PAPER

Loosely-Stabilizing Algorithm on Almost Maximal Independent Set

Rongcheng DONG[†a)], ***Nonmember***, **Taisuke IZUMI**[†], ***Member***, **Naoki KITAMURA**[†], **Yuichi SUDO**[††], ***Nonmembers***, ***and*** **Toshimitsu MASUZAWA**[†], ***Member***

SUMMARY The maximal independent set (MIS) problem is one of the most fundamental problems in the field of distributed computing. This paper focuses on the MIS problem with unreliable communication between processes in the system. We propose a relaxed notion of MIS, named almost MIS (ALMIS), and show that the loosely-stabilizing algorithm proposed in our previous work can achieve exponentially long holding time with logarithmic convergence time and space complexity regarding ALMIS, which cannot be achieved at the same time regarding MIS in our previous work.
key words: *maximal independent set, distributed algorithm, self-stabilization, loose-stabilization, unreliable communications*

1. Introduction

1.1 Background

The maximal independent set (MIS) problem is one of the most fundamental and well-studied problems in distributed algorithms. Let I be a set of processes in a network, then I is an MIS if it satisfies (1) processes in I are not adjacent to each other (i.e., I is an independent set), and (2) I is not a proper subset of any other independent set. Self-stabilization [1] is a promising paradigm for designing distributed systems that can autonomously adapt to dynamics caused by transient faults and topology changes of networks. A self-stabilizing system is characterized by two properties called *convergence* and *closure*. The convergence allows the system to eventually reach legitimate configurations (i.e., configurations satisfying the problem specification) regardless of the initial configuration, and the closure makes the system stay in legitimate configurations *forever*. The MIS problem is also one of the central topics in self-stabilizing distributed graph algorithms, and a vast number of algorithms have been presented so far (see the related work section for their details).

This paper focuses on self-stabilizing MIS algorithms in the system with unreliable communication between processes in the graph, i.e., each communication channel suffers the corruption of the transmitted messages (stochastically or adversarially). An inherent limitation of conventional self-stabilizing algorithms is that it requires the system to be fault-free during its convergence. Thus, the design of self-stabilizing algorithms under the threat of perpetual faults is often recognized as a challenging problem. The adversarial message corruption is one of the popular and strong models of perpetual faults, where at each time step the adversary chooses a set of links (whose size is typically constrained) and modifies the messages transferred through the chosen edges maliciously. Even if the number of corrupted edges is bounded, the adversarial message corruption model can preclude self-stabilizing solutions for most non-trivial problems. It can be easily proved by the standard partition-based argument: consider the MIS problem in a graph where two processes, A and B, are neighboring to each other. In a legitimate configuration where A is independent (i.e., a member of an MIS I) and B is dominated (i.e., a non-member of I) only by A, when A sends a message to inform B that A is an independent process, such message may be corrupted in the link by the adversary and B cannot get the correct information. Hence B decides to change its state to become an independent process, which leads to an illegitimate configuration. Thus, the closure property of self-stabilization is violated. This observation also yields an interest in exploring a reasonably relaxed model of message corruption. Probabilistic error models, where message corruption is modeled as a stochastic event that the adversary cannot control, are widely accepted as reasonable assumptions. It is not only standard in information theory but also popular in distributed computing. Self-stabilizing solutions are, however, still ruled out even in most of the probabilistic error-models because it still admits the partition-based argument. More precisely, it allows an execution starting from any legitimate configuration to some non-legitimate one with a non-zero probability.

To circumvent the impossibility of self-stabilization in probabilistic error models, this paper focuses on *loose-stabilization* [2], which is a relaxed variant of conventional self-stabilization. While keeping the same convergence property with self-stabilization, loose-stabilization relaxes the closure property: the system is allowed to deviate from legitimate configurations after being legitimate for a long time in expectation with high probability. Loose-stabilization is practically equivalent to self-stabilization if the duration when the system stays in legitimate configurations (called *holding time*) is much longer (e.g., exponen-

Manuscript received April 17, 2023.
Manuscript revised June 8, 2023.
Manuscript publicized August 7, 2023.
[†]The authors are with Osaka University, Suita-shi, 565–0871 Japan.
[††]The author is Hosei University, Koganei-shi, 184–8584 Japan.
a) E-mail: r-dong@ist.osaka-u.ac.jp
DOI: 10.1587/transinf.2023EDP7075

Table 1 Performance of three loosely-stabilizing MIS algorithms proposed in [3].

	Redundant-State	Step-Up	Repetition
Process ID	Unavailable	Unavailable	Available
Maximum Degree	Constant	Arbitrary	Arbitrary
Error Distribution	Uniform	Uniform	Arbitrary
maxECT	$O(\log n)$	$O(n \log^3 n)$	$O(n^2 \log n)$
minEHT	$\Omega(n^d)$	$\Omega(n^d)$	$e^{\Omega(n)}$
Space	$O(\log n)$	$O(\log n)$	$O(n\Delta \log n)$

tially longer) than the time required to reach a legitimate configuration (called *convergence time*). Therefore, even if some unexpected error occurs and causes the system to become illegitimate, in a relatively short time the system can converge to legitimate configurations again and keeps being legitimate for a sufficiently long time with high probability. In our previous work [3], we proposed three loosely-stabilizing MIS algorithms considering unreliable communications: the redundant-state, the step-up, and the repetition algorithms. Three algorithms work under different settings and have different performances listed in Table 1, where d is a sufficiently large constant, Δ is the maximum degree in the graph, *maxECT* is the maximum expected convergence time, and *minEHT* is the minimum expected holding time. One of the major differences among these three algorithms lies at the trade-off between space and expected holding time. While a reasonable goal in this context is exponentially-long holding time with logarithmic space complexity, none of them attains such a performance guarantee.

1.2 Our Contribution

The primary contribution of this paper is to present a relaxed notion of the MIS problem, named almost MIS (ALMIS), and presents a loosely-stabilizing ALMIS algorithm overcoming the trade-off between space and holding time. More precisely, our motivation for relaxing the problem specification is derived from the following observation: In many of loose-stabilizing algorithms, including the three algorithms mentioned above, an execution period of holding legitimate configurations terminates with a *small* violation of the legitimacy of configurations. In the case of MIS, for example, it is typically terminated by the event that a small number of processes violate the specification of MIS. Conversely, even if the system drops out from legitimate configurations (after a holding period), almost all processes still locally satisfy the specification of MIS. Then one can also expect that the system quickly recovers a legitimate configuration again. Hence the system is expected to keep a large portion of processes "locally" legitimate, for a period much longer than the expected holding time with respect to the specification of MIS. Our notion of ALMIS aims to capture such a behavior.

The concrete results of this paper are twofold: first, we newly formulate the ALMIS problem fitting our objective. Roughly, our formulation divides the specification of MIS into two properties referred to as *independence* and *maximality*, and quantify the magnitude of violation for each property. The precise definition of ALMIS follows a parametric specification with an acceptable level of violation. The second result is to show that our formulation certainly captures what we want, by demonstrating the loosely-stabilizing redundant-state algorithm presented in [3] can achieve the exponential minimum expected holding time regarding ALMIS with a reasonably small acceptable level of violation, while it still keeps $O(\log n)$ maximum expected convergence time. In addition, one can also show that the algorithm keeps the precise MIS within most of the holding periods. The performances of the algorithm are summarized in Table 2, where k_{AL} is a parameter, n is the number of processes, γ is a sufficiently large constant, *maxECT* is maximum expected convergence time, and *minEHT* is minimum expected holding time. The parameter k_{AL} (which will be defined in the latter chapters) represents the extent that the requirements of MIS are relaxed, and by adjusting the value of k_{AL}, we can make a trade-off between the quality of ALMIS and the performance of the algorithm. When $k_{AL} = 0$, the ALMIS problem degenerates to the MIS problem.

Table 2 Performances of the Redundant-State algorithm on ALMIS and MIS.

	ALMIS			MIS
k_{AL}	$1/\gamma$	$1/o(n)$	$1/\Theta(n)$	0
maxECT	$O(\log n)$			
minEHT	$\Omega(e^{\Theta(n)})$	$\Omega\left(e^{o(n)}\right)$	$\Omega(poly(n))$	$\Omega(poly(n))$
Space	$O(\log n)$			

1.3 Related Work

One of the most classical distributed MIS algorithms is proposed by Luby [4]. This algorithm uses randomization and solves the MIS problem in $O(\log n)$ rounds based on the Monte Carlo method, where n is the number of processes. It was first improved by Barenboim et al. [5], who proposed an MIS algorithm running in $O(\log^2 \Delta + 2^{O(\sqrt{\log \log n})})$ rounds where Δ is the maximum degree, while the message size is up to $poly(\Delta \log n)$ bits. Rozhon et al. [6] improved the round complexity to $O(\log \Delta) + poly(\log \log n)$, which is currently the fastest randomized MIS algorithm. In their paper, they also proposed a deterministic MIS algorithm that runs in $poly(\log n)$ rounds while the message size is only $O(\log n)$.

Hedetniemi et al. [7] proposed a simple self-stabilizing MIS algorithm with constant round complexity while assuming a centralized scheduler. Arapoglu et al. [8] proposed a self-stabilizing MIS algorithm under the fully distributed scheduler running in $\max\{3n-6, 2n-1\}$ moves (the total number of process actions). Turau [9] considered randomization and proposed a randomized self-stabilizing MIS algorithm in the synchronous model that converges in $O(\log n)$ rounds. Afek et al. [10] proposed a self-stabilizing alternating bit protocol while considering message loss. Dolev et al. [11] proposed a self-stabilizing data-link protocol that emulates a reliable FIFO communication channel over unreliable non-FIFO channels, where messages could be lost, duplicated, created or reordered.

The notion of (α, β)-loose-stabilization, where α and β are convergence and holding time respectively, was first proposed [2] to circumvent the impossibility that the self-stabilizing leader election problem cannot be solved in the population protocol model unless the exact number of agents is available to agents in advance [12]. In [2], authors proposed a $(O(nN \log n), \Omega(Ne^N))$-loosely-stabilizing leader election algorithm in a complete graph using the population protocol model, where n is the number of processes and N is the upper bound of n. In [13], Sudo et al. generalized the topology to arbitrary graphs while requiring identifiers or random numbers. Sooner, they showed that such requirements could be eliminated and proposed another algorithm in [14]. Izumi solved the same problem and optimized the convergence time to linear while keeping the holding time to be exponential in [15]. In [16], Sudo et al. further improved the convergence time to be polylogarithmic, while the holding time is no longer exponential but polynomial with an arbitrarily large degree. In [17], Sudo et al. proposed a time-optimal loosely-stabilizing leader election protocol with logarithm convergence and polynomial holding times in expectation. Feldmann et al. [18] first applied loose-stabilization to the message passing model on server-client networks. They proposed a $(O(polylog(c(p_{min}^{-1}+n^3))), \Omega(n^c))$-loosely-stabilizing congestion control algorithm, where c is a parameter that can be chosen depending on the context, n is the number of clients and p_{min} is the minimum probability that a client will send a message to the server.

The weak variant of MIS has been proposed in several works. In [19], authors proposed the maximal nearly independent set, which relaxed the first requirement of MIS such that the number of chosen processes that are adjacent to each other is not too large, and based on this they proposed parallel greedy approximation algorithms for set cover problems. On the other hand, the second requirement of MIS, i.e., the maximality requirement, is weakened in [5] and [20], such that there exists a restrictive number of processes that are not independent nor adjacent to any independent process in the network. Such an almost-maximal independent set is used as an intermediate result to compute a strict MIS. Differently from previous works, in this paper, we formulate the definition of ALMIS that relaxes both the requirements, and analyze the performance of our previous loosely-stabilizing MIS algorithm [3] in the ALMIS problem.

2. Preliminaries

A distributed system comprises a set of autonomous processes and the communication links that connect the processes. We abstract such a system as an undirected graph $G = (V, E)$: the vertex set $V = \{p_0, p_1, \ldots, p_{n-1}\}$ represents the set of n processes where $n \geq 1$, and the edge set $E \subseteq \{\{p_i, p_j\} \mid p_i, p_j \in V, p_i \neq p_j\}$ represents the set of communication links. If $\{p_i, p_j\} \in E$, we say p_i and p_j are the neighbors of each other and can communicate with each other.

The set of neighbors of a process p_i is denoted by $N(p_i) = \{p_j \mid \{p_i, p_j\} \in E\}$, and p_i can distinguish each of its neighbors by some local labeling mechanism. Denote the degree of process p_i by $\Delta_{p_i} = |N(p_i)|$ and the maximum degree among all processes by $\Delta = \max_i \Delta_{p_i}$. Throughout this paper, we assume $\Delta = \Theta(1)$.

Denote $N^*(I)$ as the set of the processes that are not in the set $I \subseteq V$ and adjacent to the processes in I, i.e., $N^*(I) = \{p \in V \setminus I \mid N(p) \cap I \neq \emptyset\}$. For any $I \subseteq V$, we denote the *edge boundary* of I by $\partial(I) = \{\{p, q\} \in E \mid p \in I, q \notin I\}$. By definition, we have $|N^*(I)| \leq |V - I|$ and $|\partial(I)| \leq \sum_{p \in I} \Delta_p$ for any set I.

The computational model used in this paper is the atomic-state model: an algorithm is the state-machine deployed to each process. We assume all the processes are the identical state machine. Let S be the domain of the process states. A configuration of G is a tuple $C = (s_0, s_1, \ldots, s_{n-1})$ where s_i is the state of process p_i. For a given configuration C and process p, denote by $C(p)$ the state of p in C. Each process can read its state and all of its neighbors' states but can update only its state.

With respect to unreliable communication, we assume that each time a process p_i tries to read the state of each process $p_j \in N(p_i)$, p_i obtains an incorrect state with a probability ρ $(0 < \rho < 1 - \epsilon)$ for some positive constant ϵ. Furthermore, we also assume the *uniformly distributed error* model: when p_i reads an incorrect state of its neighbor $p_j \in N(p_i)$, p_i gets the state s as p_j's current state with probability $1/(|S| - 1)$ for each $s \in S \setminus \{s_j\}$, where s_j is the true state of p_j.

Synchrony or asynchrony of a system is characterized by the *scheduler*, which decides the set of activated processes at each time step. We assume the uniformly distributed scheduler U in this paper: at each *time step* (*step* for short hereafter), each process is independently activated with a constant probability ϕ. Note that the analysis and result presented in [3] and in this paper hold even if $\phi = 1$ (so-called the synchronous scheduler), since ϕ is only a constant factor. The activated process reads the states of all its neighbors and its own, then does some local computations, and updates its state if necessary. As we only consider the uniformly distributed scheduler and the uniformly distributed error model, we omit the notations of the scheduler and the error model in the followings for simplicity. In addition, since any time measurement depends on ϕ, and the asymptotic bound is the same as far as ϕ is a constant, we do not explicitly describe such a dependency.

Given an initial configuration C_0, the scheduler U, and the error model M, we define $E_{\mathcal{A}}(C_0)$ as the set of all possible executions of algorithm $\mathcal{A}$, where each execution is an infinite sequence of configurations $C_0, C_1, \ldots$ where C_{i+1} is obtained by taking a step from C_i for all $i \geq 0$ under the error model M. Each execution has the probability of its occurrence, which is determined by U, M, and $\mathcal{A}$. A specification SP is a predicate over a configuration, which is the formal definition of the problem to be solved.

In this paper, we say that an event occurs with high probability in the system if it occurs with probability more than $1 - 1/n^a$ for some constant $a > 0$ over all possible executions.

2.1 (f, g)-ALMIS

A set of processes $I \subseteq V$ is called an *independent set* if $\{p, q\} \notin E$ holds for any $p, q \in I$. In addition, if I is not a proper subset of any other independent set, then I is a *maximal* independent set (MIS). This paper introduces a relaxed notion of MIS, which we refer to as almost MIS (ALMIS). For an MIS I in graph G, we see the independence of I as the property that the cardinality of the edge boundary of I is equal to the summation of the degrees of all processes in I. That is, I is an independent set if and only if it satisfies

$$|\partial(I)| = \sum_{p \in I} \Delta_p.$$

We also see the maximality of I as the property that the number of the processes adjacent to a process in I is equal to the number of the processes not in I. That is, the independent set I is maximal if and only if it satisfies the following equality:

$$|N^*(I)| = |V - I|.$$

Now, we introduce the notion of (f, g)-*ALMIS*, a weaker variant of the standard MIS, by relaxing the two conditions above.

Definition 1. Given a distributed system G and non-decreasing functions f and g, the set $I \subseteq V$ is called an (f, g)-ALMIS of G if it satisfies the following two requirements.

- f-almost independence:

$$|\{\{p, q\} \in E \mid p, q \in I\}| \leq f\left(\sum_{p \in I} \Delta_p\right) \tag{1}$$

- g-almost maximality:

$$|V - I - N^*(I)| \leq g\left(|V - I|\right) \tag{2}$$

Intuitively, f-almost independence relaxes the requirement of independence regarding MIS. It allows the cardinality of the edge boundary of I less than the summation of the degree of processes in I with a restricted extent, which is two times the function f of the summation of the degree of processes in I:

$$\sum_{p \in I} \Delta_p - 2f\left(\sum_{p \in I} \Delta_p\right) \leq |\partial(I)|.$$

The above inequality implies the existence of edges connecting processes in I, and the number of such edges is not larger than the function f of the summation of the degree of processes in I:

$$|\{\{p, q\} \in E \mid p, q \in I\}| \leq f\left(\sum_{p \in I} \Delta_p\right).$$

Similarly, g-almost maximality relaxes the requirement of maximality regarding MIS. It allows the number of processes adjacent to a process in I less than the number of processes not in I with a restricted extent, which is the function g of the number of processes not in I:

$$|V - I| - g\left(|V - I|\right) \leq |N^*(I)|.$$

The above inequality implies the number of processes that are not in I and not adjacent to I is not larger than the function g of the number of processes that are not in I:

$$|V - I - N^*(I)| \leq g\left(|V - I|\right).$$

In this paper, we consider the case where f and g are linear functions:

$$f(x) = k_1 x \qquad \text{and} \qquad g(x) = k_2 x,$$

where k_1 and k_2 are parameters, and $0 \leq k_1, k_2 \leq 1/\gamma$ where γ is a sufficiently large constant. When $k_1 = k_2 = 0$, ALMIS degenerates to MIS, which is the case we have analyzed in detail in our previous work [3]. Therefore, we mainly consider the case of $k_1 + k_2 > 0$ in this paper, unless stated otherwise.

2.2 Loosely-Stabilizing Algorithm

Given an algorithm $\mathcal{A}$, we define $\mathcal{C}$ as the set of all possible configurations of a distributed system G. For any $\mathcal{S} \subseteq \mathcal{C}$ and $C \in \mathcal{C}$, define $ECT_{\mathcal{A}}(C, \mathcal{S})$ as the expected number of steps until executions in $E_{\mathcal{A}}(C)$ reach a configuration in $\mathcal{S}$ under the scheduler U and the error model M. For any configuration $C \in \mathcal{C}$ and a problem $\mathcal{P}$, we define $EHT_{\mathcal{A}}(C, SP_{\mathcal{P}})$ as the expected number of steps until the executions in $E_{\mathcal{A}}(C)$ deviate from the specification $SP_{\mathcal{P}}$ of $\mathcal{P}$ for the first time under the scheduler U and the error model M.

Definition 2. An algorithm $\mathcal{A}$ under the scheduler U and the error model M is an (α, β)-loosely-stabilizing algorithm for problem $\mathcal{P}$ with specification $SP_{\mathcal{P}}$ if there exists a set $\mathcal{S}$ of configurations satisfying:

- $\max_{C \in \mathcal{C}} ECT_{\mathcal{A}}(C, \mathcal{S}) \leq \alpha$
- $\min_{C \in \mathcal{S}} EHT_{\mathcal{A}}(C, SP_{\mathcal{P}}) \geq \beta$

We call α and β the maximum expected convergence time and the minimum expected holding time of $\mathcal{A}$, respectively. A configuration in $\mathcal{S}$ is called a safe configuration. Intuitively, loose-stabilization requires that an execution starting from any configuration reaches a safe configuration in a short time (α in Definition 2), and after that, the execution satisfies the problem specification for a sufficiently long time (β in Definition 2).

3. Redundant-State Algorithm

The redundant-state ($\mathcal{RS}$) algorithm proposed in our previous work [3] is a loosely-stabilizing solution for the MIS problem regarding unreliable communications, which cannot be solved in a self-stabilizing manner. In this paper, we adopt algorithm $\mathcal{RS}$ for the ALMIS problem with unreliable communications.

The key idea of algorithm $\mathcal{RS}$ is to enlarge the domain of the process state by introducing a large amount of redundancy (more specifically, let $S = \{0, 1, \ldots, c\}$ where $c = \Theta(n^{d+1})$ and d is a sufficiently large constant) so that even if some obtained state of a neighbor is corrupted by a communication error, it becomes meaningless (i.e., the bit sequence is invalid in the correct behavior of the algorithm) with high probability. In other words, the erroneous state can be detected and does not cause any incorrect effect on the receiver with high probability. This mechanism allows processes to greatly confine the influences from erroneous communications.

Algorithm $\mathcal{RS}$ [3] is presented in Algorithm 1. Each process p_i has state s_i and sets it to value 0 or c so that the set $\{p_i \in V \mid s_i = c\}$ forms the MIS or ALMIS after the execution of the algorithm $\mathcal{RS}$. State s_i takes a value from $S = \{0, 1, \ldots, c\}$ and the values other than 0 and c are the redundant values as explained above. In Algorithm 1, s'_j denotes the (possibly corrupted) state that p_i obtains from its neighbor p_j. The algorithm works as follows. Each time a process p_i is activated, it reads the states of its neighbors. Next, if its state s_i is maximal around its neighbors and s_i is not c, it is set to c; if there exists a neighbor of p_i such that s_i is smaller than that of the neighbor, then s_i is set to 0; if s_i is c and there exists a neighbor of p_i such that the state of the neighbor is also c, then s_i is set to 0.

Algorithm 1: $\mathcal{RS}$ [3] (Behavior of process p_i)

Variables in p_i:
$s_i \in \{0, 1, \ldots, c\}$ where $c = \Theta(n^{d+1})$ and d is a sufficiently large constant

if $\forall p_j \in N(p_i) : s'_j \leq s_i \wedge s_i \neq c$ **then**
  $s_i \leftarrow c$
if $\exists p_j \in N(p_i) : s'_j > s_i$ **then**
  $s_i \leftarrow 0$
if $\exists p_j \in N(p_i) : s'_j = c \wedge s_i = c$ **then**
  $s_i \leftarrow 0$

Definition 3. We classify processes into four types.

- independent process: a process p_i is an independent process if $s_i = c$ and $\forall p_j \in N(p_i) : s_j < c$.
- dominated process: a process p_i is a dominated process if $s_i = 0$ and $\exists p_j \in N(p_i) : p_j$ is an independent process.
- pseudo-dominated process: a process p_i is a pseudo-dominated process if $s_i = 0$, $\exists p_j \in N(p_i) : s_j = c$, but p_i is not dominated (i.e., there is no independent neighbor).
- illegal process: a process p_i is illegal if p_i is not independent, dominated, or pseudo-dominated. That is, (i) $0 < s_i < c$, (ii) $s_i = 0 \wedge \forall p_j \in N(p_i) : s_j < c$, or (iii) $s_i = c \wedge \exists p_j \in N(p_i) : s_j = c$.

3.1 Specification and Safe Configurations

In this section, we define the specification SP_{AL} of the ALMIS problem and two sets of safe configurations $\mathcal{S}_{AL}$ and $\mathcal{S}_{MIS}$. For convenience, we denote the set of processes with state c in a configuration C by $I(C) = \{p_i \in V \mid C(s_i) = c\}$. We omit C and write just I when no confusion occurs.

Definition 4. Given a distributed system G and non-decreasing functions f and g, the specification $SP_{AL}(f, g)$ of ALMIS problem is defined as

$$SP_{AL}(f, g) \overset{\text{def}}{=} (I \text{ is an } (f, g)\text{-ALMIS of } G) \wedge \forall p_i \notin I : s_i = 0$$

Definition 5. Given a distributed system G and non-decreasing functions f and g, define $\mathcal{S}_{AL}$ as the set of configurations where the specification $SP_{AL}(f, g)$ is satisfied.

Define $\mathcal{S}_{MIS}$ as the set of configurations where for any configuration in $\mathcal{S}_{MIS}$, all processes in I are independent and all processes not in I are dominated (or $SP_{AL}(0, 0)$ is satisfied).

From the definition, we have $\mathcal{S}_{MIS} \subseteq \mathcal{S}_{AL}$. The reason for introducing two kinds of safe configurations is that, it is intuitive to introduce $\mathcal{S}_{AL}$ since we are considering the ALMIS problem. However, we will show that $\mathcal{S}_{AL}$ is not a good choice since some configurations in $\mathcal{S}_{AL}$ are very vulnerable regarding the almost maximality requirement, which implies only a constant minimum expected holding time. A smaller set of safe configurations is expected to enable a longer holding time while possibly sacrificing the convergence time. Therefore, we use $\mathcal{S}_{MIS}$ as the set of safe configurations and succeed to achieve the loose-stabilization with exponential holding time and logarithmic convergence time.

4. Analysis of Time Complexity

In this section, we analyze the time complexity of algorithm $\mathcal{RS}$ regarding the ALMIS problem. In Sect. 4.1, we give an overview of the analysis, which shows the analysis for the convergence time is trivial from our previous work, and we cannot achieve loose-stabilization when considering $\mathcal{S}_{AL}$

as safe configurations. In Sect. 4.2, we show algorithm $\mathcal{RS}$ is a loosely-stabilizing algorithm regarding ALMIS with an exponentially long minimum expected holding time when considering $\mathcal{S}_{MIS}$ as safe configurations.

4.1 Convergence Time and Impossibility Result

We can get the convergence time trivially by our previous work [3]. When considering $\mathcal{S}_{MIS}$ as safe configurations, we have the maximum expected convergence time for the ALMIS problem directly by the following lemma.

Lemma 1 (Theorem 1 in [3]). $\max_{C\in\mathcal{C}} ECT_{\mathcal{RS}}(C,\mathcal{S}_{MIS}) = O(\log n)$ *in terms of steps, where $\mathcal{C}$ is the set of all possible configurations of $\mathcal{RS}$.*

Now we consider the case where $\mathcal{S}_{AL}$ is the set of safe configurations. Since $\mathcal{S}_{MIS} \subseteq \mathcal{S}_{AL}$ by Definition 5, ALMIS can be regarded as an intermediate state of MIS. Thus, the maximum expected convergence time regarding ALMIS is upper-bounded by that regarding MIS, which is $O(\log n)$ steps by Lemma 1.

Now we show that the above upper bound $O(\log n)$ is tight. To lower-bound the maximum expected convergence time regarding ALMIS, consider the case where initially all processes have states neither 0 nor c. In this case, an ALMIS can be achieved only after all n processes make moves, and we can prove that such a case happens with high probability in $\Theta(\log n)$ steps by the following lemma.

Lemma 2 (Lemma 1 in [3]). *By taking $\Theta(\log n)$ steps from an arbitrary initial configuration, all processes have state 0 or c with probability $1 - e^{-\Theta(\log n)}$.*

Therefore, the maximum expected convergence time regarding ALMIS is lower-bounded by $\Theta(\log n)$. Together with Lemma 1, we have the following lemma.

Lemma 3. $\max_{C\in\mathcal{C}} ECT_{\mathcal{RS}}(C,\mathcal{S}_{AL}) = \Theta(\log n)$ *in terms of steps, where $\mathcal{C}$ is the set of all possible configurations of $\mathcal{RS}$.*

In the following, we analyze the minimum expected holding time of the case where $\mathcal{S}_{AL}$ is the set of safe configurations, and show loose-stabilization cannot be achieved in this case, as we mentioned in Sect. 3.1. First, we analyze the bounds of $|I|$ and $|V-I|$ when I is an ALMIS in the following lemma.

Lemma 4. *If I is an ALMIS, then* $\frac{1}{2(1+\Delta)} \cdot n \le |I| \le \frac{2\Delta}{1+2\Delta} \cdot n$.

Proof. First, we have

$$\begin{aligned} n &= |I| + |N^*(I)| + |V - I - N^*(I)| \\ &\le |I| + \sum_{p\in I} \Delta_p + k_2|V-I| \\ &\le (1+\Delta)|I| + k_2 \cdot (n - |I|) \\ &= (1+\Delta-k_2)|I| + k_2 n. \end{aligned}$$

We have the second line of inequality by the facts that $|N^*(I)| \le |\partial(I)|$ and $|\partial(I)| \le \sum_{p\in I}\Delta_p$. The above inequalities yields

$$|I| \ge \frac{1-k_2}{1+\Delta-k_2} \cdot n \ge \frac{1}{2(1+\Delta)} \cdot n.$$

On the other hand, we have

$$\begin{aligned} |I| &\le \sum_{p\in I} \Delta_p \\ &= 2\,|\{\{p,q\} \in E \mid p,q \in I\}| + |\partial(I)| \\ &\le 2\,|\{\{p,q\} \in E \mid p,q \in I\}| + \Delta|N^*(I)| \\ &\le 2k_1 \sum_{p\in I} \Delta_p + \Delta|N^*(I)| \\ &\le 2k_1\Delta|I| + \Delta|N^*(I)|, \end{aligned}$$

which yields

$$|N^*(I)| \ge \frac{1-2k_1\Delta}{\Delta}|I| \ge \frac{1}{2\Delta}|I|.$$

Combine the above result and the fact that $|I| + |N^*(I)| \le n$, we have

$$|I| \le \frac{2\Delta}{1+2\Delta} \cdot n.$$

□

By Lemma 4, we can directly get

$$\frac{1}{2\Delta} \cdot n \le |V-I| \le \frac{1+2\Delta}{2+2\Delta} \cdot n.$$

After we bound $|I|$ and $|V-I|$, we can also bound the number of illegal processes in the graph when I is an ALMIS in the following lemma. Denote $t^{(ii)}$ and $t^{(iii)}$ as the number of illegal processes satisfying the conditions (ii) or (iii) in the definition of illegal processes (cf. Definition 3) in the graph, respectively. Notice that we do not consider the illegal process satisfying (i) in the following because all processes have state 0 or c in $\mathcal{S}_{AL}$ by Definition 5, and after that, no processes change its state to values other than 0 or c according to algorithm $\mathcal{RS}$.

Lemma 5. *If I is an ALMIS and equalities both hold in inequalities (1) and (2), then the lower bounds of $t^{(ii)}$ and $t^{(iii)}$ are $\frac{1}{2\Delta} \cdot k_2 n$ and $\frac{1}{2\Delta(1+\Delta)} \cdot k_1 n$, respectively.*

Proof. The number $t^{(ii)}$ of illegal processes satisfying (ii) is

$$\begin{aligned} t^{(ii)} &= |V - I - N^*(I)| \\ &= k_2|V-I| \\ &\ge \frac{1}{2\Delta} \cdot k_2 n. \end{aligned}$$

The number $t^{(iii)}$ of illegal processes satisfying (iii) is

$$t^{(iii)} \ge \frac{1}{\Delta} \cdot |\{\{p,q\} \in E \mid p,q \in I\}|$$

$$= \frac{1}{\Delta} \cdot k_1 \sum_{p \in I} \Delta_p$$
$$\geq \frac{1}{\Delta} \cdot k_1 |I|$$
$$\geq \frac{1}{2\Delta(1+\Delta)} \cdot k_1 n.$$

□

In the following two lemmas, we prove loose-stabilization cannot be achieved when considering $\mathcal{S}_{AL}$ as safe configurations.

Lemma 6 (Lemma 4 in [3]). *If process p_i is an illegal process, then in any step, the probability that p_i changes its state is* $\Theta(1)$.

Lemma 7. $\min_{C \in \mathcal{S}_{AL}} EHT_{\mathcal{RS}}(C, SP_{AL}) = \Theta(1)$ *in terms of steps.*

Proof. Consider a configuration $C_i \in \mathcal{S}_{AL}$ where I is an ALMIS and equalities hold in both inequalities (1) and (2). By Lemma 5, we have

$$t_i^{\text{(ii)}} \geq \frac{1}{2\Delta} \cdot k_2 n \quad \text{and} \quad t_i^{\text{(iii)}} \geq \frac{1}{2\Delta(1+\Delta)} \cdot k_1 n.$$

Denote $\partial t_i^{\text{(ii)}}$ and $\partial t_i^{\text{(iii)}}$ as the number of illegal processes satisfying (ii) and (iii) that change state in C_i, respectively. By Lemma 6, the probability that $\partial t_i^{\text{(ii)}} + \partial t_i^{\text{(iii)}} > 0$ is (remind Δ is a constant)

$$1 - (1 - \Theta(1))^{t_i^{\text{(ii)}} + t_i^{\text{(iii)}}} \geq 1 - e^{-(k_1+k_2)\Theta(n)}.$$

Note the fact that when illegal processes satisfying (ii) change states, at most the same number of illegal processes satisfying (iii) will be created (e.g., two adjacent illegal processes satisfying (ii) change states and become illegal processes satisfying (iii) simultaneously); when a single illegal process p satisfying (iii) changes state, at most $\Delta - 1$ illegal processes satisfying (ii) will be created (e.g., $\Delta - 1$ neighbors of p has states 0 and p is the only neighbor of them that had state c). Therefore, after $\partial t_i^{\text{(ii)}}$ ($\partial t_i^{\text{(iii)}}$, respectively) illegal processes satisfying (ii) ((iii), respectively) change their states, in the worst case we have

$$t_{i+1}^{\text{(ii)}} = t_i^{\text{(ii)}} - \partial t_i^{\text{(ii)}} + (\Delta - 1)\partial t_i^{\text{(iii)}}$$

and

$$t_{i+1}^{\text{(iii)}} = t_i^{\text{(iii)}} - \partial t_i^{\text{(iii)}} + \partial t_i^{\text{(ii)}}.$$

If $\partial t_i^{\text{(iii)}} \geq \partial t_i^{\text{(ii)}}$, we have $t_{i+1}^{\text{(ii)}} > t_i^{\text{(ii)}}$; if $\partial t_i^{\text{(iii)}} < \partial t_i^{\text{(ii)}}$, we have $t_{i+1}^{\text{(iii)}} > t_i^{\text{(iii)}}$. If $t^{\text{(ii)}}$ increases by 1, the left side of inequality (2) increases by 1 and the right side increases by k_2, which leads to the violation of inequality (2) (remind that equality holds in inequality (2) before $t^{\text{(ii)}}$ increases). Similarly, if $t^{\text{(iii)}}$ increases by 1, inequality (1) is violated. Therefore, we have $C_{i+1} \notin \mathcal{S}_{AL}$ with probability at least $1 - e^{-(k_1+k_2)\Theta(n)}$, which yields only constant steps of minimum expected holding time (remind $k_1 + k_2 > 0$). □

By Lemmas 3 and 7, the minimum expected holding time is much less than the maximum expected convergence time, which yields that loose-stabilization cannot be achieved when considering $\mathcal{S}_{AL}$ as safe configurations. Intuitively, the reason that causes the impossibility is the possible existence of pseudo-dominated processes that can potentially become illegal processes, as we have shown in the proof of Lemma 7. In the next section, we focus on the analysis of the holding time when considering $\mathcal{S}_{MIS}$ as safe configurations, and loose-stabilization can be achieved in this case.

4.2 Holding Time

In this section, we show that algorithm $\mathcal{RS}$ has an exponentially long minimum expected holding time when considering $\mathcal{S}_{MIS}$ as safe configurations, which yields that $\mathcal{RS}$ can achieve loose-stabilization for the ALMIS problem. In any execution starting from a safe configuration in $\mathcal{S}_{MIS}$, no process changes its state to a value other than 0 or c. Thus, we do not need to consider the illegal process satisfying (i) in the following.

Denote PI as the total number of pseudo-dominated and illegal processes. We analyze how the action of each process affects the graph in the following two lemmas. More specifically, in the following lemmas, we prove that if independent or dominated processes change states, PI will increase; if pseudo-dominated or illegal processes change states, PI does not increase.

Lemma 8. *If an independent or dominated process change its state, PI increases by at most $\Delta^2 + 1$.*

Proof. Let p be the process that changes its state. When p is an independent process and changes its state to 0, it becomes an illegal process satisfying (ii). A neighbor q of p also becomes an illegal process satisfying (ii) when p was the only independent process adjacent to q. Therefore, PI increases by at most $\Delta + 1$.

When p is a dominated process and changes its state to c, it becomes an illegal process satisfying (iii). A neighbor q of p that was independent also becomes an illegal process satisfying (iii), and all neighbors of q except p becomes pseudo-dominated processes when q was the only independent process adjacent to them. Therefore, PI increases by at most $\Delta^2 + 1$. □

Lemma 9. *If pseudo-dominated or illegal processes change states, PI does not increase.*

Proof. By Definition 3, independent processes are not adjacent to any pseudo-dominated or illegal process (remind that we do not consider illegal processes satisfying (i) here, since no process changes its state to a value other than 0 or c). Therefore, independent processes are not affected and remain independent when pseudo-dominated and illegal processes change their states. On the other hand, dominated processes may be adjacent to pseudo-dominated and illegal processes. However, since dominated processes are adjacent to independent processes by Definition 3 and independent processes remain independent when pseudo-dominated and illegal processes change their states, dominated processes also remain dominated when pseudo-dominated and illegal processes change their states. Therefore, PI does not increase. □

By Lemma 5, we know that violation of $SP_{AL}(f, g)$ requires $t^{(\text{ii})} > \frac{1}{2\Delta} \cdot k_2 n$ or $t^{(\text{iii})} > \frac{1}{2\Delta(1+\Delta)} \cdot k_1 n$. Denote $k_{AL} = \min\{\frac{1}{2\Delta(1+\Delta)} \cdot k_1, \frac{1}{2\Delta} \cdot k_2\}$ for simplicity. In the following, we analyze the case

$$PI > k_{AL} n$$

that is weaker than the previous one so that we can upper-bound the probability that the previous case happens. To do this, we use the results of the following lemmas from [3].

Lemma 10 (Lemma 2 in [3]). *If process p_i is an independent, dominated or pseudo-dominated process, then in any step, the probability that p_i keeps the same state is $1 - O(1/c)$, where $c = \Theta(n^{d+1})$ and d is a sufficiently large constant.*

Lemma 11 (Lemma 5 in [3]). *If process p_i is an illegal process, then in any step, the probability that p_i becomes independent is $\Omega(1)$.*

Lemma 12 (Lemma 6 in [3]). *If process p_i is a pseudo-dominated process, then in any step, the probability that p_i becomes dominated is $\Omega(1)$.*

Denote P_l as the lower bound of the probabilities that an illegal process becomes independent and a pseudo-dominated process becomes dominated in any step. By Lemmas 11 and 12, P_l is a constant greater than 0.

Lemma 13. *Given an initial configuration $C_0 \in \mathcal{S}_{MIS}$ and an execution $E \in E_{\mathcal{RS}}(C_0)$. In any configuration C_i in E where $i \geq 0$, we have $PI_i \leq k_{AL} n$ with probability at least $1 - e^{-\Theta(k_{AL} \cdot n)}$, where PI_i is the value of PI in C_i.*

Proof. We prove the lemma by proving the following claims.

Claim 1. *If $PI_i = o(k_{AL} n)$, then $PI_{i+1} \leq k_{AL} n$ with probability at least $1 - O(n^{-\Theta(k_{AL} \cdot n)})$.*

Proof. By Lemma 8, PI_i increases by at most $\Delta^2 + 1$ if an independent or dominated process changes its state. Therefore, if at most $\frac{1}{2(\Delta^2+1)} \cdot k_{AL} n$ independent and dominated processes change states, we have

$$PI_{i+1} \leq PI_i + (\Delta^2 + 1) \cdot \frac{1}{2(\Delta^2 + 1)} \cdot k_{AL} n < k_{AL} n.$$

The probability such a case happens is (remind Δ is a constant)

$$1 - \binom{n - PI_i}{\frac{1}{2(\Delta^2+1)} \cdot k_{AL} n} \cdot O\left(\frac{1}{c}\right)^{\frac{1}{2(\Delta^2+1)} \cdot k_{AL} n} \geq 1 - O(n^{-\Theta(k_{AL} \cdot n)}).$$

□

Claim 2. *If $PI_i = \tau \cdot k_{AL} n$ for some constant τ s.t. $0 < \tau \leq 1$ in C_i, then $PI_{i+1} < PI_i$ with probability at least $1 - e^{-\Theta(k_{AL} \cdot n)}$.*

Proof. By Lemma 9, PI_i does not increase if pseudo-dominated or illegal processes change states. Moreover, pseudo-dominated and illegal processes may become dominated or independent with probability at least P_l (remind P_l is a constant) by Lemmas 11 and 12. By Hoeffding's inequality [21], the probability that at least $\frac{1}{2} P_l \cdot PI_i$ illegal processes independent in one step is at least

$$1 - e^{-2PI_i \cdot \left(P_l - \frac{\frac{P_l}{2} \cdot PI_i}{PI_i}\right)^2} = 1 - e^{-\Theta(k_{AL} n)}.$$

By a similar analysis with Claim 1, the probability that $\frac{P_l}{4} \cdot PI_i$ independent and dominated processes change states in one step is at least $1 - O(n^{-\Theta(k_{AL} \cdot n)})$. Therefore, with probability

$$(1 - e^{-\Theta(k_{AL} n)}) \cdot (1 - O(n^{-\Theta(k_{AL} \cdot n)})) = 1 - e^{-\Theta(k_{AL} n)},$$

we have

$$PI_{i+1} \leq PI_i - \frac{1}{2} P_l \cdot PI_i + \frac{1}{4} P_l \cdot PI_i < PI_i.$$

□

Initially, we have $PI_0 = 0$ since $C_0 \in \mathcal{S}_{MIS}$. At each configuration C_i $(i > 0)$ after C_0, if PI_{i-1} is $o(k_{AL} n)$, e.g., $PI_{i-1} = 0$, we have $PI_i \leq k_{AL} n$ with high probability by Claim 1; if $PI_{i-1} = \tau \cdot k_{AL} n$ for some constant τ s.t. $0 < \tau \leq 1$, e.g., $PI_{i-1} = \frac{1}{2} \cdot k_{AL} n$ or $PI_{i-1} = k_{AL} n$, we have $PI_i < PI_{i-1} \leq k_{AL} n$ with high probability by Claim 2. Therefore, we have the lemma. □

Combining the result of our previous work [3] and Lemma 13, we can analyze the holding time in the following lemma.

Lemma 14.

$$\min_{C \in \mathcal{S}_{MIS}} EHT_{\mathcal{RS}}(C, SP_{AL}) = \Omega\left(poly(n) + e^{\Theta(k_{AL} \cdot n)}\right)$$

in terms of steps.

Proof. To deviate from the specification of ALMIS in executions with initial configurations in $\mathcal{S}_{MIS}$, the system should deviate from $\mathcal{S}_{MIS}$ and also reach a configuration with $PI > k_{AL} n$. By the result of [3], the minimum expected number of steps to deviate from $\mathcal{S}_{MIS}$ is $\Omega(poly(n))$.

By Lemma 13, we have $PI \leq k_{AL} n$ with probability at least $1 - e^{-\Theta(k_{AL} n)}$ for any configuration in execution starting from an initial configuration in $\mathcal{S}_{MIS}$. Hence, the probability that $PI > k_{AL} n$ is at most $e^{-\Theta(k_{AL} n)}$, which yields the minimum expected $1/e^{-\Theta(k_{AL} n)} = e^{\Theta(k_{AL} n)}$ steps up to the first configuration satisfying $PI > k_{AL} n$, assuming the execution starts from a configuration in $\mathcal{S}_{MIS}$. Therefore, we can obtain the minimum expected holding time of $\Omega\left(\max\{poly(n), e^{\Theta(k_{AL} \cdot n)}\}\right) = \Omega\left(poly(n) + e^{\Theta(k_{AL} \cdot n)}\right)$. □

Combining Lemmas 1 and 14, we have our main theorem of the paper.

Theorem 1. *Algorithm* $\mathcal{RS}$ *is a* $(O(\log n), \Omega(poly(n) + e^{\Theta(k_{AL}\cdot n)}))$*-loosely-stabilizing algorithm regarding the ALMIS problem.*

By choosing different k_{AL}, we can leverage the quality of ALMIS and its minimum expected holding time.

- When $k_{AL} = 1/\gamma$ where γ is a sufficiently large constant, the minimum expected holding time is $\Omega(e^{\Theta(n)})$.
- When $k_{AL} = 1/o(n)$, e.g., $k_{AL} = 1/\log n$, the minimum expected holding time is $\Omega(e^{\Theta(n/\log n)})$, which is subexponentially long.
- When $k_{AL} = \Theta(1/n)$ or 0, the minimum expected holding time is $\Omega(poly(n))$, which is polynomial long. Note that when $k_{AL} = 0$, the problem degenerates to the MIS problem.

5. Conclusion

In this paper, we formulated the definition of almost MIS (ALMIS) and applied the loosely-stabilizing redundant-state algorithm regarding ALMIS. We showed that by considering ALMIS, the algorithm can keep the logarithmic maximum expected convergence time while achieving at most the exponential minimum expected holding time by the choice of quality of ALMIS.

References

[1] E.W. Dijkstra, "Self-stabilizing systems in spite of distributed control," Communications of the ACM, vol.17, no.11, pp.643–644, 1974.

[2] Y. Sudo, J. Nakamura, Y. Yamauchi, F. Ooshita, H. Kakugawa, and T. Masuzawa, "Loosely-stabilizing leader election in a population protocol model," Theoretical Computer Science, vol.444, pp.100–112, 2012.

[3] R. Dong, Y. Sudo, T. Izumi, and T. Masuzawa, "Loosely-stabilizing maximal independent set algorithms with unreliable communications," Theoretical Computer Science, vol.937, pp.69–84, 2022.

[4] M. Luby, "A simple parallel algorithm for the maximal independent set problem," SIAM journal on computing, vol.15, no.4, pp.1036–1053, 1986.

[5] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider, "The locality of distributed symmetry breaking," Journal of the ACM (JACM), vol.63, no.3, pp.1–45, 2016.

[6] V. Rozhoň and M. Ghaffari, "Polylogarithmic-time deterministic network decomposition and distributed derandomization," Proc. 52nd Annual ACM SIGACT Symposium on Theory of Computing, pp.350–363, 2020.

[7] S.M. Hedetniemi, S.T. Hedetniemi, D.P. Jacobs, and P.K. Srimani, "Self-stabilizing algorithms for minimal dominating sets and maximal independent sets," Computers & Mathematics with Applications, vol.46, no.5-6, pp.805–811, 2003.

[8] O. Arapoglu, V.K. Akram, and O. Dagdeviren, "An energy-efficient, self-stabilizing and distributed algorithm for maximal independent set construction in wireless sensor networks," Computer Standards & Interfaces, vol.62, pp.32–42, 2019.

[9] V. Turau, "Making randomized algorithms self-stabilizing," International Colloquium on Structural Information and Communication Complexity, pp.309–324, Springer, 2019.

[10] Y. Afek and G.M. Brown, "Self-stabilization over unreliable communication media," Distributed Computing, vol.7, no.1, pp.27–34, 1993.

[11] S. Dolev, S. Dubois, M. Potop-Butucaru, and S. Tixeuil, "Stabilizing data-link over non-fifo channels with optimal fault-resilience," Information Processing Letters, vol.111, no.18, pp.912–920, 2011.

[12] D. Angluin, J. Aspnes, M.J. Fischer, and H. Jiang, "Self-stabilizing population protocols," ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol.3, no.4, pp.1–28, 2008.

[13] Y. Sudo, F. Ooshita, H. Kakugawa, T. Masuzawa, A.K. Datta, and L.L. Larmore, "Loosely-stabilizing leader election for arbitrary graphs in population protocol model," IEEE Trans. Parallel Distrib. Syst., vol.30, no.6, pp.1359–1373, 2018.

[14] Y. Sudo, F. Ooshita, H. Kakugawa, and T. Masuzawa, "Loosely stabilizing leader election on arbitrary graphs in population protocols without identifiers or random numbers," IEICE Trans. Inf. & Syst., vol.E103.D, no.3, pp.489–499, 2020.

[15] T. Izumi, "On space and time complexity of loosely-stabilizing leader election," International Colloquium on Structural Information and Communication Complexity, pp.299–312, Springer, 2015.

[16] Y. Sudo, F. Ooshita, H. Kakugawa, T. Masuzawa, A.K. Datta, and L.L. Larmore, "Loosely-stabilizing leader election with polylogarithmic convergence time," Theoretical Computer Science, vol.806, pp.617–631, 2020.

[17] Y. Sudo, R. Eguchi, T. Izumi, and T. Masuzawa, "Time-Optimal Loosely-Stabilizing Leader Election in Population Protocols," 35th International Symposium on Distributed Computing (DISC 2021), ed. S. Gilbert, Leibniz International Proceedings in Informatics (LIPIcs), vol.209, Dagstuhl, Germany, pp.40:1–40:17, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

[18] M. Feldmann, T. Götte, and C. Scheideler, "A loosely self-stabilizing protocol for randomized congestion control with logarithmic memory," International Symposium on Stabilizing, Safety, and Security of Distributed Systems, pp.149–164, Springer, 2019.

[19] G.E. Blelloch, R. Peng, and K. Tangwongsan, "Linear-work greedy parallel approximate set cover and variants," Proc. twenty-third annual ACM symposium on Parallelism in algorithms and architectures, pp.23–32, 2011.

[20] M. Ghaffari, "Distributed maximal independent set using small messages," Proc. Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, pp.805–820, SIAM, 2019.

[21] W. Hoeffding, "Probability inequalities for sums of bounded random variables," The collected works of Wassily Hoeffding, pp.409–426, 1994.

Rongcheng Dong received the B.E. degree in computer science from Shanghai Jiao Tong University in 2018, and M.E. degree in computer science from Osaka University in 2020. Currently, he is a Ph.D. student at the Graduate School of Information Science and Technology, Osaka University. His research interest includes distributed algorithms.

Taisuke Izumi received the ME and DI degrees in computer science from Osaka University, Japan, in 2003 and 2006, respectively. He worked as an assistant professor at Nagoya Institute of Technology, Japan, during 2006-2009, and worked as an assistant professor during 2009-2020. He is currently an associate professor at the Graduate School of Information Science and Technology, Osaka University, Japan. His research interests include algorithms and distributed systems. He is a member of IEICE, IPSJ, and ACM.

Naoki Kitamura received the B.S., M.S., and Ph.D. degrees in Computer Science from Nagoya Institute of Technology in 2017, 2019, and 2022, respectively. He is now an assistant professor at the Graduate School of Information Science and Technology, Osaka University.

Yuichi Sudo received the B.E., M.E., and Ph.D. degrees in information science and technology from Osaka University in 2009, 2011, and 2015, respectively. He worked at NTT Corporation and was engaged in research on network security during 2011–2017. He was an assistant professor at the Graduate School of Information Science and Technology, Osaka University, during 2017–2021. He has been an associate professor at Faculty of Computer and Information Sciences, Hosei University, since April 2021. His research interests include distributed algorithms and graph theory. He is a member of IEEE and EATCS.

Toshimitsu Masuzawa received the B.E., M.E.and D.E. degrees in computer science from Osaka University in 1982, 1984 and 1987. He had worked at Osaka University during 1987-1994, and was an associate professor of Graduate School of Information Science, Nara Institute of Science and Technology (NAIST) during 1994–2000. He is now a professor of Graduate School of Information Science and Technology, Osaka University. He was also a visiting associate professor of Department of Computer Science, Cornell University between 1993–1994. His research interests include distributed algorithms, parallel algorithms and graph theory. He is a member of ACM, IEEE, IEICE and IPSJ.