an upper bound D on the graph diameter is available to the process and uses $D + 3$ states. The core mechanism of the logarithmic switch is identical to that of *RandPhase* for $D = 3$ (not 2!), but the underlying graph may have arbitrary (and unknown) diameter. The logarithmic switch includes also a mapping of the states to the on/off values of the switch. Unlike *RandPhase* which is used for synchronization (it achieves synchronous phases of length $D + \Theta(\log n)$), the purpose of the logarithmic switch is not synchronization, as it is not required that the switch variables of different vertices change simultaneously.

1.3 Notation

We fix now some standard notation that we use throughout the paper. Let $G = (V, E)$ be a graph on n vertices. For each vertex $u \in V$, $N(u) = \{v : (u, v) \in E\}$ is the set of neighbors of u , and $N^+(u) = N(u) \cup \{u\}$. Similarly, for a set of vertices $S \subseteq V$, we

⁵We suspect that a refinement of this argument may be useful for a broader class of graphs.

⁶The reason is that, in the induced subgraph on $O(p^{-1} \log^2 n)$ vertices considered in the second case of the analysis of the 2-state MIS process, at least a constant fraction of vertices have at most $O(\log n)$ neighbors of degree $\Omega(\log^3 n)$.

⁴Similar properties are commonly used in the analysis of distributed MIS algorithms in the literature.

define $N(S) = \bigcup_{u \in S} N(u) \setminus S$ and $N^+(S) = \bigcup_{u \in S} N^+(u) = N(S) \cup S$. For two (not necessarily disjoint) sets $S, T \subseteq V$, we let $E(S, T) = \{(u, v) \in E : u \in S, v \in T\}$ be the set of edges with one endpoint in S and the other in T . We also define $E(S) = E(S, S)$. By $G[S]$ we denote the induced subgraph of G on $S \subseteq V$, i.e., $G[S] = (S, E(S))$.

Roadmap. The rest of the paper is organized as follows. Section 2 contains the definition and some basic properties of the 2-state and 3-state MIS processes. Section 3 provides a proof of Theorem 1. Section 4 proves Theorem 2. Section 5 defines the 3-color MIS process and proves Theorem 3. Due to space limitations some of the proofs are omitted, and can be found in the full version of the paper [14].

2 THE 2-STATE AND 3-STATE MIS PROCESSES

2.1 Definition of the Processes

We define two self-stabilizing distributed graph processes that compute a maximal independent set when applied on any given graph.

Definition 4 (2-State MIS Process). In the 2-state MIS process on graph $G = (V, E)$, each vertex $u \in V$ has a binary state from the set $\{\text{black}, \text{white}\}$, and all states are updated in parallel rounds. The initial state $c_0(u)$ of vertex u can be arbitrary, and in each round $t = 1, 2, \dots$, u 's state is updated from $c_{t-1}(u)$ to $c_t(u)$ according to the following rule:

```

let  $NC_t(u) = \{c_{t-1}(v) : v \in N(u)\}$ 
if ( $c_{t-1}(u) = \text{black}$  and  $NC_t(u) \ni \text{black}$ ) or
( $c_{t-1}(u) = \text{white}$  and  $NC_t(u) \not\ni \text{black}$ ) then
  | let  $c_t(u)$  be a uniformly random state from
  |  $\{\text{black}, \text{white}\}$ 
else set  $c_t(u) = c_{t-1}(u)$ 

```

We say that a vertex is *black* or *white* if its state is black or white, respectively. The vertex is *active* if it is black and has some black neighbor, or it is white and has no black neighbors. It is *stable* if either it is black and has no black neighbors, or it is white and has a neighbor that is black *and stable*. It is immediate from the update rule that once a vertex becomes stable, it remains stable thereafter, and its state no longer changes. The *stabilization time of vertex u* is the earliest round at the end of which u is stable. The *stabilization time of the process* is the earliest round at the end of which all vertices are stable. It is easy to verify that after the stabilization time of the process, the set of black vertices is an MIS of G .

We define the following sets for each round $t \geq 0$: $B_t = \{u \in V : c_t(u) = \text{black}\}$ is the set of black vertices at the end of round t ; $W_t = V \setminus B_t$ is the set of white vertices;

$$A_t = \{u \in B_t : N(u) \cap B_t \neq \emptyset\} \cup \{u \in W_t : N(u) \cap B_t = \emptyset\}$$

is the set of active vertices; $I_t = \{u \in B_t : N(u) \cap B_t = \emptyset\}$ is the set of stable black vertices (note that I_t is an independent set and is a subset of the final MIS); and $V_t = V \setminus N^+(I_t)$ is the set of non-stable vertices.

Definition 5 (3-State MIS Process). In the 3-state MIS process on graph $G = (V, E)$, each vertex $u \in V$ has a state from the set

$\{\text{black1}, \text{black0}, \text{white}\}$, and the states are updated in parallel rounds. The initial state $c_0(u)$ of u is arbitrary, and in each round $t \geq 1$, u 's state is updated as follows:

```

let  $NC_t(u) = \{c_{t-1}(v) : v \in N(u)\}$ 
if  $c_{t-1}(u) = \text{black1}$  or ( $c_{t-1}(u) = \text{black0}$  and
 $NC_t(u) \not\ni \text{black1}$ ) or ( $c_{t-1}(u) = \text{white}$ 
and  $NC_t(u) = \{\text{white}\}$ ) then
  | let  $c_t(u)$  be a uniformly random state from
  |  $\{\text{black1}, \text{black0}\}$ 
else if  $c_{t-1}(u) = \text{black0}$  then
  | set  $c_t(u) = \text{white}$ 
else set  $c_t(u) = c_{t-1}(u)$ 

```

In the 3-state MIS process, we say that a vertex u is *black* when its state is black1 or black0. Then the *stable* vertices and the *stabilization times* are defined as before. Note that the state of a stable black vertex alternates perpetually between states black1 and black0.

At the beginning of a round, a black vertex is equally likely to be in any of the two black states. Thus (before revealing the precise states of black vertices) a black vertex with at least one black neighbor has a constant probability of becoming white, similarly to the 2-state MIS process. In the 3-state process, however, a black vertex does not need to know if it has a neighbor in the same state as itself, thus it can be implemented without sender collision detection. Another difference is that, in the 3-state MIS process, it is impossible for a black vertex and its black neighbors to all become white in the same round. We will see in Remark 11 that this results in asymptotically faster stabilization times in complete graphs, compared to the 2-state MIS process.

In this paper we focus on the 2-state MIS process, but we expect that all our upper bound results should carry over to the 3-state MIS process.

2.2 Basic Properties of the 2-State MIS Process

We now show some elementary properties of the 2-state MIS process. In the analysis, it will be convenient to assume that at the beginning of each round $t \geq 1$, we flip for each vertex u an independent coin $\phi_t(u)$ such that $\mathbb{P}[\phi_t(u) = \text{black}] = \mathbb{P}[\phi_t(u) = \text{white}] = 1/2$. Then if u must update its state to a random state in that round, i.e., if $u \in A_{t-1}$, we set $c_t(u) = \phi_t(u)$; while if $u \notin A_{t-1}$, then $\phi_t(u)$ is not used by the process.

The lemmas below apply for any graph $G = (V, E)$, and the probabilistic statements assume that we know the states of vertices at the end of round t (i.e., B_t or W_t is given). The first lemma says that an active vertex u with k active neighbors has probability $\Omega(1/k)$ to become stable black in the next $O(\log k)$ rounds.

LEMMA 6. *If $u \in A_t$ and $|N(u) \cap A_t| = k \geq 1$, then the probability that $u \in I_{t+\lceil \log(k+1) \rceil}$ is at least $(2ek)^{-1}$.*

PROOF. Let $r = \lceil \log(k+1) \rceil$. The probability that $u \in I_{t+r}$ is lower bounded by the probability that $\phi_{t+1}(v) = \dots = \phi_{t+r}(v) = \text{black}$ holds for $v = u$ and does not hold for any $v \in N(u) \cap A_t$, which is

$$(1/2)^r \cdot (1 - (1/2)^r)^k \geq (1/2)^r \cdot e^{-k/(2^{r-1})} \geq (1/2k) \cdot (1/e). \quad (1)$$

For the first inequality we used the fact $(1 - 1/n)^{n-1} \geq e^{-1}$, and for the second we used that $r \leq \log(k) + 1$ and $r \geq \log(k + 1)$. \square

Remark 7. The proof of Lemma 6 implies the slightly stronger result that if $u \in A_t$ and $|N(u) \cap A_t| = k \geq 1$, then with probability at least $(2ek)^{-1}$ it holds that $u \in B_{t'}$ for all $t' > t$ and $u \in I_{t+\log(k+1)}$.

The next statement is a generalization of Lemma 6 to multiple active vertices u_1, \dots, u_ℓ . We will apply this result to the set of active neighbors of a vertex u , to lower bound the probability that u is stable after a logarithmic number of rounds because some neighbor becomes stable black. The proof can be found in [14].

LEMMA 8. *Suppose that $u_1, \dots, u_\ell \in A_t$, and $|N(u_i) \cap A_t| = k_i > 0$, for each $1 \leq i \leq \ell$. Then the probability that $\{u_1, \dots, u_\ell\} \cap I_{t+\log(\max_i k_i)} \neq \emptyset$ is at least $(1/5) \cdot \min\{1, \sum_i (2k_i)^{-1}\}$.*

3 SIMPLE BOUNDS FOR THE 2-STATE MIS PROCESS

In this section, we show some simple bounds on the stabilization time of the 2-state MIS process on certain graph families, namely, complete graphs and graphs of bounded arboricity. We also show a basic upper bound in terms of the maximum degree on general graphs.

THEOREM 9. *The stabilization time of the 2-state MIS process on the complete graph $K_n = (V, E)$ is $O(\log n)$ in expectation and $O(\log^2 n)$ w.h.p. More concretely, for any $k > 0$, the stabilization time is at least $k \cdot \log n$ with probability $2^{-\Theta(k)}$.*

PROOF. We call round t *critical* if $|B_t| \in \{0, 1\}$, and call it *stable* if $|B_t| = 1$. Let p_i be the probability that the next critical round is stable, given that $|A_t| = i \geq 2$. Note that in K_n , $A_t = B_t$ if $|B_t| > 1$, $A_t = \emptyset$ if $|B_t| = 1$, and $A_t = V$ if $B_t = \emptyset$; thus $|A_t| \neq 1$. We argue that for any $i \geq 2$,

$$2/3 \leq p_i \leq 17/21.$$

The lower bound follows from the observation that, for any $i \geq 2$ and $t \geq 0$,

$$\mathbb{P}[|B_{t+1}| = 1 \mid |B_{t+1}| \leq 1, |A_t| = i] = \frac{\binom{i}{1} 2^{-i}}{\binom{i}{1} 2^{-i} + 2^{-i}} = \frac{i}{i+1} \geq 2/3.$$

For the upper bound we have $p_2 = 2/3 < 17/21$. And, for $i \geq 3$,

$$\begin{aligned} \mathbb{P}[|B_{t+1}| \neq 1 \mid |B_{t+1}| \leq 2, |A_t| = i] &= \frac{\binom{i}{2} 2^{-i} + 2^{-i}}{\binom{i}{2} 2^{-i} + \binom{i}{1} 2^{-i} + 2^{-i}} \\ &= \frac{i^2 - i + 2}{i^2 + i + 2} \geq 4/7. \end{aligned}$$

It follows that for $i \geq 3$, $1 - p_i \geq (4/7) \cdot (1 - p_2) \geq 4/21$, and thus $p_i \leq 17/21$.

Consider now the number of rounds r from a non-stable critical round (when all vertices are white) until the next critical round. The probability that $r > k$ is lower and upper bounded by

$$1 - e^{-n2^{-k}} \leq 1 - (1 - 2^{-k})^n \leq n2^{-k}.$$

Clearly the upper bound holds also for any non-stable round.

Combining the above we obtain that: (i) from any given non-stable round, the probability that a stable round is reached in at most $k = \log n + 1$ rounds is at least $2/3 - n2^{-k} \geq 1/6$; (ii) from any

given non-stable critical round, the probability that the next critical round is non-stable and is reached in more than $k = \log n - 4$ rounds is at least $1 - 17/21 - e^{-n2^{-k}} > 1/6$; and (iii) assuming round $t = 0$ is not critical, the probability that the first critical round is non-stable is at least $1 - 17/21$. These statements, together, imply that the stabilization time is at least $k \log n$ with probability $2^{-\Theta(k)}$. And from that, the expectation and high-probability bounds follow. \square

Remark 10. From Theorem 9, it is immediate that the *expected* stabilization time of the 2-state MIS process is $\Theta(\log^2 n)$ on a graph G that is the disjoint union of \sqrt{n} cliques $K_{\sqrt{n}}$. The same bound holds also w.h.p.

Remark 11. A similar analysis as for Theorem 9 gives an upper bound of $O(\log n)$ on the stabilization time of the 3-state MIS process on K_n , both in expectation and w.h.p. The reason is that once $B_t \neq \emptyset$ then $B_{t'} \neq \emptyset$ for all $t' \geq t$, thus the next critical round is stable.

THEOREM 12. *The stabilization time of the 2-state MIS process on any graph $G = (V, E)$ of bounded arboricity (e.g., if G is a tree) is $O(\log n)$ w.h.p.*

PROOF. Recall that the arboricity λ of G is the minimum number of forests into which its edges can be partitioned, and is equal up to a factor of 2 to the maximum average degree in any subgraph [20]. Let d be the maximum average degree in any subgraph of G . Then $d \leq 2\lambda$. Let S_t be the subset of V_t consisting of all vertices $u \in V_t$ with $|N(u) \cap V_t| \leq d$. Then $|N(u) \cap V_t| \geq d+1$ for all $u \in V_t \setminus S_t$, thus we have $|V_t \setminus S_t| \cdot (d+1) \leq |V_t| \cdot d$, which gives $|S_t| \geq |V_t|/(d+1)$. If $u \in S_t \setminus A_t$ and $|N(u) \cap V_t| = d_u$, the probability that $N(u) \subseteq W_{t+1}$ is $2^{-d_u} \geq 2^{-d}$. Thus, for each $u \in S_t$, the probability that $u \in A_t \cup A_{t+1}$ is at least 2^{-d} . And if $u \in A_t \cup A_{t+1}$, Lemma 6 gives that $u \in I_{t+\log(d+1)+1}$ with probability at least $(2ed)^{-1}$. It follows

$$\begin{aligned} \mathbb{E} \left[|V_{t+\log(d+1)+1}| \mid |V_t| \right] &\leq |V_t| - (2ed)^{-1} \cdot 2^{-d} \cdot |V_t|/(d+1) \\ &\leq (1 - \epsilon) \cdot |V_t|, \end{aligned}$$

for some constant $\epsilon = \epsilon(d)$. Let $r = \log(d+1)+1$. Applying the above inequality iteratively k times, we obtain $\mathbb{E}[|V_{kr}|] \leq (1 - \epsilon)^k n \leq e^{-\epsilon k} n$. Thus for $k = 3\epsilon^{-1} \ln n$, $\mathbb{E}[|V_{rt}|] \leq n^{-2}$, and by Markov's inequality, $\mathbb{P}[|V_{rt}| \geq 1] \leq n^{-2}$, which implies the theorem. \square

THEOREM 13. *The stabilization time of the 2-state MIS process on any graph $G = (V, E)$ of maximum degree Δ is at most $O(\Delta \log n)$ w.h.p.*

PROOF. We will use the fact that if $u \in V_t$ then $N^+(u) \cap A_t \neq \emptyset$. Let $u \in V_0$, and let $(v_1, t_1), (v_2, t_2), (v_3, t_3), \dots$ be a random sequence of vertex-round pairs determined as follows. Let $t_0 = 0$. For $i \geq 1$, if $u \in V_{t_{i-1}}$ then v_i is an arbitrary vertex from the set $N^+(u) \cap A_{t_{i-1}}$ and $t_i = \min\{j > t_{i-1} : v_i \notin A_j\}$; while if $u \notin V_{t_{i-1}}$, then we let $(v_i, t_i) = (u, t_{i-1})$.

We focus on the first $r = 6e\Delta \ln n$ elements of the sequence above, and we bound the probability that $u \in V_{t_r}$. For each $1 \leq i \leq r$, if $u \in V_{t_{i-1}}$ then the conditional probability that $v_i \in I_{t_i}$ (and thus $u \notin V_{t_i}$), given v_i and $B_{t_{i-1}}$, is at least $1/(2e\Delta)$, from Lemma 6 and Remark 7. It follows that

$$\mathbb{P}[u \in V_{t_r}] \leq (1 - 1/(2e\Delta))^r \leq e^{-r/(2e\Delta)} = n^{-3}.$$

Next, we bound the value of t_r . For each $1 \leq i \leq r$ and $t \geq t_{i-1}$, if $v_i \in A_t$ then the conditional probability that $v_i \notin A_{t+1}$, given (v_i, t_i) and B_t , is exactly $1/2$ (in all cases). It follows that the probability of $t_r > 4r$ is upper bound by the probability that a sequence of $4r$ fair coin tosses contains fewer than r heads. Thus, by a Chernoff bound,

$$\mathbb{P}[t_r > 4r] \leq e^{-(1/2)^2 2r/2} = e^{-(3/2)\epsilon \Delta \ln n} < n^{-3}.$$

From the above results, we get

$$\mathbb{P}[u \notin V_{4r}] \geq \mathbb{P}[\{u \notin V_{t_r}\} \cap \{t_r \leq 4r\}] \geq 1 - 2n^{-3}.$$

Finally, a union bound over all $u \in V$ completes the proof. \square

4 THE 2-STATE MIS PROCESS ON RANDOM GRAPHS

We first show some additional properties of the 2-state MIS process, which hold for any graph but are useful only when adjacent vertices do not have many common neighbors. Then we show some structural properties of $G_{n,p}$ random graphs. Finally, we use these properties to show a poly($\log n$) upper bound on the stabilization time of the 2-state MIS process on $G_{n,p}$ random graphs.

4.1 Refined Properties of the 2-State MIS Process

We call a vertex k -active if it is active and has at most k active neighbors, and we let

$$A_t^k = \{u \in A_t : |N(u) \cap A_t| \leq k\}$$

be the set of k -active vertices at the end of round t . From Lemma 6, a k -active vertex has probability at least $\Omega(1/k)$ to become stable black in the next $O(\log k)$ rounds. It is thus desirable to have k -active vertices for small values k .

In this section we establish lower bounds on the probability that a given vertex u becomes k -active at some point during an interval of rounds, as a function of the probability that u becomes active (but has possibly more than k active neighbors) at some point during (roughly) the same interval. The value of k depends on various factors, but the main one is the number of common neighbors that u has with subsets of its neighbors.

The next key lemma is the base of all the other results in the section. It lower bounds the probability q of a white vertex u , which is non-active and non-stable, to become k -active after a single round. The lower bound is expressed in terms of the probability p that u is active white after two rounds. The value of k depends on the number of active neighbors of u , and the number of their common neighbors with u .

LEMMA 14. *Suppose that $u \in V_t \setminus A_t$,⁷ and let θ be the number of u 's neighbors that are active or adjacent to an active neighbor of u at the end of round t , i.e., $\theta = |N(u) \cap N^+(A_t \cap N(u))|$. Let p be the probability that $u \in A_{t+2} \cap W_{t+2}$, and q the probability that $u \in A_{t+1}^k$ where $k = \theta + \lceil \log(1/p) \rceil$. Then $q \geq p^\alpha$, where $\alpha = 1/\log(4/3) \leq 2.41$.*

PROOF. Let $D = N(u) \cap A_t$. In round $t+1$, each $v \in D$ updates its state to a random state, while each $v \in N(u) \setminus D$ remains white.

⁷Note that $u \in V_t \setminus A_t$ implies $u \in W_t \cap W_{t+1}$.

Let $Z = N(u) \cap A_{t+1} \setminus N^+(D)$ be the set of active neighbors of u at the end of round $t+1$ that are at distance at least two away from set D . Clearly, Z does not depend on the random choices of vertices $v \in D$ in round $t+1$.

We have that $u \in A_{t+1}$ if and only if all $v \in D$ update their state to white in round $t+1$, i.e., $\phi_{t+1}(v) = \text{white}$.⁸ This happens with probability $(1/2)^{|D|}$. Also

$$|N(u) \cap A_{t+1}| = |N(u) \cap A_{t+1} \cap N^+(D)| + |Z| \leq \theta + |Z|,$$

since $\theta = |N(u) \cap N^+(D)|$. It follows

$$q \geq (1/2)^d \cdot \mathbb{P}[|Z| \leq \lambda],$$

where $d = |D|$ and $\lambda = \lceil \log(1/p) \rceil$. Indeed, $|Z| \leq \lambda$ implies $|N(u) \cap A_{t+1}| \leq k$, since $k = \theta + \lambda$.

We have that $u \in A_{t+2} \cap W_{t+2}$ holds only if $\phi_{t+1}(v)$ or $\phi_{t+2}(v) = \text{white}$ for every $v \in D$, and $\phi_{t+2}(v) = \text{white}$ for every $v \in Z$. It follows that

$$p \leq (3/4)^d \cdot \sum_{i \geq 0} \mathbb{P}[|Z| = i] \cdot (1/2)^i. \quad (2)$$

Let $\epsilon = \mathbb{P}[|Z| \leq \lambda]$. Then

$$p \leq (3/4)^d \cdot \left(\epsilon + (1-\epsilon)(1/2)^{\lambda+1} \right) \leq (3/4)^d \cdot (\epsilon + (1-\epsilon)p/2),$$

since $\lambda = \lceil \log(1/p) \rceil$. This implies that $p \leq \epsilon + (1-\epsilon)p/2$, which gives $p \leq 2\epsilon/(1+\epsilon)$, and substituting that above we obtain

$$p \leq (3/4)^d \cdot (\epsilon + (1-\epsilon)\epsilon/(1+\epsilon)) = (3/4)^d \cdot \frac{2\epsilon}{1+\epsilon}.$$

Finally, since $(3/4)^{d\alpha} = (1/2)^d$, and for all $x \in [0, 1]$, it holds $\left(\frac{2x}{1+x}\right)^\alpha \leq \left(\frac{2x}{1+x}\right)^2 = x \cdot \frac{4x}{(1+x)^2} \leq x$, we get

$$p^\alpha \leq (3/4)^{d\alpha} \cdot \left(\frac{2\epsilon}{1+\epsilon}\right)^\alpha \leq (1/2)^d \cdot \epsilon \leq q. \quad \square$$

We use Lemma 14 to obtain a similar result over a sequence of r rounds. For any vertex $u \in V$ and $i \geq 1$, let

$$\theta_u(i) = \max\{|N(u) \cap N^+(S)| : S \subseteq N(u), |S| \leq i\}. \quad (3)$$

LEMMA 15. *Suppose that $u \in V_t \setminus A_t$, and let $d = |N(u) \cap A_t|$. Let $r \geq 2$, let p_r be the probability that $u \in A_{t+1} \cup \dots \cup A_{t+r}$, and let q_r be the probability that $u \in A_{t+1}^k \cup \dots \cup A_{t+r-1}^k$, where*

$$k = \theta_u \left(\alpha \log \left(\frac{4r}{p_r - 2^{-d}} \right) \right) + \left\lceil \log \left(\frac{4r}{p_r - 2^{-d}} \right) \right\rceil,$$

and $\alpha = 1/\log(4/3)$. Then $q_r \geq p_r^{1-\alpha} \cdot \left(\frac{p_r - 2^{-d}}{2}\right)^\alpha$.

Lemma 15 assumes that vertex u is initially non-active. The next lemma shows a similar result for the case where u is active initially. In this case, in place of the probability p_r that u becomes active at some point in the interval $\{t+1, \dots, t+r\}$, we use the probability b_r that u becomes black at some point of a subinterval $\{t+\ell, \dots, t+r\}$. The proof proceeds by considering the first round after t when either u has at most k black neighbors, or u is white. If the first condition holds, then u has probability $1/2$ of being black, and thus of being k -active. If only the second condition holds then we are in the case of Lemma 15.

⁸Recall the discussion about coin flips $\phi_t(v)$ at the beginning of Section 2.2.

LEMMA 16. *Suppose that $u \in A_t$. Let $\ell \geq 2$ and $r \geq \ell + 2$, let b_r be the probability that $u \in B_{t+\ell} \cup \dots \cup B_{t+r}$, and suppose that $b_r \geq 1/2^{\ell-2}$. Let q_r be the probability that $u \in A_t^k \cup \dots \cup A_{t+r-1}^k$, where*

$$k = \theta_u(\alpha \log(32r/b_r)) + \log(32r/b_r) + \log(1/b_r) + 3,$$

and $\alpha = 1/\log(4/3)$. Then $q_r \geq r^{1-\alpha} \cdot (b_r/16)^\alpha$.

In the last lemma of this section, we consider the case in which Lemma 15 does not give a large enough lower bound for q_r , even though p_r is large, because the difference $p_r - 2^{-d}$ is small. We proceed by essentially reducing this case to the setting of Lemma 16, after a single round.

LEMMA 17. *Suppose that $u \in V_t \setminus A_t$, and let $d = |N(u) \cap A_t|$. Let $\ell \geq 5$ and $r \geq \ell + 2$, let p_r be the probability that $u \in A_{t+1} \cup \dots \cup A_{t+r-1}$, let b_r be the probability that $u \in B_{t+\ell} \cup \dots \cup B_{t+r}$, and suppose that $b_r \geq \max\{1/2^{\ell-4}, 2(p_r - 2^{-d})\}$. Let q_r be the probability that $u \in A_t^k \cup \dots \cup A_{t+r-1}^k$, where*

$$k = \theta_u(\alpha \log(128r/b_r)) + \log(128r/b_r) + \log(4/b_r) + 3,$$

and $\alpha = 1/\log(4/3)$. Then $q_r \geq r^{1-\alpha} \cdot (b_r/64)^\alpha$.

The proofs of Lemmas 15 to 17 can be found in [14].

4.2 Structural Properties of $G_{n,p}$ and Good Graphs

We describe now some structural properties that a graph must possess in order for our analysis in the following sections to carry through. A graph satisfying these properties is called a *good graph*. We show that a random $G_{n,p}$ graph is a good graph w.h.p.

Definition 18 (Good Graphs). Let n be a positive integer and $0 < p < 1$. A graph $G = (V, E)$ with n vertices is (n, p) -good if it satisfies all the following properties:

- (P1) For any set $S \subseteq V$, the average degree of the induced subgraph $G[S]$ is at most $\max\{8p|S|, 4 \ln n\}$.
- (P2) For any set $S \subseteq V$ of size $|S| \geq 40 \ln(n)/p$,

$$|\{u \in V \setminus S : |N(u) \cap S| < p|S|/2\}| \leq |S|/2.$$

- (P3) For any three disjoint sets $S, T, I \subseteq V$ such that $|S| \geq 2|T|$ and $(S \cup T) \cap N(I) = \emptyset$,

$$|N(T) \setminus N^+(S \cup I)| \leq |N(S) \setminus N^+(I)| + 8 \ln^2(n)/p.$$

- (P4) For any two disjoint sets $S, T \subseteq V$ such that $|S| \geq |T|$ and $|T| \leq \ln(n)/p$, it holds $|E(S, T)| \leq 6|S| \ln n$.
- (P5) No two vertices $u, v \in V$ have more than $\max\{6np^2, 4 \ln n\}$ common neighbors.
- (P6) If $p \geq 2(\ln(n)/n)^{1/2}$ then $\text{diam}(G) \leq 2$.

We prove the following lemma in the full version of the paper [14].

LEMMA 19. *A random graph $G = (V, E)$ drawn from $G_{n,p}$ is (n, p) -good with probability $1 - O(n^{-2})$.*

4.3 Analysis of the 2-State MIS Process on $G_{n,p}$

In this section, we show the following bound on the stabilization time of the 2-state MIS process on $G_{n,p}$ graphs.

THEOREM 20. *The stabilization time of the 2-state MIS process on a random graph drawn from $G_{n,p}$, where $p = O(\sqrt{\log(n)/n})$ or $p = \Omega(1/\log^{2.5} n)$, is $O(\log^{5.5} n)$ with probability $1 - O(n^{-2})$.*

The theorem follows by combining Lemma 19 and the next lemma, which analyzes the 2-state MIS process on good graphs.

LEMMA 21. *The stabilization time of the 2-state MIS process on any (n, p) -good graph $G = (V, E)$, where $p = O(\sqrt{\log(n)/n})$ or $p = \Omega(1/\log^{2.5} n)$, is $O(\log^{5.5} n)$ with probability $1 - O(n^{-2})$.*

It is straightforward to extend the above statements so that $p \leq \text{poly}(\log n) \cdot n^{-1/2}$ or $p \geq 1/\text{poly}(\log n)$, for any desired $\text{poly}(\log n)$ factor, by adjusting the exponent of $\log n$ in the stabilization time bound.

Proof of Lemma 21. We show that starting from any vector of vertex states, the process makes sufficient progress after $\text{poly}(\log n)$ rounds, where progress is measured by the expected number of vertices that become stable. All lemmas below assume $G = (V, E)$ is an arbitrary (n, p) -good graph, and the probabilistic statements assume we are given the states of the vertices at the end of round t . The first lemma considers the case in which the number of active vertices is large, namely, $|A_t| = \Omega(\log(n)/p)$.

LEMMA 22. *If $|A_t| \geq 80 \ln(n)/p$ then there is a constant $\epsilon > 0$ such that $\mathbb{E}[|V_{t+\log n}|] \leq (1 - \epsilon) \cdot |V_t|$.*

PROOF. From property (P1) in Definition 18 of good graphs, the average degree of the induced subgraph $G[A_t]$ is at most

$$k = \max\{8p|A_t|, 4 \ln n\} = 8p|A_t|.$$

Let S be a subset of A_t consisting of the $|A_t|/2$ vertices $u \in A_t$ with the smallest degree in $G[A_t]$, i.e., for any two vertices $u \in S$ and $u' \in A_t \setminus S$, $|N(u) \cap A_t| \leq |N(u') \cap A_t|$. It follows that $|N(u) \cap A_t| \leq 2k$ for any $u \in S$, thus $S \subseteq A_t^{2k}$. Let $R = \{u \in V \setminus S : |N(u) \cap S| < p|S|/2\}$. Since $|S| = |A_t|/2 \geq 40 \ln(n)/p$, property (P2) in Definition 18 yields $|R| \leq |S|/2$. Then the number of vertices $u \in V_t$ with $|N(u) \cap S| \geq p|S|/2$ is at least

$$\begin{aligned} |V_t \setminus (S \cup R)| &\geq |V_t| - (|S| + |R|) \geq |V_t| - 3|S|/2 \\ &= |V_t| - 3|A_t|/4 \geq |V_t|/4. \end{aligned}$$

Since each of those vertices u has at least $p|S|/2$ neighbors in $S \subseteq A_t^{2k}$, Lemma 8 gives that the probability at least one neighbor of u is stable black (and thus u is also stable) at the end of round $t + \log n$ is at least

$$(1/5) \cdot \min\left\{1, \frac{p|S|/2}{4k}\right\} = (1/5) \cdot \min\left\{1, \frac{p|A_t|/4}{32p|A_t|}\right\} = 1/640.$$

Then the expected number of vertices that are not stable at the end of round $t + \log n$ is

$$\mathbb{E}[|V_{t+\log n}|] \leq |V_t| - (|V_t|/4) \cdot 1/640 \leq |V_t| - |V_t|/2560. \quad \square$$

The next lemma considers the case in which the number of vertices that are not stable is large, namely $|V_t| = \Omega(\ln^2(n)/p)$, and $|A_t| = O(\ln(n)/p)$.

LEMMA 23. *If $|V_t| \geq 10 \ln^2(n)/p$ and $|A_t| \leq 80 \ln(n)/p$ then there is a constant $\epsilon > 0$ such that $\mathbb{E}[|V_{t+\log n}|] \leq (1 - \epsilon/\ln n) \cdot |V_t|$.*

PROOF. From property (P1) in Definition 18, the average degree of graph $G[A_t]$ is at most

$$k = \max\{8p|A_t|, 4 \ln n\} \leq 640 \ln n.$$

Let S be a subset of A_t consisting of the $2|A_t|/3$ vertices $u \in A_t$ with the smallest degree in $G[A_t]$, and let $T = A_t \setminus S$. Then $|N(u) \cap A_t| \leq 3k$ for any $u \in S$, thus $S \subseteq A_t^{3k}$.

We observe that the set V_t consist of (i) all the active vertices, $u \in A_t = S \cup T$, and (ii) all the non-active vertices that are not contained in $N^+(I_t)$ (these vertices are white and have at least one active neighbor). We can thus partition V_t into the four distinct sets: S , $N(S) \setminus N(I_t)$, $T \setminus N(S)$, and $N(T) \setminus N^+(S \cup I_t)$. For the sizes of these sets, we have $|T \setminus N(S)| \leq |T| < |S|$ and, by property (P3) in Definition 18,

$$|N(T) \setminus N^+(S \cup I_t)| \leq |N(S) \setminus N(I_t)| + 8 \ln^2(n)/p.$$

Combining these two inequalities, and using the fact that the sizes of the four sets above sum to $|V_t|$ and that $|V_t| \geq 10 \ln^2(n)/p$, we obtain

$$|S| + |N(S) \setminus N(I_t)| \geq (|V_t| - 8 \ln^2(n)/p)/2 \geq |V_t|/10.$$

Therefore, at least $|V_t|/10$ vertices $u \in V_t$ are in S or adjacent to a vertex from S . From Lemma 6, each $u \in S \subseteq A_t^{3k}$ is stable black (and all its neighbors are stable white) at the end of round $t + \log n$, with probability at least $1/(6ek)$. It follows that

$$\begin{aligned} \mathbb{E}[|V_{t+\log n}|] &\leq |V_t| - (|V_t|/10) \cdot 1/(6ek) \\ &\leq |V_t| - |V_t|/(1.1 \cdot 10^5 \ln n). \quad \square \end{aligned}$$

In the next lemma we analyze the remaining case, in which $|V_t| = O(\ln^2(n)/p)$ and $|A_t| = O(\ln(n)/p)$. In fact, the lemma does not require a bound on $|A_t|$. Unlike the previous lemmas, however, it requires that $p = O(\sqrt{\log(n)/n})$.

LEMMA 24. *If $|V_t| \leq 10 \ln^2(n)/p$ and $p \leq c\sqrt{\ln(n)/n}$, for some constant $c > 0$, then there is a constant $\epsilon = \epsilon(c) > 0$ such that $\mathbb{E}[|V_{t+2 \log n}|] \leq (1 - \epsilon/\ln^{3.5} n) \cdot |V_t|$.*⁹

PROOF. From property (P1) in Definition 18, the average degree of graph $G[V_t]$ is at most

$$k = \max\{8p|V_t|, 4 \ln n\} \leq 80 \ln^2 n.$$

Let T be a subset of V_t consisting of the $\min\{\ln(n)/p, |V_t|/2\}$ vertices $u \in V_t$ with the largest degree in $G[V_t]$, and let $S = V_t \setminus T$. Then, for any $u \in S$, its degree $|N(u) \cap V_t|$ in $G[V_t]$ is at most

$$d = k|V_t|/|T| \leq k \cdot \max\{p|V_t|/\ln n, 2\} \leq 800 \ln^3 n,$$

where in the first inequality we used that $|T| = \min\{\ln(n)/p, |V_t|/2\}$. From property (P4) in Definition 18, we have $|E(S, T)| \leq 6|S| \ln n$, thus the average number of neighbors in T of a vertex $u \in S$ is at most $6 \ln n$. Let

$$R = \{u \in S : |N(u) \cap T| \leq 12 \ln n\}.$$

Then $|R| \geq |S|/2 \geq |V_t|/4$, since $|S| \geq |T|$. We will show that there is a constant $\epsilon' = \epsilon'(c)$ such that for every $u \in R$,

$$\mathbb{P}[u \notin V_{t+2 \log n}] \geq \epsilon' \ln^{-\alpha-1} n \cdot (\ln \ln n)^{-\alpha}. \quad (4)$$

From this and $|R| \geq |V_t|/4$, it follows that

$$\mathbb{E}[|V_{t+2 \log n}|] \leq |V_t| - |R| \cdot \epsilon' \ln^{-\alpha-1} n \cdot (\ln \ln n)^{-\alpha}.$$

Since $\alpha = 1/\log(4/3) \leq 2.41$, the above implies the lemma. To complete the proof it remains to show (4).

Let $u \in R$. We partition the neighbors of u in $G[V_t]$ into sets $N(u) \cap S$ and $N(u) \cap T$, and define

$$x = \mathbb{P}[N(u) \cap S \cap A \neq \emptyset], \quad y = \mathbb{P}[N(u) \cap T \cap B \neq \emptyset],$$

where $A = A_t \cup \dots \cup A_{t+r-2}$, $B = B_{t+r-2} \cup B_{t+r-1} \cup B_{t+r}$, and $r = \log(48 \ln n) + 6$. We distinguish three cases: $x + y \leq 1/2$, $x \geq 1/4$, and $y \geq 1/4$.

Case $x + y \leq 1/2$: With probability at least $1 - (x + y) \geq 1/2$, we have $N(u) \cap S \cap A = \emptyset$ and $N(u) \cap T \cap B = \emptyset$. If $N(u) \cap S \cap A = \emptyset$ then $N(u) \cap S \subseteq W_{t+r-2}$ (it is easy to see that $N(u) \cap S \cap I_{t+r-2} = \emptyset$, since a vertex must be active before it becomes black). Also, if $N(u) \cap T \cap B = \emptyset$ then $N(u) \cap T \subseteq W_{t+r-2}$. Thus, with probability at least $1/2$, we have that $N(u) \subseteq W_{t+r-2}$. If $N(u) \subseteq W_{t+r-2}$, then either $u \in A_{t+r-2} \cap W_{t-r-2}$ or $u \in I_{t+r-2}$. Therefore, with probability at least $1/2$, either $u \notin V_{t+r-2}$ or $u \in A_{t+r-2}$. And if $u \in A_{t+r-2}$, then $u \in A_{t+r-2}^d$ since $|N(u) \cap V_t| \leq d$, and from Lemma 6, the probability that $u \in I_{t+r-2+\log n}$ is at least $(2ed)^{-1}$. From the last two statements, the probability that $u \notin V_{t+r-2+\log n}$ is at least $(1/2) \cdot (2ed)^{-1} \geq (8700 \ln^3 n)^{-1}$, which implies (4).

Case $x \geq 1/4$: With probability $x \geq 1/4$, there is a pair v, j such that $v \in N(u) \cap S$, $0 \leq j \leq r - 2$, and $v \in A_{t+j}$. And if such v, j exists, then $v \in A_{t+j}^d$ since $|N(v) \cap V_t| \leq d$, and from Lemma 6, the probability that $v \in I_{t+j+\log n}$ is at least $(2ed)^{-1}$. We conclude that the probability that $u \in N^+(I_{t+r-2+\log n})$ is at least $(1/4) \cdot (2ed)^{-1} \geq (17400 \ln^3 n)^{-1}$, which implies (4).

Case $y \geq 1/4$: There exists a $v^* \in N(u) \cap T$ such that

$$\mathbb{P}[v^* \in B] \geq y/|N(u) \cap T| \geq (4 \cdot 12 \ln n)^{-1} = (48 \ln n)^{-1}.$$

If $v^* \in A_t$ then we apply Lemma 16, for $\ell = r - 2 \geq \log(48 \ln n) + 2$ and $b_r \geq (48 \ln n)^{-1}$, to obtain that $v^* \in A_t^\lambda \cup \dots \cup A_{t+r-1}^\lambda$ with probability at least $q = r^{1-\alpha} \cdot (16 \cdot 48 \ln n)^{-\alpha}$, where

$$\lambda = \theta_{v^*}(\alpha \log(32r \cdot 48 \ln n)) + O(\log(r \ln n)).$$

Suppose now that $v^* \in V_t \setminus A_t$, and let

$$p^* = \mathbb{P}[v^* \in A_{t+1} \cup \dots \cup A_{t+r-1}] - 2^{-|N(v^*) \cap A_t|}.$$

If $p^* \geq \mathbb{P}[v^* \in B]/2 \geq (96 \ln n)^{-1}$, then we apply Lemma 15 (using $r - 1$ in place of r), to obtain that $v^* \in A_t^{\lambda'} \cup \dots \cup A_{t+r-2}^{\lambda'}$ with probability at least $q' = r^{1-\alpha} \cdot (p^*/2)^\alpha \geq r^{1-\alpha} \cdot (192 \ln n)^{-\alpha}$, where

$$\lambda' = \theta_{v^*}(\alpha \log(4r \cdot 96 \ln n)) + O(\log(r \ln n)).$$

If $p^* < \mathbb{P}[v^* \in B]/2$, then we apply Lemma 17, for $\ell = r - 2 \geq \log(48 \ln n) + 4$ and $b_r = \mathbb{P}[v^* \in B] \geq (48 \ln n)^{-1}$, to obtain that $v^* \in A_t^{\lambda''} \cup \dots \cup A_{t+r-1}^{\lambda''}$ with probability at least $q'' = r^{1-\alpha} \cdot (64 \cdot 48 \ln n)^{-\alpha}$, where

$$\lambda'' = \theta_{v^*}(\alpha \log(128r \cdot 48 \ln n)) + O(\log(r \ln n)).$$

⁹If c is super constant then the proof gives $\mathbb{E}[|V_{t+2 \log n}|] \leq (1 - \epsilon/(c^2 \ln^{3.5} n)) \cdot |V_t|$.

We observe that $\min\{q, q', q''\} \geq \epsilon \ln^{-\alpha} n \cdot (\ln \ln n)^{1-\alpha}$, for some constant $\epsilon > 0$. Also, $\max\{\lambda, \lambda', \lambda''\} \leq \beta \ln n \cdot \ln \ln n$, for some constant $\beta = \beta(c) > 0$, because property (P5) in Definition 18 and the assumption that $p \leq c\sqrt{\ln(n)}/n$ imply that for any vertex $v \in V$, $\theta_v(i) \leq i \cdot (6c^2 + 4) \ln n$ (recall the definition of θ_v from (3)). Therefore, the probability that v^* is $(\beta \ln n \cdot \ln \ln n)$ -active at the end of some round in $\{t, \dots, t+r-1\}$ is at least $\epsilon \ln^{-\alpha} n \cdot (\ln \ln n)^{1-\alpha}$, and from Lemma 6, the probability that $v^* \in I_{t+r-1+\log n}$ is at least $\epsilon \ln^{-\alpha} n \cdot (\ln \ln n)^{1-\alpha} \cdot (2e\beta \ln n \cdot \ln \ln n)^{-1}$. Thus with at least that probability we have $u \notin V_{t+r-1+\log n}$. This completes the proof of (4), and the proof of Lemma 24. \square

Putting the Pieces Together. First, suppose that $p \leq c\sqrt{\ln(n)}/n$ for some constant $c > 0$. From Lemmas 22 to 24, for any $t \geq 0$, $\mathbb{E}[|V_{t+2\log n}|] \leq (1 - \epsilon/\ln^{3.5} n) \cdot \mathbb{E}[|V_t|]$. Iteratively applying this inequality, we obtain that for any $i \geq 0$,

$$\mathbb{E}[|V_{2i\log n}|] \leq \left(1 - \epsilon/\ln^{3.5} n\right)^i \cdot n.$$

Substituting $i = 3 \ln^{4.5} n / \epsilon$ yields $\mathbb{E}[|V_{(6/\epsilon)\log n \cdot \ln^{4.5} n}|] \leq n^{-2}$, and by Markov's inequality, $\mathbb{P}[|V_{(6/\epsilon)\log n \cdot \ln^{4.5} n}| \geq 1] \leq n^{-2}$.

If $p \geq \epsilon/\ln^{2.5} n$ for some constant $\epsilon > 0$, then we use Lemmas 22 and 23 as above to obtain that $\mathbb{P}[|V_t| \geq 10 \ln^2(n)/p] \leq n^{-2}$ for some $t = O(\log^3 n)$. We also observe that if $|V_t| < 10 \ln^2(n)/p$ then the maximum degree of graph $(V, E(V_t))$ is $\Delta < |V_t| \leq 10 \ln^2(n)/p \leq 10 \ln^{4.5}(n)/\epsilon$, and Theorem 13 yields a bound of $O(\Delta \log n) = O(\log^{5.5} n)$. Combining these two completes the proof of this case, and of Lemma 21.

Remark 25. Some of the logarithmic factors can be shaved off with a more careful analysis. For example, using a ‘‘pipelining’’ argument, one could improve the bound on halving $|V_t|$ obtained from Lemma 24, from $O(\log n \cdot \ln^{3.5} n)$ to $O(\log n + \ln^{3.5} n)$, thus saving one logarithmic factor.

5 LOGARITHMIC SWITCH AND THE 3-COLOR MIS PROCESS

We present an extension of the 2-state MIS process, called *3-color MIS process*, which uses one additional color, *gray*, and includes also a sub-process, called *logarithmic switch*, which runs in parallel to the main process. We analyze the 3-color MIS process on $G_{n,p}$ random graphs.

5.1 The Logarithmic Switch Process

A *logarithmic switch* process associates a binary on/off value with each vertex u in each round t . We first introduce an abstract logarithmic switch process, by specifying the properties that the sequence of on/off values must satisfy at each vertex. Then we describe an actual randomized graph process that satisfies these properties with high probability and in a self-stabilizing manner, using 6 states per vertex.

Definition 26 (Logarithmic Switch Process). An (a, b) -logarithmic switch process on graph $G = (V, E)$ generates a binary sequence $\sigma_0(u), \sigma_1(u), \dots$ for each vertex $u \in V$, where $\sigma_t(u) \in \{\text{on}, \text{off}\}$ for each $t \geq 0$, such that the following properties hold for all $u \in V$:

- (S1) Every run of consecutive off values in sequence $\sigma_0(u), \sigma_1(u), \dots$ has length at most $a \ln n$.
- (S2) If $\text{diam}(G) \leq 2$ then every run of consecutive off values in sequence $\sigma_t(u), \sigma_{t+1}(u), \dots$ has length at least $\frac{a}{6} \ln n$, where $t = \min\{i \geq \frac{a}{6} \ln n : \sigma_i(u) = \text{on}\}$.
- (S3) If $\text{diam}(G) \leq 2$ then every run of consecutive on values in sequence $\sigma_t(u), \sigma_{t+1}(u), \dots$ has length at most b , where t is some constant independent of n .

Definition 27 (Randomized Logarithmic Switch). In the *randomized logarithmic switch process* on graph $G = (V, E)$, each vertex $u \in V$ has a state, called *level*, that takes on values in the set $\{0, 1, \dots, 5\}$. The initial level of u , $\text{level}_0(u)$, can be arbitrary, and in each round $t \geq 1$, u 's level is updated according to the following rule, which uses a global parameter $0 < \zeta < 1$:

```

if  $\text{level}_{t-1}(u) = 5$  then
  | choose a random bit  $b_t(u)$  such that  $\mathbb{P}[b_t(u) = 0] = \zeta$ 
end
if  $(\text{level}_{t-1}(u) = 5$  and  $b_t(u) = 1)$  or  $\text{level}_{t-1}(u) = 0$  then
  | set  $\text{level}_t(u) = 5$ 
else set  $\text{level}_t(u) = \max\{\text{level}_{t-1}(v) : v \in N^+(u)\} - 1$ 

```

We have the following mapping of levels to the binary on/off values of Definition 26. For each $u \in V$ and $t \geq 0$,

$$\sigma_t(u) = \begin{cases} \text{on} & \text{if } \text{level}_t(u) \leq 2 \\ \text{off} & \text{if } \text{level}_t(u) \geq 3. \end{cases}$$

LEMMA 28. For any graph $G = (V, E)$, the randomized logarithmic switch process with parameter $0 < \zeta \leq 1/2$ satisfies the properties (S1) to (S3) for $a = 4/\zeta$ and $b = 3$, with probability $1 - O(n^{-2})$, in the first n rounds.

PROOF. Let $u \in V$, and let $S_u \subseteq V$ be the set of vertices at distance at most 2 from u . If u has level at least 3 in all rounds $t, \dots, t + a \ln n$, then no vertex $v \in S_u$ has level 0 in rounds $t + 2, \dots, t + a \ln n$; and thus at least one vertex $v \in S_u$ must be at level 5 in all rounds $t + 2, \dots, t + a \ln n - 2$. It follows that the probability there is some $u \in V$ and round $t \leq n$ such that u has level at least 3 in all rounds $t, \dots, t + a \ln n$ is at most $n^2(1-\zeta)^{a \ln n - 4} \leq n^{2-a\zeta}/(1-\zeta)^4 \leq 16n^{-2}$, since $a\zeta = 4$ and $\zeta \leq 2$. Thus, property (S1) holds with probability at least $1 - O(n^{-2})$.

The rest of the proof is similar to that in [10], and can be found in the full version of the paper [14]. \square

5.2 The 3-Color MIS Process

In the 3-state MIS process we have three colors, black, white and gray. Now, a black vertex becomes gray instead of white, a gray vertex becomes white after a while, and other vertices treat gray vertices as white. The transition from gray to white is controlled by a logarithmic switch process.

Definition 29 (3-Color MIS Process). The 3-color MIS process consists of two sub-processes that run in parallel on graph $G = (V, E)$. The first is an $(a, 3)$ -logarithmic switch process, where $a = 512$, which generates a value $\sigma_t(u) \in \{\text{on}, \text{off}\}$ for each vertex

$u \in V$ in each round $t \geq 0$. The second is a variant of the 2-state MIS process, where each vertex $u \in V$ has a state $c_t(u) \in \{\text{black, white, gray}\}$, $c_0(u)$ can be arbitrary, and in each round $t \geq 1$, u 's state is updated as follows:

```

let  $NC_t(u) = \{c_{t-1}(v) : v \in N(u)\}$ 
if  $c_{t-1}(u) = \text{black}$  and  $NC_t(u) \ni \text{black}$  then
  | let  $c_t(u)$  be a uniformly random state from
  |    $\{\text{black, gray}\}$ 
else if  $c_{t-1}(u) = \text{white}$  and  $NC_t(u) \not\ni \text{black}$  then
  | let  $c_t(u)$  be a uniformly random state from
  |    $\{\text{black, white}\}$ 
else if  $c_{t-1}(u) = \text{gray}$  and  $\sigma_{t-1}(u) = \text{on}$  then
  | set  $c_t(u) = \text{white}$ 
else set  $c_t(u) = c_{t-1}(u)$ 

```

There are two differences in the update rule above compared to that for the 2-state MIS process: a black vertex with a black neighbor changes to gray with probability $1/2$, rather than to white; and a gray vertex changes to white if its switch value is on. Other than that, gray vertices are treated similarly to non-active white vertices.

We use the same definitions and notations as in the 2-state MIS process, except that now a vertex is *stable* if it is black and has no black neighbors, or it is *not black* and has some stable black neighbor. Also, $\Gamma_t = V \setminus (B_t \cup W_t)$ is the set of gray vertices at the end of round t .

The definition of the 3-color MIS process above assumes an arbitrary logarithmic switch process. We can use the randomized logarithmic switch from Definition 27, which uses 6 states per vertex, to obtain a 3-color MIS process that uses $6 \cdot 3 = 18$ states in total. From Lemma 28, it suffices to choose $\zeta = 4/a = 2^{-7}$ to ensure that the randomized logarithmic switch is an $(a, 3)$ -logarithmic switch w.h.p. Then at most 7 random bits are required per round for each vertex (plus one more for each active vertex).

We note that Lemmas 6 and 8 and their proofs carry over to the 3-color MIS process, without changes. The two simple lemmas below are specific to the 3-color MIS process. Recall that $a = 512$ is a parameter of the logarithmic switch.

LEMMA 30. *If $t \geq a \ln n$ and $u \in \Gamma_t$ then $u \in A_{t-a \ln n} \cup \dots \cup A_{t-1}$.*

PROOF. By property (S1) in Definition 26 of the logarithmic switch, a vertex is gray for at most $a \ln n$ consecutive rounds. Also if a vertex becomes gray in round $j > 0$, it must be active black at the end of round $j - 1$. Combining these two facts implies the lemma. \square

LEMMA 31. *If $\text{diam}(G) \leq 2$, $t \geq \frac{a}{6} \ln n$, and $t' = t + \frac{a}{6} \ln n$, then the expected number of times that any given vertex $u \in V$ is active black between rounds t and t' is*

$$\mathbb{E} \left[\left| \{j : u \in B_j \cap V_j\} \cap \{t, \dots, t'\} \right| \mid B_t, W_t \right] \leq 4.$$

PROOF. From properties (S2) and (S3) in Definition 26, it follows that sequence $c_t(u), \dots, c_{t'}(u)$ contains at most two runs of consecutive black states. Moreover, the expected length of the prefix of each black run before u becomes stable black or the run finishes (and u becomes gray) is at most 2. It follows that u is non-stable black in at most 4 rounds in expectation. \square

Lemmas 14 and 15 hold also for the 3-color MIS process, when $u \in V_t \setminus (A_t \cup \Gamma_t)$.¹⁰ Finally, we will use the next simple lemma, together with Lemma 15.

LEMMA 32. *Let $u \in V$, $t \geq 0$, and $d > 0$. Let $t' \geq t$ be the first round when either u is white and has at least d black neighbors, or u is stable. Then the expected number of rounds $t < j < t'$ at which u is black and has at least d black neighbors is at most 3.*

PROOF. The lemma is obtained using the observations that: each time u 's state changes from white to black, it is equally likely that it remained white; and, when u is active black, it becomes gray in the next step with probability $1/2$. \square

5.3 Analysis of the 3-Color MIS Process on $G_{n,p}$

Unlike the 2-state MIS process, for which we succeeded in showing a poly($\log n$) bound just for a certain range of values for p , we show that the stabilization time of the 3-color MIS process is poly($\log n$) on $G_{n,p}$ random graphs for the complete range of values $0 \leq p \leq 1$.

THEOREM 33. *The stabilization time of the 3-color MIS process on a random graph drawn from $G_{n,p}$ is $O(\log^6 n)$ with probability $1 - O(n^{-2})$.*

As before, it suffices to show that the above bound holds for good graphs, and apply Lemma 19.

LEMMA 34. *The stabilization time of the 3-color MIS process on any (n, p) -good graph $G = (V, E)$ is $O(\log^6 n)$ with probability $1 - O(n^{-2})$.*

The proof of Lemma 34 is similar to that of Lemma 21. We show that sufficient progress is made in expectation after poly($\log n$) rounds, from any initial vertex states. The main difference is that now we show that this is also true even in the case of $|V_t| = O(\log^2(n)/p)$ when $\text{diam}(G) \leq 2$, which corresponds to the case of $p = \Omega(\sqrt{\log(n)/n})$, by property (P6) in Definition 18. This is precisely the case that we could not handle in the analysis of the 2 state MIS process. The complete proof of Lemma 34 can be found in the full version of the paper [14].

ACKNOWLEDGMENTS

Support is gratefully acknowledged from the Agence Nationale de la Recherche (ANR) under project ByBloS (ANR-20-CE25-0002), and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 834861).

REFERENCES

- [1] Yehuda Afek, Noga Alon, Ziv Bar-Joseph, Alejandro Cornejo, Bernhard Haeupler, and Fabian Kuhn. 2013. Beeping a maximal independent set. *Distributed Comput.* 26, 4 (2013), 195–208. <https://doi.org/10.1007/s00446-012-0175-7>
- [2] Yehuda Afek, Noga Alon, Omer Barad, Eran Hornstein, Naama Barkai, and Ziv Bar-Joseph. 2011. A biological solution to a fundamental distributed computing problem. *Science* 331, 6014 (January 2011), 183–185. <https://doi.org/10.1126/science.1193210>
- [3] Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikael Rabie, and Jukka Suomela. 2021. Lower Bounds for Maximal Matchings and Maximal Independent Sets. *J. ACM* 68, 5 (2021), 39:1–39:30. <https://doi.org/10.1145/3461458>

¹⁰The proofs require just minor modifications, mostly replacing some occurrences of “white” by “not black” or “gray”.

- [4] Leonid Barenboim, Michael Elkin, and Uri Goldenberg. 2022. Locally-Iterative Distributed $(\Delta + 1)$ -coloring and Applications. *J. ACM* 69, 1 (2022), 5:1–5:26. <https://doi.org/10.1145/3486625>
- [5] Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. 2016. The Locality of Distributed Symmetry Breaking. *J. ACM* 63, 3 (2016), 20:1–20:45. <https://doi.org/10.1145/2903137>
- [6] Stephen A. Cook. 1983. An Overview of Computational Complexity. *Commun. ACM* 26, 6 (1983), 400–408. <https://doi.org/10.1145/358141.358144>
- [7] Alejandro Cornejo and Fabian Kuhn. 2010. Deploying Wireless Networks with Beeps. In *Proc. Distributed Computing, 24th International Symposium, DISC*. 148–162. https://doi.org/10.1007/978-3-642-15763-9_15
- [8] Edsger W. Dijkstra. 1974. Self-stabilizing Systems in Spite of Distributed Control. *Commun. ACM* 17, 11 (1974), 643–644. <https://doi.org/10.1145/361179.361202>
- [9] Shlomi Dolev. 2000. *Self-Stabilization*. MIT Press.
- [10] Yuval Emek and Eyal Keren. 2021. A Thin Self-Stabilizing Asynchronous Unison Algorithm with Applications to Fault Tolerant Biological Networks. In *Proc. ACM Symposium on Principles of Distributed Computing, PODC*. 93–102. <https://doi.org/10.1145/3465084.3467922>
- [11] Yuval Emek and Roger Wattenhofer. 2013. Stone age distributed computing. In *Proc. ACM Symposium on Principles of Distributed Computing, PODC*. 137–146. <https://doi.org/10.1145/2484239.2484244>
- [12] Mohsen Ghaffari. 2017. Distributed MIS via All-to-All Communication. In *Proc. ACM Symposium on Principles of Distributed Computing, PODC*. 141–149. <https://doi.org/10.1145/3087801.3087830>
- [13] Mohsen Ghaffari, Christoph Grunau, and Václav Rozhon. 2021. Improved Deterministic Network Decomposition. In *Proc. ACM-SIAM Symposium on Discrete Algorithms, SODA*. 2904–2923. <https://doi.org/10.1137/1.9781611976465.173>
- [14] George Giakkoupis and Isabella Ziccardi. 2023. Distributed Self-Stabilizing MIS with Few States and Weak Communication. *CoRR* abs/2301.05059 (2023). <https://doi.org/10.48550/arXiv.2301.05059> arXiv:2301.05059 Full version of this paper.
- [15] Wayne Goddard, Stephen T. Hedetniemi, David Pokrass Jacobs, and Pradip K. Srimani. 2003. Self-Stabilizing Protocols for Maximal Matching and Maximal Independent Sets for Ad Hoc Networks. In *Proc. 17th International Parallel and Distributed Processing Symposium (IPDPS 2003)*. 162. <https://doi.org/10.1109/IPDPS.2003.1213302>
- [16] S.M. Hedetniemi, S.T. Hedetniemi, D.P. Jacobs, and P.K. Srimani. 2003. Self-stabilizing algorithms for minimal dominating sets and maximal independent sets. *Computers & Mathematics with Applications* 46, 5 (2003), 805–811. [https://doi.org/10.1016/S0898-1221\(03\)90143-X](https://doi.org/10.1016/S0898-1221(03)90143-X)
- [17] Michiyo Ikeda, Sayaka Kamei, and Hirotsugu Kakugawa. 2002. A space-optimal self-stabilizing algorithm for the maximal independent set problem. In *Proc. 3rd International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT*. 70–74.
- [18] Peter Jeavons, Alex Scott, and Lei Xu. 2016. Feedback from nature: Simple randomised distributed algorithms for maximal independent set selection and greedy colouring. *Distributed Comput.* 29, 5 (2016), 377–393. <https://doi.org/10.1007/s00446-016-0269-8>
- [19] Michael Luby. 1986. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM J. Comput.* 15, 4 (1986), 1036–1053. <https://doi.org/10.1137/0215074>
- [20] C. St.J. A. Nash-Williams. 1964. Decomposition of Finite Graphs Into Forests. *J. Lond. Math. Soc.* s1-39, 1 (1964), 12–12. <https://doi.org/10.1112/jlms/s1-39.1.12>
- [21] Sandeep K. Shukla, Daniel J. Rosenkrantz, and Sekharipuram S. Ravi. 1995. Observations on self-stabilizing graph algorithms for anonymous networks. In *Proc. 2nd Workshop on Self-Stabilizing Systems, SSS*. 15.
- [22] Volker Turau. 2007. Linear self-stabilizing algorithms for the independent and dominating set problems using an unfair distributed scheduler. *Inf. Process. Lett.* 103, 3 (2007), 88–93. <https://doi.org/10.1016/j.ipl.2007.02.013>
- [23] Volker Turau. 2019. Making Randomized Algorithms Self-stabilizing. In *Proc. Structural Information and Communication Complexity - 26th International Colloquium, SIROCCO*. 309–324. https://doi.org/10.1007/978-3-030-24922-9_21
- [24] Volker Turau and Christoph Weyer. 2006. Randomized Self-stabilizing Algorithms for Wireless Sensor Networks. In *Proc. Self-Organizing Systems, First International Workshop, IWSOS, and Third International Workshop on New Trends in Network Architectures and Services, EuroNGI*. 74–89. https://doi.org/10.1007/11822035_8
- [25] Leslie G. Valiant. 1982. Parallel computation. In *Proc. 7th IBM Symposium on Mathematical Foundations of Computer Science*.