

# Near-Optimal Leader Election in Population Protocols on Graphs

Dan Alistarh  
dan.alistarh@ist.ac.at  
IST Austria  
Klosterneuburg, Austria

Joel Rybicki  
joel.rybicki@ist.ac.at  
IST Austria  
Klosterneuburg, Austria

Sasha Voitovych  
sasha.voitovych@mail.utoronto.ca  
University of Toronto  
Toronto, Canada

## ABSTRACT

In the *stochastic population protocol* model, we are given a connected graph with  $n$  nodes, and in every time step, a scheduler samples an edge of the graph uniformly at random and the nodes connected by this edge interact. A fundamental task in this model is *stable leader election*, in which all nodes start in an identical state and the aim is to reach a configuration in which (1) exactly one node is elected as leader and (2) this node remains as the unique leader no matter what sequence of interactions follows. On *cliques*, the complexity of this problem has recently been settled: time-optimal protocols stabilize in  $\Theta(n \log n)$  expected steps using  $\Theta(\log \log n)$  states, whereas protocols that use  $O(1)$  states require  $\Theta(n^2)$  expected steps.

In this work, we investigate the complexity of stable leader election on general graphs. We provide the first non-trivial time lower bounds for leader election on general graphs, showing that, when moving beyond cliques, the complexity landscape of leader election becomes very diverse: the time required to elect a leader can range from  $O(1)$  to  $\Theta(n^3)$  expected steps. On the upper bound side, we first observe that there exists a protocol that is time-optimal on many graph families, but uses polynomially-many states. In contrast, we give a near-time-optimal protocol that uses only  $O(\log^2 n)$  states that is at most a factor  $\log n$  slower. Finally, we show that the constant-state protocol of Beauquier et al. [OPODIS 2013] is at most a factor  $n \log n$  slower than the fast polynomial-state protocol. Moreover, among constant-state protocols, this protocol has near-optimal *average case complexity* on dense random graphs.

## CCS CONCEPTS

• **Theory of computation** → **Distributed algorithms; Randomness, geometry and discrete structures.**

## KEYWORDS

randomized algorithms, leader election, population protocols, graph algorithms, lower bounds

## ACM Reference Format:

Dan Alistarh, Joel Rybicki, and Sasha Voitovych. 2022. Near-Optimal Leader Election in Population Protocols on Graphs. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing (PODC '22)*, July 25–29, 2022, Salerno, Italy. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3519270.3538435>



This work is licensed under a Creative Commons Attribution International 4.0 License.

PODC '22, July 25–29, 2022, Salerno, Italy.

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9262-4/22/07.

<https://doi.org/10.1145/3519270.3538435>

## 1 INTRODUCTION

Leader election is one of the most fundamental symmetry-breaking problems in distributed computing [8]: given a distributed system consisting of  $n$  identical nodes, the goal is to designate exactly one node as a leader and all others as followers. In this work, we study the computational complexity of leader election in the *stochastic population protocol model*, a popular model of distributed computation among a population of (initially) indistinguishable agents that reside on a graph and interact in a random manner [10, 13].

### 1.1 The stochastic population model on graphs

In the stochastic population protocol model, or simply the *population model*, the system is described by a finite, connected graph  $G = (V, E)$  with  $n$  nodes. Each node represents an agent, corresponding to a finite state automaton. Initially, all nodes are identical and anonymous. Computation proceeds asynchronously, in a series of random pairwise interactions between neighbouring nodes. In each discrete time step, the following happen:

- (1) the scheduler samples an ordered pair  $(u, v)$  *uniformly at random* among all pairs of nodes connected by an edge,
- (2) the selected nodes  $u$  and  $v$  interact by exchanging information and updating their local states, and
- (3) every node maps their local state to an output value.

When the scheduler selects the ordered pair  $(u, v)$  of nodes upon an interaction step, we say that  $u$  is the *initiator* of the interaction and  $v$  is the *responder*. The algorithm is described by a state transition function, which is typically given by a collection of local update rules of the form  $A + B \rightarrow C + D$ , where  $A$  and  $B$  are the states of the initiator and the responder at the start of an interaction, and  $C$  and  $D$  are the resulting states after the interaction.

In the case of leader election, nodes have two possible output values to indicate whether they are a *leader* or a *follower*. The goal is to design the local update rules so that the system reaches a *stable configuration* in which (1) exactly one node  $v \in V$  is elected as the leader and all other nodes are followers and (2) the node  $v$  remains as the unique leader no matter what sequence of interactions follows (i.e., all reachable configurations from a stable configuration have the same output). The *time complexity* is measured by *stabilization time*, which is the *total number of interaction steps* needed to reach a stable configuration. The typical aim is to guarantee that stabilization time is small both in expectation and with high probability. Finally, we measure *space complexity* as the maximum number of distinct node states employed by the protocol.

### 1.2 Prior work on leader election in the population model

The foundational work on population protocols [10, 13] already raised the question of how the structure of the interaction graph

influences both the computational power [12] and the complexity [9] of stable computation in the population model. Prior to our work, the complexity of stable leader election on general interaction graphs was an open problem. Instead, most work in this area has focused on a special case of the population model, where the interaction graph is restricted to be a clique [5, 25]. While this special case naturally corresponds to *well-mixed systems*, it is often too simplistic when modelling systems where the interaction patterns among agents are influenced by some underlying spatial structure.

Leader election has been recognized as an especially important problem in this model: for instance, the early work of Angluin, Aspnes and Eisenstat [11] showed that having a leader can be useful in the population model on cliques: semilinear predicates can be stably computed in  $O(n \log^5 n)$  time, and randomized LOGSPACE computation can be performed with small error [11]. The above result has motivated a vast amount of follow-up work on the complexity of leader election on *cliques* [3–5, 15, 16, 24, 25, 27, 30, 31, 40, 42]. By now, the complexity of leader election on the clique is well-understood: there exists a protocol that solves leader election in  $\Theta(n \log n)$  expected steps using  $\Theta(\log \log n)$  states per node [15], which is optimal. To elect a leader in the clique model, all protocols require  $\Omega(n \log n)$  expected steps [40], any  $o(\log \log n)$ -state protocol requires  $n^2/\text{polylog } n$  expected steps [3] and the time complexity bound for constant-state protocols is  $\Theta(n^2)$  expected steps [24].

Somewhat surprisingly, much less is known about the complexity of leader election on general interaction graphs. Angluin, Aspnes, Fischer and Jiang [12] showed that *self-stabilizing* leader election is not generally possible on all connected interaction graphs. At the same time, Beauquier, Blanchard and Burman [14] showed that there exists a constant-state protocol that solves stable leader election as long as self-stabilization is not required. Subsequently, research on leader election in the population model has largely fallen into two categories: (1) work that tries to understand computational complexity and space-time complexity trade-offs of leader election under uniform random pairwise interactions on the clique [4, 15, 16, 24, 27, 30, 31, 40, 42, 43], and (2) work that aims to understand in which interaction graphs and under what model assumptions leader election can be solved in, e.g., a fault-tolerant manner [12, 14, 19, 20, 41, 44–46].

An interesting question left open by this line of work is the *computational complexity* of stable leader election, without the requirement of self-stabilization, on *general interaction graphs* [6]. One reason why this might still be open is that algorithmic [5, 15, 25, 27, 42, 43] and lower bound techniques [3, 24, 40] developed for the clique model do not readily extend to the case of general interaction graphs. More broadly, establishing tight bounds for randomized leader election is known to be challenging even in well-studied synchronous models of distributed computing, such as the LOCAL and CONGEST models [35, 38].

### 1.3 Limits of existing techniques

Existing *upper bound techniques* on the clique naturally rely on the fact that every pair of nodes can potentially interact. Specifically, fast and space-efficient algorithms [15, 27, 43] combine (1) fast information dissemination, typical for the clique, with (2) careful time-keeping across “juntas of nodes” to obtain space-efficient

phase clocks. It is not straightforward to generalize either of these techniques to e.g. poorly-connected graphs, nor is it clear that they would be time-optimal on sparse graphs.

The only existing work to explicitly consider the complexity of non-self-stabilizing leader election on graphs is by Alistarh, Gelashvili and Rybicki [6], whose overall goal was broader, that is to find general ways of porting clique-based algorithms to regular interaction graphs. (Chen and Chen [20] considered complexity of *self-stabilizing* leader election in regular graphs, but this is computationally harder than stable leader election [12].) Specifically for regular graphs, Alistarh et al. [6] gave a leader election protocol that stabilizes in  $1/\phi^2 \cdot n$  polylog  $n$  steps in expectation and with high probability and uses  $1/\phi^2 \cdot \text{polylog } n$  states, where  $\phi$  is the conductance of the interaction graph. While this approach yields to fairly efficient leader election protocols in graphs with high conductance, it performs poorly in low-conductance graphs. For example, on cycles the protocol uses  $n^2$  polylog  $n$  states and requires  $\Omega(n^3)$  steps to stabilize.

Alistarh et al. [6] also showed that the constant-state protocol of Beauquier et al. [14] stabilizes in the order of  $Dmn^2 \log n$  steps in expectation and with high probability on any graph with diameter  $D$  and  $m$  edges. This upper bound can be further refined to  $O(C(G) \cdot n \log n)$ , where  $C(G)$  is the cover time of a classic random walk on the graph  $G$ , by leveraging the recent results of Sudo, Shibata, Nakamura, Kim and Masuzawa [45]. However, beyond the case of cliques, there are no results indicating whether this bound could be improved.

Specifically, existing *lower bound techniques* for population protocols on the clique [3, 24, 40] do not directly generalize to general graphs. In particular, such approaches usually rely on the fact that short executions can lead to “populous” configurations which have large “leader generating” sets of nodes; then, by carefully interleaving interactions between nodes in such sets, short executions can be extended to create new leaders. This suggests that short executions are unlikely to yield stable configurations. However, to create new leaders, existing arguments require the set of nodes to be connected in the underlying graph. This is straightforward on the clique, but non-trivial for general graphs.

The situation seems even more challenging when trying to establish space-time complexity trade-offs, such as showing that constant-state protocols cannot run in sublinear time. In this case, the only known approach is the *surgery technique* [3, 24], which requires keeping track of the distribution of certain states that can be used to generate a leader. On *general graphs*, one would therefore also need to keep track of the *spatial distribution* of states created by the protocol, which appears highly for general protocols and interaction graphs.

To illustrate the difficulty of extending the above techniques to general interaction graphs, a useful exercise is to consider the case of star graphs, as there is a constant-state protocol that elects a leader in a single interaction in any star of size  $n$ . Thus, the lower bound of  $\Omega(n^2)$  expected steps for constant-state protocols or the general lower bound of  $\Omega(n \log n)$  expected steps cannot hold in general, as in some graphs, the graph structure can be used to break symmetry fast.

## 1.4 Our contributions

In this work, we give new upper bounds and lower bounds for stable leader election in the population protocol model on general graphs. For many graph families, we obtain either tight or almost tight bounds; please see Table 1 for a summary of our results. We now overview our main results. The detailed proofs of these results are given in the full version of the paper [7].

*Bounds on information propagation in the population model.* We phrase our upper bounds in terms of *worst-case expected broadcast time*  $\mathbf{B}(G)$  on the graph  $G$ . Informally,  $\mathbf{B}(G)$  denotes the maximum expected time until a broadcast originating from a single node reaches all other nodes in the graph  $G$ . This process is often called “one-way epidemics” in the population protocol literature [11]. In Section 3, we establish the worst-case broadcast time upper bounds of  $O(mD + m \log n)$  and  $O(m/\beta \cdot \log n)$  for any  $m$ -edge graph with diameter  $D$  and edge expansion  $\beta$ . We also provide lower bounds on the time that information propagates to a given distance  $k$ . These bounds are used to bound leader election time for general protocols. While for regular graphs these dynamics correspond to well-studied asynchronous rumour spreading [22, 29], when the graph is not regular, the dynamics in the population model behave differently.

*Fast space-efficient leader in close-to-broadcast time.* We first observe that, if we disregard space complexity, there exists a simple protocol that solves leader election in  $O(\mathbf{B}(G) + n \log n)$  expected steps, on any graph  $G$ : nodes can generate unique identifiers, and then broadcast them to elect a leader. However, generating unique identifiers will require polynomially-many states. Our first contribution is a *space-efficient protocol* that elects a leader in  $O(\mathbf{B}(G) \cdot \log n)$  steps in expectation and with high probability using only  $O(\log n \cdot H(G))$  states, where  $H(G) \in O(\log n)$  is a parameter depending on the broadcast time  $\mathbf{B}(G)$ . Contrasting to the identifier-based approach, this space-efficient protocol achieves exponentially smaller space complexity of  $O(\log^2 n)$ , with a factor  $O(\log n)$  increase in stabilization time.

Our protocol builds on a time-optimal approach on the clique by Sudo, Fukuhito, Izumi, Kakugawa and Masuzawa [43], and significantly improves upon the state-of-the-art on general graphs. Specifically, Alistarh, Gelashvili and Rybicki [6] gave a protocol for leader election on  $\Delta$ -regular graphs that stabilizes in  $O(n/\phi^2 \cdot \log^6 n)$  expected steps and uses  $O(\log^7 n/\phi^2)$  states per node, where  $\phi = \beta/\Delta$  is the conductance of the graph. Our protocol has stabilization time  $O(n/\phi \cdot \log^2 n)$  on regular graphs; this improves the dependency on the conductance  $\phi$  by a linear factor and the polylogarithmic dependence from  $\log^6 n$  to  $\log^2 n$ . In terms of space complexity, we get an *exponential* improvement in conductance, as the parameter  $H$  in the space complexity bound satisfies  $H \in O(\log \log n + \log(1/\phi))$  in regular graphs.

We emphasize that our protocol also works in *non-regular graphs*, and guarantees that the elected leader has degree  $\Theta(\Delta)$  with high probability. Our protocol has high-degree nodes driving a space-efficient and approximate distributed phase clock: nodes with degree  $\Theta(\Delta)$  generate “clock ticks” roughly every  $\mathbf{B}(G)$  steps with high probability. With this in place, we devise a protocol in which high-degree nodes participate in a tournament that lasts for  $O(\log n)$  phases, each of which lasting for  $O(\mathbf{B}(G))$  steps.

*Time lower bounds for general protocols.* On the negative side, we show how to construct families of graphs in which leader election and broadcast have the same asymptotic time complexity. Our approach is based on a probabilistic indistinguishability argument similar in spirit to the lower bound argument of Kutten, Pandurangan, Peleg, Robinson and Trehan [35] for randomized leader election in the synchronous LOCAL and CONGEST models. However, in the population model, communication patterns are asynchronous and stochastic instead of synchronous, so we need a more refined approach to establish the lower bounds.

Roughly speaking, we show that if (a) the nodes of the graph can be divided into constantly many subsets  $V_1, \dots, V_K$  such that the local neighbourhoods of these sets are isomorphic up to some distance  $\ell$  and (b) there are sets whose distance- $\ell$  neighbourhoods are disjoint, then any leader election protocol must propagate information at least up to distance  $\ell$  to reach a stable configuration. If propagation takes at least  $f(n)$  steps with at least a constant probability, then we get a lower bound of order  $f(n)$  for the expected stabilization time. We call such graphs *f-renitent* (see Section 6 for a formal definition).

In general, it is fairly straightforward to construct graphs with diameter  $\Theta(D)$  and  $\Theta(m)$  edges, which are  $\Omega(Dm)$ -renitent for any  $1 \leq D \leq n$  and  $n \leq m \leq n^2$ . Moreover, in these graphs broadcast time is  $\Theta(Dm)$ . Our proof works for a general variant of the population model, in which we do not restrict the state space of the protocol and give each node an *infinite* stream of uniform, fair random bits that assign unique identifiers for each node with probability 1 at the start of the execution. Finally, we also show that in *any* sufficiently dense graph, leader election requires  $\Omega(n \log n)$  expected steps. This part of the argument extends the lower bound argument of Sudo and Masuzawa [40] from cliques to dense graphs.

*Worst-case and average-case complexity of constant-state protocols.* As a baseline result, we show that the constant-state protocol of Beauquier et al. [14] stabilizes in  $O(\mathbf{B}(G) \cdot n \log n)$  steps in expectation and with high probability. It follows from the analyses of Alistarh et al. [6] and Sudo et al. [45] that this stabilizes in  $O(\mathbf{C}(G) \cdot n \log n)$  steps in expectation and with high probability, where  $\mathbf{C}(G)$  is the cover time of a (classic) random walk on  $G$ . Our alternative analysis shows that the worst-case hitting and meeting times of random walks in the population model are bounded by  $O(\mathbf{B}(G) \cdot n)$ .

As our final contribution, we show that, in the class of constant-state protocols, the *average-case complexity* of this protocol on dense random graphs is optimal up to a  $O(\log^2 n)$  factor. More formally, we show that the expected stabilization time of any leader election protocol, that works on all connected graphs, is  $\Omega(n^2)$  on a connected Erdős-Rényi random graph  $G \sim G_{n,p}$  for any constant  $p > 0$ . This is tight up to a polylogarithmic factor, as the broadcast time satisfies  $\mathbf{B}(G) \in O(n \log n)$  with high probability in these graphs. Therefore, the 6-state protocol stabilizes in  $O(n^2 \log^2 n)$  steps with high probability on an average, connected random graph  $G \sim G_{n,p}$ . To achieve this result, we extend the *surgeries technique*, used to prove space-time lower bounds for population protocols so far only in the clique [3, 24], to the case of dense random graphs.

Graph family	Stabilization time	States	Notes
General graphs	$O(\mathbf{B}(G) + n \log n)$	$O(n^4)$	Folklore + Theorem 4.2
	$O(\mathbf{B}(G) \cdot \log n)$	$O(\log^2 n)$	Theorem 5.1
	$O(\mathbf{B}(G) \cdot n \log n)$	$O(1)$	Protocol of [14] + Theorem 4.1
Renitent graphs*	$\Omega(\mathbf{B}(G))$	$\infty$	Theorem 6.5
Regular graphs	$O(n\phi^{-2} \log^6 n)$	$O(\phi^{-2} \log^7 n)$	[6]
	$O(n\phi^{-1} \log n)$	$O(n^3)$	Theorem 4.2
	$O(n\phi^{-1} \log^2 n)$	$O(\log n \cdot (\log \log n - \log \phi))$	Corollary 5.2
	$O(n^2\phi^{-1} \log^2 n)$	$O(1)$	Theorem 4.1
Cliques	$\Theta(n \log n)$	$\Theta(\log \log n)$	[3, 15, 40]
	$\Theta(n^2)$	$O(1)$	[24]
Dense random graphs**	$\Theta(n \log n)$	$O(n^4)$	Theorem 6.6; Theorem 4.2
	$O(n \log^2 n)$	$O(\log^2 n)$	Theorem 5.1
	$\Omega(n^2)$ and $O(n^2 \log^2 n)$	$O(1)$	Theorem 7.1; Theorem 4.1
Star graphs	$O(1)$	$O(1)$	Trivial

**Table 1: Complexity bounds for stable leader election. Stabilization time refers to the expected number of steps required to reach a stable configuration. Here  $\mathbf{B}(G)$  is a characteristic of information dissemination dynamics in the population model defined in Section 3; it can range from  $\Theta(n \log n)$  to  $\Theta(n^3)$ . The  $O(\log^2 n)$  and  $O(1)$ -state protocols also stabilize in the reported time w.h.p. For regular graphs,  $\phi$  denotes the conductance of the graph. (\*) We define the class of renitent graphs in Section 6. (\*\*) For dense random graphs, the bounds are for *average-case complexity*, when the input graph is a connected Erdős-Rényi random graph  $G \sim G_{n,p}$  for any constant  $p > 0$ . In these graphs, broadcast time is  $O(n \log n)$  w.h.p., which implies the given bounds.**

## 2 PRELIMINARIES

*Graphs.* Let  $G = (V, E)$  be a undirected graph, where  $V = V(G)$  is the set of nodes and  $E(G) = E$  is the set of edges of the graph. We use  $n = |V(G)|$  to denote the number of nodes and  $m = |E(G)|$  the number of edges. The degree  $\deg(v)$  of a node  $v$  is the number of edges incident to it. We use  $\Delta = \max\{\deg(v) : v \in V\}$  to denote the maximum degree and  $\delta = \min\{\deg(v) : v \in V\}$  the minimum degree of the graph. We assume all interaction graphs are connected unless otherwise specified.

Given a nonempty set  $S \subseteq V(G)$ , the edge boundary  $\partial S$  of  $S$  is the set  $\partial S = \{\{u, v\} \in E(G) : u \in S, v \in V(G) \setminus S\}$ . The *edge expansion* of  $G$  is given by  $\beta(G) = \min\{|\partial S|/|S| : \emptyset \neq S \subseteq V(G), |S| \leq n/2\}$ . We also define  $G[S]$  to be a subgraph of  $G$  induced on vertices of  $S$ . The distance between two vertices  $u$  and  $v$  is denoted by  $\text{dist}(u, v)$ . The radius- $r$  neighbourhood of  $u$  is  $B_r(u) = \{v \in V : \text{dist}(u, v) \leq r\}$  and  $B_r(U) = \bigcup_{u \in U} B_r(u)$ . For  $r = 1$  we use  $B(u) = B_1(u)$ . The diameter  $D(G)$  is given by  $D(G) = \max\{\text{dist}(u, v) : u, v \in V(G)\}$ . For any two graphs  $G$  and  $H$ , we write  $G \simeq H$  if they are isomorphic. For random graphs, we use the Erdős-Rényi random graph model  $G_{n,p}$ . In this model, a random graph  $G \sim G_{n,p}$  is sampled as follows. We start with  $n$  nodes and for each  $u, v \in V$ , we add the edge  $\{u, v\}$  with probability  $p$  independently of all other edges.

*Population protocols on graphs.* A (stochastic) *schedule* on a graph  $G$  is an infinite sequence  $(e_t)_{t \geq 1}$  of ordered pairs of nodes  $(v, u)$ , where each  $e_t$  is sampled independently and uniformly at random among all pairs of nodes connected by an edge in  $G$  (there are  $2m$  such pairs). The order of nodes in the pair is used to distinguish between initiator and a responder. A *protocol* is a tuple  $\mathcal{A} = (\Lambda, \xi, \Sigma_{\text{in}}, \Sigma_{\text{out}}, \text{init}, \text{out})$ , where  $\Lambda$  is the set of states,

$\xi: \Lambda \times \Lambda \rightarrow \Lambda \times \Lambda$  is the state transition function,  $\Sigma_{\text{in}}$  and  $\Sigma_{\text{out}}$  are the sets of input and output labels, respectively,  $\text{init}: \Sigma_{\text{in}} \rightarrow \Lambda$  is the initialization function, and  $\text{out}: \Lambda \rightarrow \Sigma_{\text{out}}$  is the output function.

A configuration is a map  $x: V \rightarrow \Lambda$ , where  $x(v)$  is the state of the node  $v$  in configuration  $x$ . For any  $e = (u, v)$  and configurations  $x$  and  $x'$ , we write  $x \Rightarrow_e x'$  if  $x'(u), x'(v) = \xi(x(u), x(v))$  and  $x'(w) = x(w)$  for all  $w \in V \setminus \{u, v\}$ . For any sequence  $\sigma = (e_1, \dots, e_t)$  we write  $x_0 \Rightarrow_{\sigma} x_t$  if  $x_i \Rightarrow_{e_{i+1}} x_{i+1}$  for each  $i \geq 0$ . We say that  $x'$  is reachable from  $x$  on  $G$  if there exists some  $k \geq 1$  and  $\sigma = (e_1, \dots, e_k)$  such that  $x \Rightarrow_{\sigma} x'$ . Given input  $f: V(G) \rightarrow \Sigma_{\text{in}}$ , a protocol and a schedule  $(e_t)_{t \geq 1}$ , an *execution* is the infinite sequence  $(x_t)_{t \geq 0}$  of configurations, where  $x_0 = \text{init} \circ f$  is the initial configuration and  $x_t \Rightarrow_{e_{t+1}} x_{t+1}$  for  $t \geq 0$ . Note that throughout, the time step  $t$  denotes the total number of pairwise interactions that have occurred so far.

In the case of leader election, we assume that the input is a constant function, unless otherwise specified. That is, all nodes start in the same state. We say that a configuration  $x$  is *correct* if  $\text{out}(x(v)) = \text{leader}$  for exactly one node  $v \in V$  and for all  $u \in V \setminus \{v\}$  we have  $\text{out}(x(u)) = \text{follower}$ . A configuration  $x$  is *stable* if for every configuration  $x'$  reachable from  $x$  we have  $\text{out}(x(v)) = \text{out}(x'(v))$  for every node  $v \in V$ . The *stabilization time* of a leader election protocol  $\mathcal{A}$  is the minimum  $t$  such that  $x_t$  is stable and correct. The *state complexity* of a protocol is  $|\Lambda|$ , the number of distinct states.

Some of the protocols we consider are non-uniform in the following sense: the state space and transition function of the protocol can depend on parameters that capture high-level structural information about the population and the interaction graph (e.g.,

number of nodes and edges, broadcast time or the maximum degree). However, upon initialization, all nodes receive exactly the same information. For example, nodes do not initially know their own degree or identity in the interaction graph.

*Probability-theoretic tools.* Let  $X$  and  $Y$  be real-valued random variables defined on the same probability space. We say that  $X$  *stochastically dominates*  $Y$ , written as  $Y \leq X$ , if  $\Pr[X \geq x] \geq \Pr[Y \geq x]$  for all  $x \in \mathbb{R}$ . We start with three concentration bounds. The first is a folklore result; see e.g. [18] for a proof. The second is also standard Chernoff bounds for sums of Bernoulli random variables. The third result gives tail bounds on the sums of geometric random variables, via Janson [32, Theorems 2.1 and 3.1].

LEMMA 2.1. *Let  $X \sim \text{Poisson}(\lambda)$  be a Poisson random variable with mean  $\lambda$ . Then*

- (a)  $\Pr[X \geq c\lambda] \leq \exp(-\lambda \cdot (c-1)^2/c)$  for  $c \geq 1$ ,
- (b)  $\Pr[X \leq c\lambda] \leq \exp(-\lambda \cdot (1-c)^2/(2-c))$  for  $c \leq 1$ .

LEMMA 2.2. *Let  $X = X = Y_1 + \dots + Y_k$  be a sum of independent Bernoulli random variables with  $\Pr[Y_i = 1] = p_i$ . Then*

- (a)  $\Pr[X \geq (1+\lambda) \cdot \mathbb{E}[X]] \leq \exp(-\mathbb{E}[X] \cdot \lambda^2/3)$  for any  $\lambda \geq 1$ ,
- (b)  $\Pr[X \leq (1-\lambda) \cdot \mathbb{E}[X]] \leq \exp(-\mathbb{E}[X] \cdot \lambda^2/2)$  for any  $\lambda \leq 1$ .

*Note that in the special case when  $p_i = p$  for all  $1 \leq i \leq k$ , the sum  $X \sim \text{Bin}(k, p)$  is a Binomial random variable.*

LEMMA 2.3. *Let  $p_1, \dots, p_k \in (0, 1]$  and  $X = Y_1 + \dots + Y_k$  be a sum of independent geometric random variables with  $Y_i \sim \text{Geom}(p_i)$ . Define  $p = \min\{p_i : 1 \leq i \leq k\}$  and  $c(\lambda) = \lambda - 1 - \ln \lambda$ . Then*

- (a)  $\Pr[X \geq \lambda \cdot \mathbb{E}[X]] \leq e^{-p \cdot \mathbb{E}[X] \cdot c(\lambda)}$  for any  $\lambda \geq 1$ ,
- (b)  $\Pr[X \leq \lambda \cdot \mathbb{E}[X]] \leq e^{-p \cdot \mathbb{E}[X] \cdot c(\lambda)}$  for any  $0 < \lambda \leq 1$ .

LEMMA 2.4 (WALD'S IDENTITY). *Let  $(X_i)_{i \geq 1}$  be a sequence of real-valued i.i.d. random variables and  $N$  a non-negative integer-valued random variable independent of  $(X_i)_{i \geq 1}$ . If  $N$  and all  $X_i$  have finite expectation, then  $\mathbb{E}[X_1 + \dots + X_N] = \mathbb{E}[N] \cdot \mathbb{E}[X_1]$ .*

### 3 BOUNDS ON INFORMATION PROPAGATION

Our results will rely on notions of *broadcast time* and *propagation time* in the population model. For this, we define the following infection process on a graph  $G$ : initially, each node  $v \in V$  holds a unique message. In every step, when nodes  $u$  and  $v$  randomly interact they inform each other about *all messages* they have so far received. The distance- $k$  propagation time is the minimal time until some message has reached a node at distance  $k$  from its source. The broadcast time is the expected time until all nodes in the network are aware of all messages. Propagation time is used in our lower bounds, whereas broadcast time appears in our upper bounds. Before we formalize these notions below, we briefly discuss some work on related, but different stochastic information propagation dynamics.

#### 3.1 Information propagation in related models

Many variants of the above broadcasting process have been studied in settings ranging from information dissemination [1, 21–23, 29, 34, 39] to models of epidemics [26, 36, 37]. For example, in the synchronous push-pull model [23, 34], Chierichetti, Lattanzi and Panconesi [22] first showed that broadcast succeeds

with high probability in  $O(\log^4 n / \phi^6)$  rounds on graphs of conductance  $\phi$ . Subsequently, they improved the running time bound to  $O(\log n / \phi \cdot \log^2(1/\phi))$  rounds [21]. Finally, Giakkoupis [28] showed that the push-pull algorithm succeeds in  $O(\log n / \phi)$  rounds with high probability, and showed that for all  $\phi \in \Omega(1/n)$  there is a family of graphs in which this bound is tight.

In the asynchronous setting, Acan, Collevocchio, Mehrabian and Wormald [1] and Giakkoupis, Nazari and Woelfel [29] studied broadcasting in the continuous-time push-pull model, where each *node* has a (probabilistic) Poisson clock that rings at unit rate. They showed that on graphs in which the protocol runs in  $T$  rounds, the asynchronous protocol runs in  $O(T + \log n)$  continuous time. Ottino-Löffler, Scott and Strogatz [37] studied an infection model that is similar to this asynchronous setting, and characterized broadcast time in cliques, stars, lattices and Erdős–Rényi random graphs.

Although the interaction patterns in the stochastic population model and the above asynchronous models are the same for *regular graphs*, they are different in general graphs. In the population model, instead of sampling a node and then one of its neighbours in each step, our scheduler samples an *edge*. In the continuous-time setting, this corresponds to having an independent Poisson clock at each *edge* rather than each *node* in the network. Thus, high-degree nodes interact more often than low-degree nodes in the population model.

#### 3.2 Information propagation in the population model

We now define information propagation dynamics in our setting. Let  $(e_t)_{t \geq 1}$  be a stochastic schedule on a graph  $G = (V, E)$ . For each node  $v \in V$ , let  $I_0(v) = \{v\}$ . For  $t \geq 0$ , define

$$I_{t+1}(v) = \begin{cases} I_t(v) \cup I_t(u) & \text{if } e_{t+1} = (u, v) \text{ or } e_{t+1} = (v, u); \\ I_t(v) & \text{otherwise.} \end{cases}$$

Following Sudo and Masuzawa [40], we say that  $I_t(v)$  is the *set of influencers* of node  $v$  at the end of step  $t$ . Nodes in  $I_t(v)$  are nodes who can (in principle) influence what is the state of node  $v$  at step  $t$ . The above dynamics can be seen as a rumour spreading process, where each node starts with a unique message, and whenever two nodes interact, they inform each other about all messages they possess.

*Broadcast and propagation time.* Let  $T(v, u) = \min\{t : v \in I_t(u)\}$  be the minimum time until node  $u$  is influenced by node  $v$ . The *broadcast time from source  $v$*  is  $T(v) = \max\{T(v, u) : u \in V(G)\}$ . We define the *worst-case expected broadcast time* on  $G$  to be

$$\mathbf{B}(G) = \max\{\mathbb{E}[T(v)] : v \in V\}.$$

For each  $k \geq 0$ , let  $T_k(u) = \min\{T(u, v) : v \in V, \text{dist}(u, v) = k\}$ . The *distance- $k$  propagation time* in  $G$  is  $T_k(G) = \min\{T_k(u) : u \in V\}$ . If there are no nodes at distance  $k$  from node  $u$ , then  $T_k(u) = \infty$ . Moreover,  $T_k(G) = \infty$  for all  $k > D(G)$ . Note that the distance- $k$  propagation time gives lower bound for the expected broadcast time as  $\mathbb{E}[T_k(G)] \leq \mathbb{E}[T_D(G)] \leq \mathbf{B}(G)$  for each  $1 \leq k \leq D(G)$ .

*Sampling edge sequences.* For a finite sequence  $\rho \in E^k$  of  $k$  edges, let  $X(\rho)$  be the number of steps until the scheduler has sampled each edge from  $\rho$  in order. Note that  $X(\rho) = Y_1 + \dots + Y_k$  is a sum of i.i.d. geometric random variables, where  $Y_i \sim \text{Geom}(1/m)$  is the number

of steps until the  $i$ th edge of  $\rho$  is sampled after sampling the  $(i-1)$ th edge in the sequence  $\rho$ . The next lemma follows from Lemma 2.3.

LEMMA 3.1. *Let  $c(\lambda) = \lambda - 1 - \ln \lambda$ . For any fixed sequence  $\rho \in E^k$  we have  $\mathbf{E}[X(\rho)] = km$  and*

- (a)  $\Pr[X(\rho) > \lambda km] \leq e^{-kc(\lambda)}$  for  $\lambda \geq 1$ , and
- (b)  $\Pr[X(\rho) < \lambda km] \leq e^{-kc(\lambda)}$  for  $0 < \lambda \leq 1$ .

With the above lemma, it is fairly straightforward to establish the following upper bound on the worst-case expected broadcast time  $\mathbf{B}(G)$ . We give the details in the full version of the paper [7].

THEOREM 3.2. *Let  $G$  be a graph with  $n$  nodes,  $m$  edges, edge expansion  $\beta$  and diameter  $D$ . Then*

$$\mathbf{B}(G) \in O\left(m \cdot \min\left\{\frac{\log n}{\beta}, \log n + D\right\}\right).$$

Note that there are graphs in which  $\ln n/\beta > D$ , e.g. cycles, and  $\ln n/\beta < D$ , e.g., cliques. We will later give leader election protocols whose stabilization time is bounded as a function of  $\mathbf{B}(G)$  on any graph  $G$ . In general, for any increasing function  $T$  between  $\Omega(n \log n)$  and  $O(n^3)$ , we can find families of graphs in which both the expected broadcast time and leader election time are  $\Theta(T)$ . We give the construction in Section 6.

## 4 TWO BASELINES FOR STABLE LEADER ELECTION ON GRAPHS

In this section, we discuss two protocols, which act as our baselines for time complexity and space complexity. First, we observe that constant-state protocol given by Beauquier et al. [14] stabilizes in  $O(\mathbf{B}(G) \cdot n \log n)$  steps in expectation and with high probability. Second, we note that if we allow polynomially-many states, then there is a simple protocol that elects a leader in  $O(\mathbf{B}(G) + n \log n)$  expected steps. This protocol is time-optimal for a large class of graphs.

*First baseline: A space-efficient protocol.* Our baseline for *space-efficient protocols* is the constant-state leader election protocol given by Beauquier et al. [14]. This protocol stabilizes in any connected graph in finite expected time. The idea of the protocol is simple: as input, we are given a nonempty set of leader candidates. At the start of the execution, each leader candidate creates a “black token”. In each interaction, the selected nodes swap their tokens with their interaction partners. When ever two black tokens meet, one of them is colored white and the other token is left black. Whenever a leader candidate receives a white token, the candidate drops out of the race by becoming a follower and removes the token from the system. Eventually exactly one black token and leader remain. That is, the protocol is always correct. Similarly to Alistarh et al. [6], we exploit this property by using this constant-state protocol as a *backup protocol* for faster protocols that may fail to elect a unique leader with a polynomially small probability.

Recently, Alistarh et al. [6] and Sudo et al. [45] bounded the meeting and hitting times of randomly walking tokens in the population model. Using these analyses, it is possible to show that the protocol stabilizes in  $O(\mathbf{C}(G) \cdot n \log n)$  steps, where  $\mathbf{C}(G)$  is the cover time of the classic random walk [2, 17, 33]. In the full version of the paper [7], we give an alternative analysis of the protocol, which connects hitting and meeting times of the randomly walking

tokens to the broadcast time  $\mathbf{B}(G)$ . With this, we can bound the stabilization time of the 6-state protocol using the broadcast time  $\mathbf{B}(G)$ ; this allows a clean comparison of performance of the three different protocols we consider in this work. In Section 7, we show that the *average-case time complexity* of this protocol on dense random graphs is almost-optimal among constant-state protocols.

THEOREM 4.1. *Given a nonempty set of leader candidates as input, there is a 6-state protocol that elect exactly one candidate as a leader in  $O(\mathbf{B}(G) \cdot n \log n)$  steps in expectation and with high probability.*

*Second baseline: A time-efficient protocol.* We next discuss our baseline for *fast protocols*. For this, we use a simple protocol that stabilizes in  $O(\mathbf{B}(G) + n \log n)$  expected steps using polynomially many states. The results in Section 6 show that this protocol is time-optimal for a large class of graphs, as there are graphs where leader election requires  $\Theta(\mathbf{B}(G))$  steps. In this protocol, we first generate unique identifiers with high probability by using the stochasticity of the scheduler and a large state space. Once we have unique identifiers, we can elect the node with the largest identifier as the leader by a broadcast process.

The only non-trivial part is to get finite expected stabilization time. This is achieved by interleaving the always-correct constant-state protocol with the broadcasting process. Once a node has generated its identifier, the node starts an instance of the constant-state protocol labelled with its own identifier and designating itself as a leader. If a node encounters an instance of the constant-state protocol labeled with an identifier higher than the identifier of its current instance (or its own identifier), the node joins as a follower to the instance with the higher identifier. In the case that two or more nodes generated the same (highest) identifier, the constant-state protocol ensures that eventually only one leader candidate remains. The analysis of this protocol, given in the full version of the paper [7], yields the next result.

THEOREM 4.2. *There is a protocol that uses  $O(n^4)$  states on general graphs and  $O(n^3)$  states on regular graphs that elects a leader in  $O(\mathbf{B}(G) + n \log n)$  steps in expectation.*

## 5 SPACE-EFFICIENT LEADER ELECTION IN CLOSE-TO-BROADCAST TIME

We now give a leader election protocol whose stabilization time is parameterized by the worst-case expected broadcast time  $\mathbf{B}(G)$  and whose state complexity depends on the expansion properties of the graph. The approach is inspired by a time-optimal algorithm on the clique due to Sudo et al. [43], but with significant differences: for instance, our algorithm works on any connected graph, and guarantees that a high-degree node is elected as a leader.

THEOREM 5.1. *For any graph  $G$  with maximum degree  $\Delta$ , there is a leader election protocol that uses  $O(\log n \cdot H(G))$  states and stabilizes in  $O(\mathbf{B}(G) \cdot \log n)$  steps in expectation and with high probability, where  $H(G) = \log(\Delta/\beta \cdot \log n)$ .*

Observe that  $H(G) \in O(\log n)$ , so the protocol uses  $O(\log^2 n)$  states. Moreover, for graphs where the ratio  $\Delta/\beta$  is small, we can obtain  $o(\log n^2)$  space complexity. For example, in regular graphs we get the following bounds.

**COROLLARY 5.2.** *In any regular graph with conductance  $\phi = \beta/\Delta$ , there is a leader election protocol that stabilizes in  $O(1/\phi \cdot n \log^2 n)$  steps in expectation and with high probability using  $O(\log n \cdot (\log \log n - \log \phi))$  states.*

The algorithm consists of three parts. First, we describe a space-efficient way for nodes to approximately count the number of *local* interactions. The second part is a two-phase protocol that first removes low-degree nodes from the set of leader candidates and then reduces the number of high-degree leader candidates to one with high probability. Finally, to guarantee finite expected stabilization time, we use the constant-state token-based leader election protocol given in Theorem 4.1 as a backup protocol to handle the unlikely cases where the fast part fails.

## 5.1 Local approximate clocks on graphs: Triggering events at a controlled frequency

We first describe a subroutine that is used to trigger events every  $\Theta(2^H)$  expected interactions using exactly  $H + 1$  local states, where  $H \geq 1$  is a given parameter controlling the frequency of the triggered events. Each node  $v$  maintains a variable  $\text{streak}(v) \in \{0, \dots, H\}$ , which is initialized to 0. When  $v$  interacts, it updates its streak counter as follows:

- If  $v$  is the initiator, then set  $\text{streak}(v) \leftarrow \text{streak}(v) + 1$ . Otherwise, set  $\text{streak}(v) \leftarrow 0$ .
- If  $\text{streak}(v) = H$ , then node  $v$  is said to *complete a streak*. Set  $\text{streak}(v) \leftarrow 0$ .

Let  $K$  denote the number of times a fixed node needs to interact until it completes a streak. Here  $K$  is the number of fair coin flips needed to observe  $H$  consecutive heads, as the scheduler picks the role of initiator and responder uniformly at random and independently from previous interactions. We start with a technical result that approximates the distribution of  $K$  using geometric random variables. Recall that  $X \leq Y$  denotes that  $Y$  stochastically dominates  $X$ .

**LEMMA 5.3.** *The random variable  $K$  satisfies  $Z_0 \leq K \leq Z_1 + H$ , where  $Z_0 \sim \text{Geom}(2^{-H})$  and  $Z_1 \sim \text{Geom}(2^{-H-1})$ .*

We use  $X(d)$  to denote the number of *steps* (i.e., the number of node pairs sampled by the scheduler) until a fixed node of degree  $d$  completes a streak. Note that high degree nodes have a higher probability to complete their streaks, as they interact more often. The next lemma summarizes some useful properties of  $K$  and  $X(d)$ . The proof follows by application of concentration bounds on geometric random variables (Lemma 2.3) and Wald's identity (Lemma 2.4).

**LEMMA 5.4.** *Let  $1 \leq d \leq n$ . The random variables  $K$  and  $X(d)$  satisfy the following:*

- (a) *The expected value of  $K$  is  $\mathbb{E}[K] = 2^{H+1} - 2$ .*
- (b) *The expected value of  $X(d)$  is  $\mathbb{E}[X(d)] = \mathbb{E}[K] \cdot m/d$ .*
- (c) *For any  $0 \leq \lambda \leq 1$ ,  $\Pr[X(d) \leq \lambda \mathbb{E}[X(d)]] \leq 4\lambda + 2^{1-\lambda H}$ .*

Together with the above results, we can show that the number  $R$  of interactions to complete  $\ell \geq \ln n$  streaks is strongly concentrated around the interval  $[\mathbb{E}[R]/2, 4\mathbb{E}[R]]$ . Thus, the above process can be essentially used as space-efficient local clock which ticks at a desired (approximate) frequency. In the following, we write  $c(\lambda) = \lambda - 1 - \ln \lambda$  for any  $\lambda > 0$ .

**LEMMA 5.5.** *Let  $\ell \geq \ln n$  and  $R$  be the number of interactions a node needs to complete  $\ell$  streaks. Then*

- (a) *The expected value of  $R$  is  $\mathbb{E}[R] = (2^{H+1} - 2)\ell$ .*
- (b)  *$\Pr[R \leq \lambda \cdot \mathbb{E}[R]/2] \leq 1/n^{c(\lambda)}$  for all  $0 < \lambda \leq 1$*
- (c)  *$\Pr[R \geq \lambda \cdot 4\mathbb{E}[R]] \leq 1/n^{c(\lambda)}$  for all  $\lambda \geq 1$*

Finally, we examine the concentration of the number of steps until a node completes a certain number of streaks.

**LEMMA 5.6.** *Suppose  $H \in \omega(1)$ . Let  $\ell \geq \ln n$  and  $S = S(d, \ell)$  be the number of steps until a fixed node of degree  $d$  completes  $\ell$  streaks. Then for all sufficiently large  $n$ , we have*

- (a)  *$\mathbb{E}[S] = \mathbb{E}[K] \cdot \ell m/d = (2^{H+1} - 2) \cdot \ell m/d$ ,*
- (b)  *$\Pr[S \leq \lambda^2 \cdot \mathbb{E}[S]/4] \leq 2/n^{c(\lambda)}$  for any  $\lambda \leq 1$ , and*
- (c)  *$\Pr[S \geq \lambda^2 \cdot 8\mathbb{E}[S]] \leq 2/n^{c(\lambda)}$  for any  $\lambda \geq 1$ .*

## 5.2 The fast leader election protocol

With the time-keeping mechanism in place, we now describe and analyse the leader election protocol that reduces the number of leader candidates to *one*, with high probability, in  $O(\mathbf{B}(G) \cdot \log n)$  steps. Let  $\tau \geq 1$  be an arbitrary fixed constant that controls the probability that the protocol fails (increasing  $\tau$  decreases the probability of failure). Fix the parameters

$$H = 8 + \lceil \log(\mathbf{B}(G) \cdot \Delta/m) \rceil \quad \text{and} \quad L = \lceil 2\tau \log n \rceil,$$

where  $\Delta$  denotes the maximum degree of the graph  $G$ . Note that with this choice of parameters  $X(d) \in O(\mathbf{B}(G))$  for any  $d \leq \Delta$  and also  $X(d) \in \Theta(\mathbf{B}(G))$  for  $d \in \Theta(\Delta)$ . We also note that  $H \in \Omega(\log \log n)$ , as in any graph  $\mathbf{B}(G) \geq m \ln(n-1)/\Delta$ .

*The protocol.* As a subroutine, each node runs the streak counter protocol with  $H$  fixed as above. Every node  $v$  also maintains two state variables  $\text{status}(v) \in \{\text{leader}, \text{follower}\}$  and a local counter  $\text{level}(v) \in \{0, \dots, \alpha(\tau) \cdot L\}$ , where  $\alpha(\tau) > 1$  is a constant we fix later in the analysis. Each node  $v$  initializes the variables to  $\text{status}(v) \leftarrow \text{leader}$  and  $\text{level}(v) \leftarrow 0$ . When  $v$  interacts with  $u$ , node  $v$  updates its state using the following rules applied in sequence:

- (1) If  $v$  completes a streak and  $\text{status}(v) = \text{leader}$ , then set  $\text{level}(v) \leftarrow \min\{\text{level}(v) + 1, \alpha(\tau) \cdot L\}$ .
- (2) If  $\text{level}(v) < \text{level}(u)$  and  $\text{level}(u) \geq L$ , then set  $\text{status}(v) \leftarrow \text{follower}$ .
- (3) If  $\max\{\text{level}(u), \text{level}(v)\} \geq L$ , then set  $\text{level}(v) = \max\{\text{level}(u), \text{level}(v)\}$ .

*The analysis.* We now analyse the protocol. We say that a node is at level  $\ell$  at time step  $t$  if its level variable is  $\ell$  at time step  $t$ . A node is in the *elimination phase* if its at least at level  $L$ . Otherwise, it is in the *waiting phase*. When a node  $v$  in the waiting phase interacts with a node in the elimination phase, then  $v$  moves to the elimination phase (as a follower).

Note that a node can only remain a leader and increase its level if it completes a streak. When a node increases its level without completing a streak, it must become a follower by Rules (2) and (3). Moreover, by Rule (2), in every time step one of the nodes in the graph with the highest level must be a leader. Thus, the protocol guarantees that there is always at least one leader in every step. Finally, as  $\mathbf{B}(G)$  is the worst-case expected broadcast time, Rule (3)

implies that if some node is at level  $\ell \geq L$  at step given step, then within  $\mathbf{B}(G)$  expected steps all nodes are at level at least  $\ell$ .

The first step in the analysis considers fixed pairs of nodes, characterizing the period of time after which at least one node from a given pair drops out of contention.

**LEMMA 5.7.** *Let  $u$  and  $v$  be nodes with degree at least  $d$ . If  $u$  and  $v$  have level at least  $L$  and less than  $\alpha(\tau) \cdot L$  at step  $t$ , then at least one of them is a follower at time step  $t + 16 \cdot (\mathbf{E}[X(d)] + \mathbf{B}(G))$  with probability at least  $1/8$ .*

The second technical lemma leverages this to show a higher concentration result for eliminating all-but-one candidate from contention, assuming all nodes are in the elimination phase.

**LEMMA 5.8.** *Suppose all leader candidates have degree at least  $d$  and all nodes are in the elimination phase. Let  $t(d) = 16(\tau + 2) \cdot \log_{8/7} n \cdot (\mathbf{E}[X(d)] + \mathbf{B}(G))$ . If no node reaches level  $\alpha(\tau)L$  by time  $t + t(d)$ , then exactly one leader candidate remains at time step  $t + t(d)$  with probability at least  $1 - O(n^{-\tau})$ .*

Next, we provide an upper bound on the time when, with high probability, all nodes are in the elimination phase, and all nodes of small degree have been eliminated.

**LEMMA 5.9.** *There exist constants  $\lambda \geq 1$  and  $\gamma \geq 1$  such that at time  $\lambda L \cdot \mathbf{B}(G)$  the following holds with probability  $1 - O(n^{-\tau})$ :*

- (1) all nodes are in the elimination phase, and
- (2) all nodes of degree at most  $\Delta/\gamma$  are followers.

Finally, we put everything together to obtain a w.h.p. bound on the time by which there is a single candidate left.

**LEMMA 5.10.** *There exist constants  $\alpha(\tau)$  and  $C = C(\tau)$  such that there is exactly one leader candidate at time step  $C \cdot \mathbf{B}(G) \cdot \log n$  with probability at least  $1 - O(n^{-\tau})$ .*

Finally, to guarantee finite expected stabilization time, the protocol includes a backup phase following the same approach as in [6]. The first node to reach level  $\alpha(\tau)L$  must be a leader candidate. When a node  $v$  reaches level  $\alpha(\tau)L$ , it switches to executing the constant-state token-based leader election protocol. When this happens, node initializes the constant-state protocol with the input  $\text{status}(v) \in \{\text{leader}, \text{follower}\}$  and starts running the protocol while simultaneously continues broadcasting its  $\text{level}(v)$  value using Rule (3). Within  $\mathbf{B}(G)$  expected steps, all nodes are running the constant-state protocol. This protocol guarantees that eventually only one leader remains after polynomially many expected steps.

**THEOREM 5.1.** *For any graph  $G$  with maximum degree  $\Delta$ , there is a leader election protocol that uses  $O(\log n \cdot H(G))$  states and stabilizes in  $O(\mathbf{B}(G) \cdot \log n)$  steps in expectation and with high probability, where  $H(G) = \log(\Delta/\beta \cdot \log n)$ .*

**PROOF.** The protocol uses  $O(HL)$  states. By Theorem 3.2,  $\mathbf{B}(G) \leq Cm/\beta \log n$  for some constant  $C$ . Thus,

$$\begin{aligned} H &= 8 + \left\lceil \log \left( \frac{\mathbf{B}(G) \cdot \Delta}{m} \right) \right\rceil \\ &\leq 8 + \left\lceil \log \left( \frac{Cm \log n \cdot \Delta}{m\beta} \right) \right\rceil \\ &\in O \left( \log \left( \frac{\log n \cdot \Delta}{\beta} \right) \right). \end{aligned}$$

Clearly,  $L \in \Theta(\log n)$ , so the claim on the state complexity follows. By Lemma 5.10, the fast protocol stabilizes in  $O(\mathbf{B}(G) \cdot \log n)$  steps with probability at least  $1 - O(n^{-\tau})$ . By Theorem 4.1, the constant-state backup protocol stabilizes in  $O(n^4 \log n)$  expected steps, as  $\mathbf{B}(G)$  is  $O(n^3)$  for any connected graph by Theorem 3.2. With probability at most  $O(n^{-\tau})$ , at least two leader candidates enter the backup phase. Choose  $\tau \geq 4$  and let  $T(\tau)$  be stabilization time of the protocol. Then  $T(\tau) \in O(\mathbf{B}(G) \cdot \log n + n^{-\tau} \cdot n^4 \log n)$ , which is  $O(\mathbf{B}(G) \cdot \log n)$ .  $\square$

## 6 TIME LOWER BOUNDS FOR GENERAL PROTOCOLS

In this section, we establish time lower bounds for leader election for general protocols with unbounded state space. First, we give a fairly general technique for constructing graphs, where leader election has desired time complexity. This technique can be also applied to specific graph families to characterize the complexity of leader election in these families. We also give a result that shows that in any sufficiently dense graph leader election requires  $\Omega(n \log n)$  steps.

### 6.1 The lower bound for renitent graphs

We first introduce the notion of *isolating covers*. The idea is that we can cover the vertices of the graph with at most  $K$  subsets of the same size, each of which has isomorphic neighbourhood up to some distance  $\ell \geq 0$ , and that there are at least two such sets that are sufficiently far apart. Let  $G = (V, E)$  be a graph and  $C = \{V_0, \dots, V_{K-1}\} \subseteq 2^V$  be a collection of subsets of  $V$ . We say that  $C$  is a  $(K, \ell)$ -cover of the graph  $G$  if

- (1) for each  $0 \leq i < j < K$  there exists an isomorphism  $\phi$  between  $G[B_\ell(V_i)]$  and  $G[B_\ell(V_j)]$  such that  $\phi(V_i) = V_j$ ,
- (2) there exists some  $V_i$  and  $V_j$  such that  $B_\ell(V_i) \cap B_\ell(V_j) = \emptyset$ ,
- (3)  $V_0 \cup \dots \cup V_{K-1} = V(G)$ .

That is, (1) the local neighbourhoods are isomorphic up to distance  $\ell$  and this isomorphism maps vertices of  $V_i$  to  $V_j$ , (2) there are two sets whose vertices are all far apart, and (3) the union of the sets covers the entire graph.

Define  $Y(C) = \min\{t : I_t(V_i) \setminus B_\ell(V_i) \neq \emptyset \text{ for some } V_i\}$  to be the *isolation time* of the cover  $C$ . This is the minimum time until some node in  $V_i$  is influenced by some node at distance greater than  $\ell$  from all nodes of  $V_i$ . We say that  $C$  is *t-isolating* if  $\Pr[Y(C) \geq t] \geq 1/2$ . This property states that it is unlikely that during the first  $t$  steps, nodes in the set  $V_i$  can be influenced by nodes that are far away from nodes in  $V_i$ . Note that if the distance- $\ell$  propagation time on  $G$  satisfies  $\Pr[T_\ell(G) < t] \leq 1/2$ , then any  $(K, \ell)$ -cover of  $G$  is *t-isolating*. Thus, we may bound the minimum propagation times to show that a cover is isolating.

Let  $\mathcal{G}$  be an infinite family of graphs and  $f: \mathbb{N} \rightarrow \mathbb{N}$  be an increasing function. We say that graphs in  $\mathcal{G}$  are *f-renitent* if there exists a constant  $K \geq 2$  and function  $\ell: \mathbb{N} \rightarrow \mathbb{N}$  and such that every  $n$ -node graph  $G \in \mathcal{G}$  has an  $f(n)$ -isolating  $(K, \ell(n))$ -cover. In this section, we give the following result.

**THEOREM 6.1.** *If the graph  $G$  is  $f$ -renitent, then any leader election protocol takes  $\Omega(f)$  expected steps to stabilize on  $G$ .*

Our approach is inspired by the lower bound construction for randomized leader election in synchronous message-passing models by Kutten et al. [35, Theorem 3.13]. However, in the population model communication is both stochastic and asynchronous with sequential interactions, so we need to further refine the approach to make it work in our setting. We prove our result in a stronger variant of the population model: we do not restrict the number of states used by the nodes and give each node access to its own (independent and infinite) sequence of random bits.

Formally, we assume that each node  $v \in V$  is given as input a random value  $y(v)$  sampled independently and uniformly at random from the unit interval  $[0, 1)$ . Since we do not restrict the state space of the nodes, the nodes can locally store this value to access an infinite sequence of i.i.d. random bits. The random bits assign nodes unique identifiers with probability 1. Any protocol that does not use these random bits can ignore them.

*Outline of the proof.* Suppose  $G$  is  $f$ -renitent and  $\mathcal{A}$  is leader election protocol on  $G$  that stabilizes in  $T$  steps. Without loss of generality, assume  $f(n) \geq 6$ , as otherwise the claim of Theorem 6.1 is trivially true. Fix any  $f(n)$ -isolating  $(K, \ell)$ -cover  $C = \{V_0, \dots, V_{K-1}\}$  of the graph  $G$ , where  $K$  is a constant independent of  $n$ , and let  $Y = Y(C)$  be the isolation time of the cover. Let  $X \sim \text{Poisson}(\lambda)$  be a Poisson random variable with mean  $\lambda = f(n)/2$  and  $\mathcal{E}$  be the event that  $X < Y$ . Intuitively,  $X$  represents a random time step (independent of  $Y$ ) at which we investigate the state of the system. To this end, we define  $L_i$  to be the event that some node  $v \in V_i$  outputs that it is a leader at step  $X$ .

LEMMA 6.2. *The following hold:*

- (a)  $\Pr[L_0 \mid \mathcal{E}] = \Pr[L_i \mid \mathcal{E}]$  for each  $0 \leq i < K$ , and
- (b)  $\Pr[L_i \cap L_j \mid \mathcal{E}] = \Pr[L_i \mid \mathcal{E}] \cdot \Pr[L_j \mid \mathcal{E}]$  for some  $0 \leq i < j < K$ .

LEMMA 6.3. *There exists a constant  $C(K) > 0$  such that the stabilization time  $T$  of protocol  $\mathcal{A}$  satisfies  $\Pr[T > X] \geq C(K)$ .*

These two lemmas together imply our main result of this section.

THEOREM 6.1. *If the graph  $G$  is  $f$ -renitent, then any leader election protocol takes  $\Omega(f)$  expected steps to stabilize on  $G$ .*

*Constructing renitent graphs.* We now give examples of  $f$ -renitent graphs; by Theorem 6.1 the expected stabilization time on these graphs will be  $\Omega(f)$ . For example, it is not hard to see that cycles are  $\Omega(n^2)$ -renitent: we can split the cycle into four paths  $V_0, \dots, V_3$  of length roughly  $n/4$  and information propagation from set  $V_1$  to  $V_3$  requires  $\Omega(n^2)$  steps with constant probability. In fact, for any constant  $k > 0$ , the idea generalizes to higher dimensions:  $k$ -dimensional toroidal grids are  $\Omega(n^{1+1/k})$ -renitent; one can partition such grids into constantly many subcubes of diameter  $\Theta(n^{1/k})$  and observe that propagating information to distance  $D$  in regular graphs requires  $\Omega(Dn)$  steps with constant probability.

The next lemma allows us to obtain  $\Omega(Dm)$ -renitent graphs for essentially any  $D$  and  $m$ .

LEMMA 6.4. *Let  $G$  be a connected graph with  $n$  nodes,  $m$  edges and diameter  $D$ . For any integer  $\ell$  such that  $D \leq \ell \leq n$ , there exists an  $\Omega(\ell m)$ -renitent graph  $G'$  with  $\Theta(n)$  nodes,  $\Theta(m)$  edges and diameter  $\Theta(\ell)$ . In addition,  $\mathbf{B}(G') \in \Omega(\ell m)$ .*

This lemma can be used to show the following lower bound.

THEOREM 6.5. *For any increasing function  $T: \mathbb{N} \rightarrow \mathbb{N}$  such that  $n \log n \leq T(n) \leq n^3$ , there is an infinite family of graphs in which leader election takes  $\Theta(T(n))$  expected steps and the broadcast time satisfies  $\mathbf{B}(G) \in \Theta(T)$ .*

PROOF. For any  $N \geq 1$ , we construct a graph  $G$  with  $n \geq N$  nodes as follows. We distinguish two cases. First, if  $T \in \omega(n^2 \log n)$ , then apply Lemma 6.4 with a clique  $H$  of size  $N$  and  $\ell = \lceil T(N)/N^2 \rceil$ . Otherwise, if  $T \in O(n^2 \log n)$ , then set  $\ell = \lceil \log N + T(N)/(N \log N) \rceil$ , take a star graph and add  $\Theta(T(N)/\ell)$  edges in an arbitrary fashion to obtain the graph  $H$ . Adding this many edges is always possible since  $T(N)/\ell \in O(N^2)$ . In both cases, apply Lemma 6.4 with  $H$  and  $\ell$  to obtain a graph  $G$ . Note that Lemma 6.4 implies that the graph  $G$  will be  $\Omega(T)$ -renitent and satisfy  $\mathbf{B}(G) \in \Omega(T)$ . By Theorem 6.1 leader election will take  $\Omega(T)$  expected steps on this graph. The graph  $H$  has constant diameter, and since  $\ell \in \Omega(\log n)$ ,  $G$  has diameter  $\Omega(\log n)$ . By construction,  $Dm \in \Theta(T)$ , and therefore, Theorem 3.2 implies that  $\mathbf{B}(G) \in O(T)$ . Now Theorem 4.2 implies the upper bound for leader election time.  $\square$

## 6.2 A lower bound for dense graphs

The above gives graph families in which expected leader election and broadcast time are of the same order. However, this is not generally true. Leader election time can be much lower than broadcast time in graphs, where the local structure helps break symmetry fast. The star graph (i.e., a tree of depth one) is the simplest example: there is a trivial constant-state protocol that elects a leader in one interaction, but broadcast time in a star is  $\Theta(n \log n)$  by a simple coupon collector argument. This rules out the existence of a general  $\Omega(n \log n)$  lower bound for leader election in sparse graphs. In the full version, we show that in dense graphs with sufficiently high minimum degrees, we cannot easily exploit local graph structure to break symmetry fast.

THEOREM 6.6. *Let  $0 < \lambda < 1$  and  $0 < \phi < 1$  be constants. Suppose  $G$  has minimum degree  $\delta \geq \lambda n^\phi$  and at least  $m \geq \lambda n^2$  edges. Then any leader election protocol on  $G$  requires  $\Omega(n \log n)$  expected steps to stabilize.*

At its core, the argument is an extension of lower bound result of Sudo and Masuzawa [40] from cliques to general high-degree graphs. However, to deal with the general structure of the interaction graph, we introduce the two new concepts: multigraphs of influencers and leader generating interaction patterns.

We assume  $G$  is as in the above theorem. Our proof strategy is roughly as follows. We assume that there is a fast protocol that stabilizes in  $o(n \log n)$  steps. First, we aim to capture the spatial structure of the part of the graph that influenced a node  $v$  to be elected as a leader; we call this structure “leader generating interaction pattern”. We show that this structure is fairly small and almost tree-like. We show that this can be unfolded into a larger tree, without growing the tree too much. Then we argue that because the graph has high degrees, a tree of this size can be found in the set of nodes that have not interacted yet.

## 7 LOWER BOUND FOR CONSTANT-STATE PROTOCOLS

In this section, we give lower bounds for constant-state protocols that stabilize in finite expected time on any connected graph. In the clique, the classic approach has been to utilize the so-called *surgeries technique* of Doty and Soloveichik [24], later extended by Alistarh et al. [3] to show lower bounds for super-constant state protocols. Roughly, surgeries consists of carefully “stitching together” transition sequences, in order to completely eliminate states whose count decreases too fast (e.g., the leader state), thus resulting in incorrect executions (e.g., executions without a leader).

When moving beyond cliques, applying surgeries is difficult. The key challenge is that in addition to keeping track of the counts of states, we also need to control for the *spatial distribution* of generated states in order to determine if a given configuration is stable. For example, if we know that an interaction  $(a, b) \rightarrow (c, d)$  produces a leader, then in the case of a clique it suffices to check if states  $a$  and  $b$  are present in the overall population to determine if this interaction can produce a new leader. However, in the case of general interaction graphs, the rule  $(a, b) \rightarrow (c, d)$  can only produce a leader if some nodes with states  $a$  and  $b$  in the configuration are *adjacent*.

We circumvent this obstacle by considering a *random graph setting*, where the interaction graph itself is *probabilistic*. Instead of showing a lower bound for a given graph, we give a lower bound that holds in *most* graphs, where “most” is interpreted as having graphs coming from a certain probability distribution. We will focus on the Erdős–Rényi random graph model  $G_{n,p}$ . As we are only interested in connected graphs  $G$ , we adopt the convention that stabilization time  $T_{\mathcal{P}}(G) = \infty$ , if  $G$  is disconnected. Our main result is the following.

**THEOREM 7.1.** *Suppose  $\mathcal{P}$  is a protocol that stabilizes on any connected graph and whose state transition function is independent of the communication graph. Fix a constant  $p > 0$  and let  $G \sim G_{n,p}$ . Then the stabilization time  $T_{\mathcal{P}}(G)$  of  $\mathcal{P}$  on  $G$  satisfies*

$$\mathbb{E}[T_{\mathcal{P}}(G) \mid G \text{ is connected}] \in \Omega(n^2).$$

This result generalizes the lower bound of Doty and Soloveichik [24] from cliques to dense random graphs. As such, we follow a similar approach, but provide new ideas to deal with the structure of the interaction graph. First, we show that any protocol starting from a uniform initial configuration, passes through a “fully dense” configuration with very high probability on a sufficiently-dense Erdős–Rényi random graph. Second, we show that if the protocol stabilizes too fast, then there exist reachable configurations with many states in low count. Finally, we use surgeries to show that such a protocol must fail even on the clique.

## 8 CONCLUSIONS

We have performed the first focused investigation of time-space trade-offs in the complexity of leader election on general graphs, in the population model. We have provided some of the first time and space-efficient protocols for leader election, and the first bounds that are tight within logarithmic factors. We introduced “graphical” variants of classic techniques, such as information dissemination

and approximate phase clocks on the upper bound side, and indistinguishability and surgeries for lower bounds.

Our work leaves open the question of *tight* bounds for both space and time complexity on general graph families, particularly in the case of sparse graphs. Another direction is considering other fundamental problems, such as majority, in the same setting, for which our techniques should prove useful.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. We gratefully acknowledge funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 805223 ScaleML).

## REFERENCES

- [1] Hüseyin Acan, Andrea Collecchio, Abbas Mehrabian, and Nick Wormald. 2017. On the push & pull protocol for rumor spreading. *SIAM Journal on Discrete Mathematics* 31, 2 (2017), 647–668. <https://doi.org/10.1137/15M1033113>
- [2] Romas Aleliunas, Richard M Karp, Richard J Lipton, László Lovász, and Charles Rackoff. 1979. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science (FOCS 1979)*. IEEE Computer Society, 218–223. <https://doi.org/10.1109/SFCS.1979.34>
- [3] Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L Rivest. 2017. Time-space trade-offs in population protocols. In *Proc. 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017)*. 2560–2579. <https://doi.org/10.5555/3039686.3039855>
- [4] Dan Alistarh and Rati Gelashvili. 2015. Polylogarithmic-Time Leader Election in Population Protocols. In *Proc. 42nd International Colloquium on Automata, Languages, and Programming (ICALP 2015)*. 479–491. [https://doi.org/10.1007/978-3-662-47666-6\\_38](https://doi.org/10.1007/978-3-662-47666-6_38)
- [5] Dan Alistarh and Rati Gelashvili. 2018. Recent Algorithmic Advances in Population Protocols. *SIGACT News* 49, 3 (2018), 63–73. <https://doi.org/10.1145/3289137.3289150>
- [6] Dan Alistarh, Rati Gelashvili, and Joel Rybicki. 2021. Fast graphical population protocols. In *Proc. 25th International Conference on Principles of Distributed Systems (OPODIS)*. <https://arxiv.org/abs/2102.08808>
- [7] Dan Alistarh, Joel Rybicki, and Sasha Voitovych. 2022. Near-Optimal Leader Election in Population Protocols on Graphs (Full Online Version). <https://arxiv.org/abs/2205.12597>
- [8] Dana Angluin. 1980. Local and global properties in networks of processors. In *Proc. of the 12th Annual ACM Symposium on Theory of Computing (STOC)*. 82–93.
- [9] Dana Angluin, James Aspnes, Melody Chan, Michael J Fischer, Hong Jiang, and René Peralta. 2005. Stably computable properties of network graphs. In *International Conference on Distributed Computing in Sensor Systems*. Springer, 63–74. [https://doi.org/10.1007/11502593\\_8](https://doi.org/10.1007/11502593_8)
- [10] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J Fischer, and René Peralta. 2006. Computation in networks of passively mobile finite-state sensors. *Distributed Computing* 18, 4 (2006), 235–253. <https://doi.org/10.1007/s00446-005-0138-3>
- [11] Dana Angluin, James Aspnes, and David Eisenstat. 2008. Fast computation by population protocols with a leader. *Distributed Computing* 21, 3 (2008), 183–199. <https://doi.org/10.1007/s00446-008-0067-z>
- [12] Dana Angluin, James Aspnes, Michael J Fischer, and Hong Jiang. 2008. Self-stabilizing population protocols. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 3, 4 (2008), 1–28. <https://doi.org/10.1145/1452001.1452003>
- [13] James Aspnes and Eric Ruppert. 2009. An introduction to population protocols. In *Middleware for Network Eccentric and Mobile Applications*. Springer, 97–120.
- [14] Joffroy Beauquier, Peva Blanchard, and Janna Burman. 2013. Self-stabilizing leader election in population protocols over arbitrary communication graphs. In *International Conference on Principles of Distributed Systems (OPODIS)*. Springer, 38–52. <https://hal.archives-ouvertes.fr/hal-00867287v2>
- [15] Petra Berenbrink, George Giakkoupis, and Peter Kling. 2020. Optimal time and space leader election in population protocols. In *Proc. 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC 2020)*. 119–129. <https://doi.org/10.1145/3357713.3384312>
- [16] Petra Berenbrink, Dominik Kaaser, Peter Kling, and Lena Otterbach. 2018. Simple and Efficient Leader Election. In *1st Symposium on Simplicity in Algorithms (SOSA 2018)*, Vol. 61. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 9:1–9:11. <https://doi.org/10.4230/OAS18.SOSA.2018.9>
- [17] Andrei Broder and Anna Karlin. 1989. Bounds on the cover time. *Journal of Theoretical Probability* 2 (06 1989), 101–120. <https://doi.org/10.1007/BF01048273>

- [18] Clément Canonne. 2019. A short note on Poisson tail bounds. <http://www.cs.columbia.edu/~ccanonne/files/misc/2017-poissonconcentration.pdf> Accessed January 27, 2022.
- [19] Hsueh-Ping Chen and Ho-Lin Chen. 2019. Self-stabilizing leader election. In *Proc. ACM Symposium on Principles of Distributed Computing (PODC)*. 53–59. <https://doi.org/10.1145/3293611.3331616>
- [20] Hsueh-Ping Chen and Ho-Lin Chen. 2020. Self-Stabilizing Leader Election in Regular Graphs. In *Proc. 39th Symposium on Principles of Distributed Computing (PODC '20)*. Association for Computing Machinery, New York, NY, USA, 210–217. <https://doi.org/10.1145/3382734.3405733>
- [21] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2010. Almost tight bounds for rumour spreading with conductance. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*. 399–408. <https://doi.org/10.1145/1806689.1806745>
- [22] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2010. Rumour spreading and graph conductance. In *Proc. 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1657–1663. <https://doi.org/10.1137/1.9781611973075.135>
- [23] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. 1987. Epidemic algorithms for replicated database maintenance. In *Proc. 6th Annual ACM Symposium on Principles of Distributed Computing (PODC)*. 1–12. <https://doi.org/10.1145/41840.41841>
- [24] David Doty and David Soloveichik. 2018. Stable leader election in population protocols requires linear time. *Distributed Computing* 31, 4 (2018), 257–271.
- [25] Robert Elsässer and Tomasz Radzik. 2018. Recent Results in Population Protocols for Exact Majority and Leader Election. *Bulletin of the EATCS* 126 (2018). <http://bulletin.eatcs.org/index.php/beatcs/article/view/549/546>
- [26] Nazim Fatès and Lucas Gerin. 2008. Examples of fast and slow convergence of 2D asynchronous cellular systems. In *International Conference on Cellular Automata*. Springer, 184–191. [https://doi.org/10.1007/978-3-540-79992-4\\_24](https://doi.org/10.1007/978-3-540-79992-4_24)
- [27] Leszek Gąsieniec and Grzegorz Stachowiak. 2020. Enhanced Phase Clocks, Population Protocols, and Fast Space Optimal Leader Election. *Journal of the ACM (JACM)* 68, 1 (2020), 1–21. <https://doi.org/10.1145/3424659>
- [28] George Giakkoupis. 2011. Tight bounds for rumor spreading in graphs of a given conductance. In *28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 9)*, Thomas Schwentick and Christoph Dürr (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 57–68. <https://doi.org/10.4230/LIPIcs.STACS.2011.57>
- [29] George Giakkoupis, Yasamin Nazari, and Philipp Woelfel. 2016. How asynchrony affects rumor spreading time. In *Proc. 2016 ACM Symposium on Principles of Distributed Computing (PODC)*. 185–194. <https://doi.org/10.1145/2933057.2933117>
- [30] Leszek Gąsieniec and Grzegorz Stachowiak. 2018. Fast space optimal leader election in population protocols. In *Proc. 29th ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*. <https://doi.org/10.1137/1.9781611975031.169>
- [31] Leszek Gąsieniec, Grzegorz Stachowiak, and Przemysław Uznański. 2019. Almost Logarithmic-Time Space Optimal Leader Election in Population Protocols. In *Proc. 31st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2019)*. <https://doi.org/10.1145/3323165.3323178>
- [32] Svante Janson. 2018. Tail bounds for sums of geometric and exponential variables. *Statistics & Probability Letters* 135 (2018), 1–6. <https://doi.org/10.1016/j.spl.2017.11.017>
- [33] Jeff D Kahn, Nathan Linial, Noam Nisan, and Michael E Saks. 1989. On the cover time of random walks on graphs. *Journal of Theoretical Probability* 2, 1 (1989), 121–128. <https://doi.org/10.1007/BF01048274>
- [34] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. 2000. Randomized rumor spreading. In *Proc. 41st Annual Symposium on Foundations of Computer Science (FOCS 2000)*. 565–574.
- [35] Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson, and Amitabh Trehan. 2015. On the complexity of universal leader election. *Journal of the ACM (JACM)* 62, 1 (2015), 1–27. <https://doi.org/10.1145/2699440>
- [36] Thomas M. Liggett. 2005. *Interacting Particle Systems*. Springer-Verlag, Berlin, Heidelberg. <https://doi.org/10.1007/b138374>
- [37] Bertrand Ottino-Löffler, Jacob G Scott, and Steven H Strogatz. 2017. Takeover times for a simple model of network infection. *Physical Review E* 96, 1 (2017), 012313. <https://doi.org/10.1103/PhysRevE.96.012313>
- [38] David Peleg. 2000. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics.
- [39] Thomas Sauerwald. 2010. On mixing and edge expansion properties in randomized broadcasting. *Algorithmica* 56, 1 (2010), 51–88. <https://doi.org/10.1007/s00453-008-9245-4>
- [40] Yuichi Sudo and Toshimitsu Masuzawa. 2020. Leader election requires logarithmic time in population protocols. *Parallel Processing Letters* 30, 01 (2020), 2050005. <https://doi.org/10.1142/S012962642050005X>
- [41] Yuichi Sudo, Toshimitsu Masuzawa, Ajay K Datta, and Lawrence L Larmore. 2016. The same speed timer in population protocols. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 252–261.
- [42] Yuichi Sudo, Fukuhito Ooshita, Taisuke Izumi, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. 2019. Logarithmic expected-time leader election in population protocol model. In *Proc. International Symposium on Stabilizing, Safety, and Security of Distributed Systems (SSS 2019)*. 323–337. [https://doi.org/10.1007/978-3-030-34992-9\\_26](https://doi.org/10.1007/978-3-030-34992-9_26)
- [43] Yuichi Sudo, Fukuhito Ooshita, Taisuke Izumi, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. 2020. Time-Optimal Leader Election in Population Protocols. *IEEE Transactions on Parallel and Distributed Systems* 31, 11 (2020), 2620–2632. <https://doi.org/10.1109/TPDS.2020.2991771>
- [44] Yuichi Sudo, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. 2020. Loosely stabilizing leader election on arbitrary graphs in population protocols without identifiers or random numbers. *IEICE Transactions on Information and Systems* 103, 3 (2020), 489–499.
- [45] Yuichi Sudo, Masahiro Shibata, Junya Nakamura, Yonghwan Kim, and Toshimitsu Masuzawa. 2021. Self-stabilizing Population Protocols with Global Knowledge. *IEEE Transactions on Parallel and Distributed Systems* 32, 12 (2021), 3011–3023. <https://doi.org/10.1109/TPDS.2021.3076769>
- [46] Daisuke Yokota, Yuichi Sudo, and Toshimitsu Masuzawa. 2020. Time-optimal self-stabilizing leader election on rings in population protocols. In *International Symposium on Stabilizing, Safety, and Security of Distributed Systems*. Springer, 301–316.