

# Leader Election Requires Logarithmic Time in Population Protocols

Yuichi Sudo\* and Toshimitsu Masuzawa†

Graduate School of Information Science and Technology, Osaka University, Japan

## Abstract

This paper shows that every leader election protocol requires logarithmic stabilization time both in expectation and with high probability in the population protocol model. This lower bound holds even if each agent has knowledge of the exact size of a population and is allowed to use an arbitrarily large number of agent states. This lower bound concludes that the protocol given in [Sudo et al., SSS 2019] is time-optimal in expectation.

## 1 Introduction

We consider the *population protocol* (PP) model [1] in this paper. A network called *population* consists of a large number of automata, called *agents*. Agents make *interactions* (i.e., pairwise communication) with each other by which they update their states. Agents are strongly anonymous: they do not have identifiers and they cannot distinguish their neighbors with the same state. As with the majority of studies on population protocols [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], we assume that the network of agents is a complete graph and that the scheduler selects an interacting pair of agents at each step uniformly at random.

In this paper, we focus on the leader election problem, which is one of the most fundamental and well studied problems in the PP model. The leader election problem requires that starting from a specific initial configuration, a population reaches a safe configuration in which exactly one leader exists and the population keeps that unique leader thereafter.

### 1.1 Related Work

There have been many works which study the leader election problem in the PP model (Tables 1 and 2). Angluin et al. [1] gave the first leader election protocol, which stabilizes in  $O(n)$  parallel time in expectation and uses only constant space of each agent, where  $n$  is the number of agents and “parallel time” means the number of steps divided by  $n$ . If we stick to constant space, this linear parallel time is optimal; Doty and Soloveichik [8] showed that any constant space protocol requires linear parallel time to elect a unique leader. Alistarh and Gelashvili [2] made a breakthrough in 2015; they achieved poly-logarithmic stabilization time ( $O(\log^3 n)$  parallel time) by increasing the number of states from  $O(1)$  to only  $O(\log^3 n)$ . Thereafter, the stabilization time has been improved by many studies [11, 4, 5, 6, 7]. Gąsieniec et al. [6] gave a state-of-art protocol that stabilizes in  $O(\log n \cdot \log \log n)$  parallel time with only  $O(\log \log n)$  states. Its space complexity is optimal; Alistarh et al. [3] showed that any poly-logarithmic parallel time algorithm requires  $\Omega(\log \log n)$  states. Michail et al. [7] gave a protocol with  $O(\log n)$  parallel time but with

\*Corresponding Author: y-sudou@ist.osaka-u.ac.jp

†masuzawa@ist.osaka-u.ac.jp

Table 1: Leader Election Protocols (Stabilization time is shown in terms of expected parallel time)

	States	Stabilization Time
[1]	$O(1)$	$O(n)$
[2]	$O(\log^3 n)$	$O(\log^3 n)$
[3]	$O(\log^2 n)$	$O(\log^{5.3} n \cdot \log \log n)$
[4]	$O(\log n)$	$O(\log^2 n)$
[5]	$O(\log \log n)$	$O(\log^2 n)$
[6]	$O(\log \log n)$	$O(\log n \cdot \log \log n)$
[7]	$O(n)$	$O(\log n)$
[12]	$O(\log n)$	$O(\log n)$

Table 2: Lower Bounds for Leader Election (Stabilization time is shown in terms of expected parallel time)

	States	Stabilization Time
[8]	$O(1)$	$\Omega(n)$
[3]	$< 1/2 \log \log n$	$\Omega(n/\text{polylog} n)$
This work	any large	$\Omega(\log n)$

a linear number of states. Our previous work [12] gave a protocol with  $O(\log n)$  parallel time and  $O(\log n)$  states. Those protocols with non-constant number of states [2, 3, 11, 4, 5, 6] are not *uniform*, that is, they require some rough knowledge of  $n$ . For example, in the protocol of [5], a  $\Theta(\log \log n)$  value must be hard-coded to set the maximum value of one variable (named  $l$  in that paper). One can find detailed information about the leader election in the PP model in two survey papers [13, 14].

There is a folklore that any leader election protocol requires  $\Omega(\log n)$  parallel time in the population protocol model. One may think that this lower bound trivially holds because several agents have no interactions during  $o(\log n)$  parallel time with probability  $1 - o(1)$ . However, as Alistarh and Gelashvili [2] pointed out, this idea is not sufficient to prove the folklore. Let us discuss it in detail here. The lower bound of  $\Omega(\log n)$  expected parallel time holds almost trivially if the initial output of the agents is  $L$  (i.e., all the agents are leaders initially). This is because we need  $\Omega(\log n)$  expected parallel time before  $n - 1$  agents have at least one interaction each. What if the initial output is  $F$  (i.e., all the agents are non-leaders initially)? For any small constant  $\epsilon$ , we can prove that with a constant probability,  $\Omega(n^{1-\epsilon})$  agents remains still *inexperienced* after the first period of  $o(\log n)$  parallel time in an execution, that is, they have no interactions during the period. However, this does not immediately mean that  $\Omega(\log n)$  parallel time is necessary to elect a leader in expectation because those  $\Omega(n^{1-\epsilon})$  inexperienced agents are non-leaders. We have to show that no leader election protocol can create a unique leader with  $o(\log n)$  expected parallel time starting from the initial configuration where all agents are non-leaders. To the best of our knowledge, there is no proof in the literature for this folklore, that is, the lower bound of  $\Omega(\log n)$  parallel time on the stabilization time for leader election.

## 1.2 Our Contribution

In this paper, we prove the above folklore, that is, we show that any leader election protocol requires  $\Omega(\log n)$  parallel time in expectation. As mentioned above, most of recent protocols uses a non-constant (poly-logarithmic, in most cases) number of states and assume that rough knowledge of the population size is given to each agent. This lower bound holds even if each agent can use an arbitrarily large number of states and knows the exact size of a population. Thus, by this lower bound, we can say that the protocols of [7] and [12] are optimal in terms of expected stabilization time.

In our proof for the lower bound, we do not assume that every leader election protocol always stabilizes to elect a unique leader. Therefore, our lower bound holds even if we allow a protocol to have a (small) probability that it fails to elect a unique leader.

Strictly speaking, we give a stronger lower bound than  $\Omega(\log n)$  parallel stabilization time *in expectation*. Instead, we show that every leader election protocol requires  $\Omega(\log n)$  parallel time to stabilize with probability  $1 - o(1)$ . This lower bound immediately gives the above lower bound in expectation. Moreover, it immediately yields that no leader election protocol stabilizes within  $o(\log n)$  parallel time with high probability; every leader election protocol stabilizes within  $o(\log n)$  parallel time with probability  $o(1)$ .

To prove the lower bound, we introduce a novel notion that we call *influencers*. At any time of an execution, the influencers of an agent  $v$  is the set of agents that could influence on the current state of  $v$ . The size of the influencers is monotonically non-decreasing, and grows with the same speed as *epidemics*, which Angluin et al. [15] introduced in order to analyze fast protocols to compute any semi-linear predicate. Actually, we will prove the lower bound essentially by showing that  $\Omega(\log n)$  parallel time is necessary for the number of influencers of any agent  $v$  to reach  $\Omega(n^{2/3})$ .

## 2 Preliminaries

In this section, we specify the population protocol model. For simplicity, we omit some elements of the population protocol model that are not needed to study leader election. Specifically, we remove input symbols and input functions from the definition of population protocols.

A *population* is a network consisting of *agents*. We denote the set of all the agents by  $V$  and let  $n = |V|$ . We assume that a population is a complete graph, thus every pair of agents  $(u, v)$  can interact, where  $u$  serves as the *initiator* and  $v$  serves as the *responder* of the interaction.

A *protocol*  $P(Q, s_{\text{init}}, T, Y, \pi_{\text{out}})$  consists of a finite set  $Q$  of agent states, an initial state  $s_{\text{init}} \in Q$ , a transition function  $T : Q \times Q \rightarrow Q \times Q$ , a finite set  $Y$  of output symbols, and an output function  $\pi_{\text{out}} : Q \rightarrow Y$ . Every agent is in state  $s_{\text{init}}$  when an execution of protocol  $P$  begins. When two agents interact,  $T$  determines their next states according to their current states. The *output* of an agent is determined by  $\pi_{\text{out}}$ : The output of an agent in state  $q$  is  $\pi_{\text{out}}(q)$ . As with all papers listed in Table 1 except for [1], we assume that a rough knowledge of  $n$  is available. Specifically, we assume that an integer  $m$  such that  $m \geq \log_2 n$  and  $m = \Theta(\log n)$  is given, thus we can design  $P(Q, s_{\text{init}}, T, Y, \pi_{\text{out}})$  using this input  $m$ , i.e.,  $Q, s_{\text{init}}, T, Y$ , and  $\pi_{\text{out}}$  can depend on  $m$ .

A *configuration* is a mapping  $C : V \rightarrow Q$  that specifies the states of all the agents. We define  $C_{\text{init}, P}$  as the configuration of  $P$  where every agent is in state  $s_{\text{init}}$ . We say that a configuration  $C$  changes to  $C'$  by the interaction  $e = (u, v)$ , denoted by  $C \xrightarrow{e} C'$ , if  $(C'(u), C'(v)) = T(C(u), C(v))$  and  $C'(w) = C(w)$  for all  $w \in V \setminus \{u, v\}$ .

A *schedule*  $\gamma = \gamma_0, \gamma_1, \dots = (u_0, v_0), (u_1, v_1), \dots$  is a sequence of interactions. A schedule determines which interaction occurs at each *step*, i.e., interaction  $\gamma_t$  happens at step  $t$  under schedule  $\gamma$ . We consider a *uniformly random scheduler*  $\Gamma = \Gamma_0, \Gamma_1, \dots$  where each  $\Gamma_t$  ( $t \geq 0$ ) is a random variable that specifies the interaction  $(u_t, v_t)$  at step  $t$  and satisfies  $\Pr(\Gamma_t = (u, v)) = \frac{1}{n(n-1)}$  for any distinct  $u, v \in V$ . Given a schedule  $\gamma = \gamma_0, \gamma_1, \dots$ , the *execution* of protocol  $P$  starting from a configuration  $C_0$  is uniquely defined as  $\Xi_P(C_0, \gamma) = C_0, C_1, \dots$  such that  $C_t \xrightarrow{\gamma_t} C_{t+1}$  for all  $t \geq 0$ . We usually focus on  $\Xi_P(C_{\text{init}, P}, \Gamma)$ . We say that agent  $v \in V$  *participates* in  $\Gamma_t$  if  $v$  is either the initiator or the responder of  $\Gamma_t$ . We say that a configuration  $C$  of protocol  $P$  is *reachable* if the initial configuration  $C_{\text{init}, P}$  changes to  $C$  by some finite sequence of interactions  $\gamma_0, \gamma_1, \dots, \gamma_k$ . We define  $\mathcal{C}_{\text{all}}(P)$  as the set of all reachable configurations of  $P$ .

The leader election problem requires that every agent should output  $L$  or  $F$  which means “leader” or “follower” respectively. Let  $\mathcal{S}_{P, LE}$  be the set of the configurations of  $P$  such that each  $C \in \mathcal{S}_{P, LE}$  satisfies the following:

- exactly one agent outputs  $L$  (i.e., is a leader) in  $C$ , and
- no agent changes its output in the execution  $\Xi_P(C, \gamma)$  for any schedule  $\gamma$ .

We call the configurations of  $\mathcal{S}_{P,LE}$  the *safe* configurations of  $P$ . We say that an execution of  $P$  *stabilizes* when it reaches a configuration in  $\mathcal{S}_{P,LE}$ . For any leader election protocol  $P$ , we define the *stabilization time* of  $P$  as the number of steps during which execution  $\Xi_P(C_{\text{init},P}, \mathbf{\Gamma})$  reaches a configuration in  $\mathcal{S}_{P,LE}$ , divided by the number of agents  $n$ . The division by  $n$  implies that we evaluate the stabilization time in terms of parallel time. Since  $\mathbf{\Gamma}$  is a random variable, the stabilization time of  $P$  is also a random variable. Thus, we usually evaluate it in terms of “in expectation” or “with high probability”.

### 3 Lower Bound

Let  $P(Q, s_{\text{init}}, T, Y, \pi_{\text{out}})$  be any leader election protocol. We fix protocol  $P$  and its execution  $\Xi = \Xi_P(C_{\text{init},P}, \mathbf{\Gamma}) = C_0, C_1, \dots$  throughout this section. We call  $C_t$  the configuration at step  $t$  or  $t$ -th configuration. Note that each  $C_t$  is a random variable. Our goal is to prove the following proposition.

**Proposition 1.** *For some constant  $c$ , the (parallel) stabilization time of  $P$  is at least  $c \ln n$  with probability  $1 - o(1)$ .*

We prove Proposition 1 in the rest of this section. First, we prove the following lemma in a similar way as a standard analysis of the coupon collector’s problem.

**Lemma 1.** *Let  $\epsilon$  be any (small) positive constant and  $f(n)$  be any function such that  $f(n) = O(n^{1-\epsilon})$ . There exists some constant  $c$  such that execution  $\Xi$  requires at least  $cn \ln n$  steps with probability  $1 - o(1)$  to reach a configuration where less than  $f(n)$  agents are in state  $s_{\text{init}}$ .*

*Proof.* Without loss of generality, we assume that an agent never gets state  $s_{\text{init}}$  once it has an interaction. (A transition going back to  $s_{\text{init}}$  just increases the probability that  $\Xi$  requires  $\Omega(n \log n)$  steps to reach a configuration with less than  $f(n)$  agents in state  $s_{\text{init}}$ .) Consider a configuration that exactly  $i$  agents are in  $s_{\text{init}}$ . Then, at least one of the  $i$  agents has an interaction and leaves state  $s_{\text{init}}$  in the next step with probability  $p_i = \frac{iC_2 + i(n-i)}{nC_2} = \frac{i(2n-i-1)}{n(n-1)}$ . Let  $X_i$  be a geometric random variable with parameter  $p_i$ , that is, the number of coin flips until it lands on heads where the coin lands on heads with probability  $p_i$  in each flip. Let  $X = \sum_{i \in \{f^*(n), f^*(n)+2, \dots, n\}} X_i$  where  $f^*(n) = 2\lceil f(n)/2 \rceil$ . Since  $p_n = p_{n-1} = 1$  and  $p_i$  is monotonically increasing in  $i \in [0, n-1]$ , for any integer  $a$ , the probability that  $\Xi$  requires at least  $a$  steps to reach a configuration with less than  $f(n)$  agents in state  $s_{\text{init}}$  is lower bounded by  $\Pr(X \geq a)$ . Thus, it suffices to show  $\Pr(X \geq cn \log n) = 1 - o(1)$  for some constant  $c$ .

In what follows, we analyze the expectation and the variance of  $X$  and then obtain  $\Pr(X \geq cn \log n) = 1 - o(1)$  by Chebyshev’s inequality. We obtain the lower bounds of the expectation and the variance as follows:

$$\begin{aligned} \mathbf{E}[X] &= \sum_{i \in \{f^*(n), f^*(n)+2, \dots, n\}} \frac{1}{p_i} \geq \sum_{i \in \{f^*(n), f^*(n)+2, \dots, n\}} \frac{n}{2i} = \Omega\left(n \log \frac{n}{f(n)}\right) = \Omega(n \log n^\epsilon) = \Omega(n \log n), \\ \mathbf{Var}[X] &= \sum_{i \in \{f^*(n), f^*(n)+2, \dots, n\}} \frac{1-p_i}{p_i^2} \leq \sum_{i=1,2,\dots,n} \frac{1}{p_i^2} \leq \sum_{i=1,2,\dots,\infty} \frac{n^2}{i^2} = \frac{\pi^2 n^2}{6} < 2n^2, \end{aligned}$$

where we use  $1^2 + 2^2 + 3^2 + \dots = \pi^2/6$  for the last equality. Let  $d$  be a constant such that  $\mathbf{E}[X] \geq dn \ln n$  holds for any sufficiently large  $n$ . Then, by Chebyshev’s Inequality, we obtain

$$\Pr\left(X \leq \frac{dn \ln n}{2}\right) \leq \Pr\left(X \leq \mathbf{E}[X] - \frac{dn \ln n}{2}\right) \leq \frac{4\mathbf{Var}[X]}{(dn \ln n)^2} = O(1/\log^2 n).$$

Thus, we have  $\Pr(X > dn \ln n/2) = 1 - O(1/\log^2 n) = 1 - o(1)$ . □

**Corollary 1.** *Proposition 1 holds if the initial output of  $P$  is  $L$ , i.e.,  $\pi_{\text{out}}(s_{\text{init}}) = L$ .*

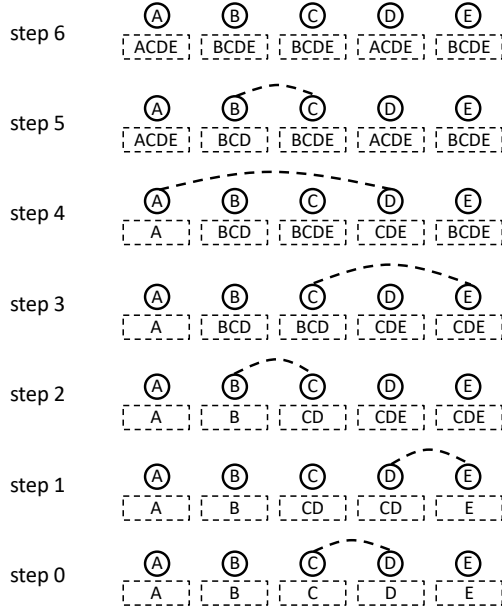


Figure 1: An example of the influencers. The circles represent the agents. The dashed lines represents the interactions in steps 0, 1, ..., 5. The box below each circle represents the set of influencers of the corresponding agent at each step.

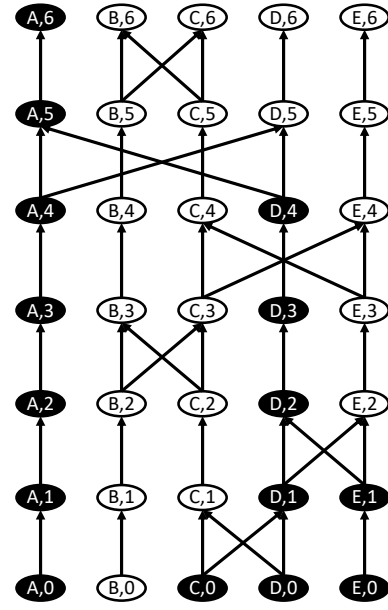


Figure 2: The graph  $H$  that corresponds to the interaction sequence in Figure 1. The black ellipses represent the nodes from which  $(A, 6)$  is reachable.

In the rest of this section, we assume  $\pi_{\text{out}}(s_{\text{init}}) = F$ . Recall that a configuration  $C$  of  $P$  is safe if and only if there exists exactly one leader in  $C$  and no agent can change its output in an execution after  $C$ . In what follows, we use Lemma 1 by letting  $f(n) = n^{2/3}$  while the lemmas and corollaries in the rest of this section hold for more general  $f(n) = O(n^{1-\epsilon})$ . The following corollary immediately follows from Lemma 1.

**Corollary 2.** *Suppose that Proposition 1 does not hold, that is, the parallel stabilization time of  $P$  is less than  $c \ln n$  with probability  $1 - o(1)$  for any constant  $c$ . Then, there exists some safe configuration of  $P$  where at least  $n^{2/3}$  agents are in state  $s_{\text{init}}$ .*

Corollary 2 implies that an execution of  $P$  must involve more than  $n^{2/3}$  agents to create a new leader if Proposition 1 does not hold. This is because otherwise an execution of  $P$  creates a new leader with only interactions involving only at most  $n^{2/3}$  agents, a contradiction to the existence of a safe configuration with at least  $n^{2/3}$  agents in state  $s_{\text{init}}$ . In what follows, we elaborate this proposition as Lemma 2 after introducing the notion of *influencer*. The set of influencers of agent  $v$  at step 0, denoted by  $F(v, 0)$ , is only  $\{v\}$ . Thereafter, the influencers of agent  $v$  is expanded every time it has an interaction with another agent. Specifically, for  $i > 0$ ,  $F(v, i) = F(v, i-1) \cup F(u, i-1)$  if  $v$  has an interaction with an agent  $u$  at step  $i$ , that is, if there exists agent  $u \in V$  such that  $\Gamma_{i-1} = (u, v)$  or  $\Gamma_{i-1} = (v, u)$ . Otherwise,  $F(v, i) = F(v, i-1)$ . See Figure 1 that depicts the set of influencers where the population consists of five agents  $\{A, B, C, D, E\}$ . In this example, by the interactions at steps 0, 1, ..., 5, the set of the influencers of agent  $A$  expands from  $\{A\}$  to  $\{A, C, D, E\}$ . We can represent  $F(v, t)$  more intuitively. Consider the directed graph

$H = (V_H, E_H)$  where  $V_H = \{(u, i) \mid u \in V, i = 0, 1, \dots, t\}$  and  $E_H$  is defined as follows:

$$E_H = \{((u, i), (u, i+1)) \mid u \in V, i = 0, 1, \dots, t-1\} \\ \cup \{((u, i), (w, i+1)) \mid u, w \in V, i = 0, 1, \dots, t-1, (u, w) \in \Gamma_i \vee (w, u) \in \Gamma_i\}.$$

(See Figure 2 for the graph  $H$  that corresponds to the example of Figure 1.) It is obvious that a node  $u$  belongs to  $F(v, t)$  if and only if node  $(v, t)$  is reachable from node  $(u, 0)$  in graph  $H$ .

**Lemma 2.** *If Proposition 1 does not hold, an execution of  $P$  never reaches a safe configuration before the number of influencers of some agent becomes greater than  $n^{2/3}$ , that is,  $C_t$  is a safe configuration only if  $|F(v, t)| > n^{2/3}$  holds for some  $v \in V$ .*

*Proof.* Assume that Proposition 1 does not hold. Then, by Corollary 2, there exists a safe configuration  $C$  of  $P$  such that  $m \geq n^{2/3}$  agents are in state  $s_{\text{init}}$ . Since  $C$  is a safe configuration, there is no sequence of interactions that leads to create another leader starting from  $C$ . This means that we cannot create a new leader by interacting only  $m$  agents with state  $s_{\text{init}}$  even if we let them interact each other infinitely many times. Therefore, we require that the number of influencers of some agent becomes greater than  $m \geq n^{2/3}$  to create a new leader. In other words, in execution  $\Xi$ , an agent  $v$  becomes a leader only at step  $t$  such that  $|F(v, t)| > n^{2/3}$ . The lemma holds because no leader exists in a configuration  $C_0 = C_{\text{init}, P}$  and thus  $\Xi$  must create a leader to reach a safe configuration.  $\square$

By Lemma 2, it suffices to show that the expansion of influencers is not so fast in order to prove Proposition 1. More specifically, our goal is now to show that  $\Omega(n \log n)$  steps are needed until some agent  $v \in V$  satisfies  $F(v, *) > n^{2/3}$ . Fortunately, the expansion of influencers is symmetric to the expansion of the epidemic [15] and can be analyzed similarly. Let  $t$  be any non negative integer. We define a sequence of sets  $I_{v,t}(0), I_{v,t}(1), \dots, I_{v,t}(t) \in 2^V$  based on the digraph  $H$  defined just above Lemma 2, as follows: for any  $i = 0, 1, \dots, t$ , a node  $u \in V$  belongs to  $I_{v,t}(i)$  if and only if  $(v, t)$  is reachable from  $(u, i)$  in  $H$ . In the example of Figure 2, we have  $I_{A,6}(6) = I_{A,6}(5) = \{A\}$ ,  $I_{A,6}(4) = I_{A,6}(3) = I_{A,6}(2) = \{A, D\}$ ,  $I_{A,6}(1) = \{A, C, D\}$ ,  $I_{A,6}(0) = \{A, C, D, E\}$ . By definition, we have the following observation.

**Observation 1.** *Let  $v \in V$  and  $t \in \mathbb{N}_{\geq 0}$ . Then, we have  $F(v, t) = I_{v,t}(0)$ .*

Let  $i \in [0, t-1]$ . Note that  $I_{v,t}(i)$  is determined only by interactions  $\Gamma_{t-1}, \Gamma_{t-2}, \dots, \Gamma_i$ . Hence,  $I_{v,t}(i)$  depends on  $I_{v,t}(i+1)$ , but  $I_{v,t}(i+1)$  is independent of  $I_{v,t}(i)$ . Suppose  $|I_{v,t}(i+1)| = k$ . Then,  $|I_{v,t}(i)| = k+1$  holds if and only if one of the  $k$  agents in  $I_{v,t}(i+1)$  and one of the  $n-k$  agents in  $V \setminus I_{v,t}(i+1)$  interact at step  $i$  (i.e., in  $\Gamma_i$ ). Therefore, we have the following observation.

**Observation 2.** *Let  $v \in V$  and  $t \in \mathbb{N}_{\geq 0}$ . Then, we have  $0 \leq |I_{v,t}(i)| - |I_{v,t}(i+1)| \leq 1$  and  $\Pr(|I_{v,t}(i)| = k+1 \mid |I_{v,t}(i+1)| = k) = \frac{2k(n-k)}{n(n-1)}$  for any integer  $k = 1, 2, \dots, n$ .*

We show that the above sufficient condition for Proposition 1 holds, as the following lemma.

**Lemma 3.** *Let  $t_{\min}$  be the smallest integer such that  $|F(v, t_{\min})| > n^{2/3}$  holds for some  $v \in V$ . Then, there exists some constant  $c$  such that  $\Pr(t_{\min} \geq cn \ln n) = 1 - o(1)$ .*

*Proof.* Let  $v$  be any agent in  $V$ . In what follows, we show  $\Pr(|F(v, \lfloor c_v n \ln n \rfloor)| > \lceil n^{2/3} \rceil) = O(n^{-2})$  for some constant  $c_v$ . This yields  $\Pr(t_{\min} < cn \ln n) = O(n^{-1}) = o(1)$  by the union bounds where  $c = \min\{c_v \mid v \in V\}$ .

By Observation 1,  $F(v, \lfloor c_v n \ln n \rfloor) = I_{v, \lfloor c_v n \ln n \rfloor}(0)$  holds. Therefore, letting  $X_k$  be a geometric random variable with parameter  $p_k = \frac{2k(n-k)}{n(n-1)}$  and  $S_{i,j} = \sum_{i \leq k \leq j} X_k$ , we obtain the following inequality by Observation 2:

$$\Pr(|F(v, \lfloor c_v n \ln n \rfloor)| > \lceil n^{2/3} \rceil) = \Pr(|I_{v, \lfloor c_v n \ln n \rfloor}(0)| > \lceil n^{2/3} \rceil) = \Pr(S_{1, \lceil n^{2/3} \rceil} \leq \lfloor c_v n \ln n \rfloor).$$

Let  $r = \lfloor \sqrt{n} \rfloor$  and  $\kappa = \lceil n^{2/3} / r \rceil$ . To make use of Chernoff bounds, we divide  $S_{1, \lceil n^{2/3} \rceil}$  to  $\kappa = O(n^{1/6})$  groups,  $S_{1,r}, S_{r+1,2r}, \dots, S_{(\kappa-1)r+1, \kappa r}$ .<sup>1</sup> Rename  $S'_i = S_{ir+1, (i+1)r-1}$  for any  $i = 0, 1, \dots, \kappa - 1$ . While  $k \leq n/2$ , probability  $p_k$  is monotonically increasing. Since  $\lceil n^{2/3} \rceil \ll n/2$  holds for sufficiently large  $n$ , we can assume  $\mathbf{E}[X_1] > \mathbf{E}[X_2] > \dots > \mathbf{E}[X_{\lceil n^{2/3} \rceil}]$ . Thus, letting  $B(l, p)$  be a binomial random variable with parameters  $l$  and  $p$ , where  $l$  is the number of trials and  $p$  is the success probability, we have

$$\begin{aligned} \Pr \left( S'_i \leq \left\lfloor \frac{r}{2} \cdot \mathbf{E}[X_{(i+1)r}] \right\rfloor \right) &< \Pr \left( B \left( \left\lfloor \frac{r}{2} \cdot \mathbf{E}[X_{(i+1)r}] \right\rfloor, p_{(i+1)r} \right) \geq r \right) \\ &\leq \exp \left( -\frac{1}{3} \cdot \left( \frac{r}{2} - 1 \right) \right) \\ &\ll n^{-3} \end{aligned}$$

for sufficiently large  $n$ , where we use the Chernoff Bound for the second inequality. Let  $E' = \sum_{0 \leq i < \kappa} \left\lfloor \frac{r}{2} \mathbf{E}[X_{(i+1)r}] \right\rfloor$ . Then, we have

$$E' = \sum_{0 \leq i < \kappa} \left\lfloor \frac{r}{2} \cdot \frac{n(n-1)}{2(i+1)r(n-(i+1)r)} \right\rfloor = \Omega \left( \sum_{0 \leq i < \kappa} \frac{n}{i} \right) = \Omega(n \log \kappa) = \Omega(n \log n).$$

Thus, for some (small) constant  $c_v$  and sufficiently large  $n$ , we have  $\lfloor c_v n \ln n \rfloor < E'$ . To conclude, we have

$$\begin{aligned} \Pr(|F(v, \lfloor c_v n \ln n \rfloor)| > \lceil n^{2/3} \rceil) &= \Pr(S_{1, \lceil n^{2/3} \rceil} \leq \lfloor c_v n \ln n \rfloor) \\ &< \Pr \left( \sum_{0 \leq i < \kappa} S'_i \leq \lfloor c_v n \ln n \rfloor \right) \\ &< \Pr \left( \sum_{0 \leq i < \kappa} S'_i \leq E' \right) \\ &< \sum_{0 \leq i < \kappa} \Pr \left( S'_i \leq \left\lfloor \frac{r}{2} \mathbf{E}[X_{(i+1)r}] \right\rfloor \right) \\ &\ll n^{-2} \end{aligned}$$

for sufficiently large  $n$ . □

**Theorem 1.** *Proposition 1 holds. That is, every leader election protocol requires  $\Omega(\log n)$  (parallel) stabilization time with probability  $1 - o(1)$ .*

*Proof.* Assume that the expected parallel stabilization time of protocol  $P$  is  $o(\log n)$ . By Lemma 2, an execution cannot reach a safe configuration before  $|F(v, t)| \geq n^{2/3}$  holds for some  $v \in V$ . However, Lemma 3 yields that this requires  $\Omega(\log n)$  parallel time, contradiction. □

The following two theorems immediately follows from Theorem 1.

**Theorem 2.** *Every leader election protocol requires  $\Omega(\log n)$  (parallel) stabilization time in expectation.*

**Theorem 3.** *No leader election protocol stabilizes within  $o(\log n)$  time with high probability (i.e., with probability  $1 - O(n^{-1})$ ).*

---

<sup>1</sup> We ignore the last segment  $S_{\kappa r+1, \lceil n^{2/3} \rceil}$  when  $\lceil n^{2/3} \rceil$  is not divisible by  $r$ . This ignorance only increase the error probability and thus does not ruin the proof, as we will see in the last sentence of the proof of this lemma.

## 4 Conclusion

In this paper, we proved that in the population protocol model, any leader election protocol requires  $\Omega(\log n)$  parallel stabilization time both in expectation and with high probability. This lower bound holds even if the protocol use an arbitrarily large number of agent states and each agent knows the exact size  $n$  of a population.

## Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers 17K19977, 18K18000, and 19H04085 and JST SICORP Grant Number JPMJSC1606.

## References

- [1] Dana. Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006.
- [2] Dan Alistarh and Rati Gelashvili. Polylogarithmic-time leader election in population protocols. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming*, pages 479–491. Springer, 2015.
- [3] Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L Rivest. Time-space trade-offs in population protocols. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2560–2579. SIAM, 2017.
- [4] Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2221–2239. SIAM, 2018.
- [5] Leszek Gąsieniec and Grzegorz Stachowiak. Fast space optimal leader election in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2653–2667. SIAM, 2018.
- [6] Leszek Gąsieniec, Grzegorz Stachowiak, and Przemysław Uznanski. Almost logarithmic-time space optimal leader election in population protocols. In *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures*, pages 93–102. ACM, 2019.
- [7] Othon Michail, Paul G Spirakis, and Michail Theofilatos. Simple and fast approximate counting and leader election in populations. In *Proceedings of the 20th International Symposium on Stabilizing, Safety, and Security of Distributed Systems*, pages 154–169. Springer, 2018.
- [8] David Doty and David Soloveichik. Stable leader election in population protocols requires linear time. *Distributed Computing*, 31(4):257–271, 2018.
- [9] Yuichi Sudo, Junya Nakamura, Yukiko Yamauchi, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Loosely-stabilizing leader election in a population protocol model. *Theoretical Computer Science*, 444:100–112, 2012.
- [10] Yuichi Sudo, Fukuhito Ooshita, Hirotsugu Kakugawa, Toshimitsu Masuzawa, Ajoy K Datta, and Lawrence L Larmore. Loosely-stabilizing leader election with polylogarithmic convergence time. In *22nd International Conference on Principles of Distributed Systems (OPODIS 2018)*, pages 30:1–30:16, 2018.



- [11] Andreas Bilke, Colin Cooper, Robert Elsässer, and Tomasz Radzik. Brief announcement: Population protocols for leader election and exact majority with  $o(\log^2 n)$  states and  $o(\log^2 n)$  convergence time. In *Proceedings of the 38th ACM Symposium on Principles of Distributed Computing*, pages 451–453. Springer, 2017.
- [12] Yuichi Sudo, Fukuhito Ooshita, Taisuke Izumi, Hirotugu Kakugawa, and Toshimitsu Masuzawa. Logarithmic expected-time leader election in population protocol model. In *Proceedings of the 21st International Symposium on Stabilization, Safety, and Security of Distributed Systems*, page (to appear), 2019.
- [13] Dan Alistarh and Rati Gelashvili. Recent algorithmic advances in population protocols. *ACM SIGACT News*, 49(3):63–73, 2018.
- [14] Robert Elsässer and Tomasz Radzik. Recent results in population protocols for exact majority and leader election. *Bulletin of EATCS*, 3(126), 2018.
- [15] Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21(3):183–199, 2008.