PAPER Special Section on Foundations of Computer Science — Algorithm, Theory of Computation, and their Applications —

Space-Optimal Population Protocols for Uniform Bipartition Under Global Fairness*

Hiroto YASUMI[†a)], ***Student Member***, **Fukuhito OOSHITA**[†b)], **Ken'ichi YAMAGUCHI**[††c)], ***Members***, ***and*** **Michiko INOUE**[†d)], ***Senior Member***

SUMMARY In this paper, we consider a uniform bipartition problem in a population protocol model. The goal of the uniform bipartition problem is to divide a population into two groups of the same size. We study the problem under global fairness with various assumptions: 1) a population with or without a base station, 2) symmetric or asymmetric protocols, and 3) designated or arbitrary initial states. As a result, we completely clarify solvability of the uniform bipartition problem under global fairness and, if solvable, show the tight upper and lower bounds on the number of states.
key words: *population protocol, uniform bipartition, distributed protocol*

1. Introduction

1.1 The Background

A population protocol model [2] is an abstract model that represents computation on a network of low-performance devices. We refer to such devices as agents and a set of agents as a population. Agents can update their states by interacting with other agents, and proceed with computation by repeating the pairwise interactions. The population protocol model can be applied to many systems. For example, one may construct sensor networks to monitor wild birds by attaching sensors to them. In this system, sensors collect and process data based on pairwise interactions when two sensors (or birds) come sufficiently close to each other. Another future example is a system of low-performance molecular robots [3]. The system is being developed, for example, to deploy inside a human body and diagnose the physical condition. To realize such systems, many protocols have been proposed as building blocks in the population protocol model [4]. For example, they include leader election protocols [5]–[12], counting protocols [13]–[16], and majority protocols [5], [17]–[19].

In this paper, we consider a uniform bipartition problem, which divides a population into two groups of the same size. The uniform bipartition problem is a special case of a group composition problem, which divides a population into multiple groups to satisfy some conditions. Some protocols for the group composition problem are developed as subroutines to realize fault-tolerant protocols [20] and periodic functions [21]. However, the complexity of the problem has not been studied deeply yet. For this reason, as the first step to study the complexity of the group composition problem, we focus on the space complexity of the uniform bipartition problem. Note that the uniform bipartition problem itself has some applications. For example, we can reduce energy consumption by switching on one group and switching off the other. In another example, we can assign a different task to each group and make agents execute multiple tasks at the same time. This can be regarded as differentiation of a population in the sense that initially identical agents are eventually divided into two groups and execute different tasks. In addition, by repeating uniform bipartition, we can divide a population into an arbitrary number of groups with almost the same size. For example, by repeating uniform bipartition four times, we can make sixteen groups of the same size. We can regroup the sixteen groups to three groups with almost the same size by partitioning them into five, five, and six groups.

1.2 Our Contributions

For the uniform bipartition problem, we clarify solvability and minimum requirements of agent space under various assumptions. More concretely, we consider three types of assumptions, 1) a population with or without a base station, 2) symmetric or asymmetric protocols, and 3) designated or arbitrary initial states. A base station (BS) is a distinguishable agent with a powerful capability. When a single BS exists in a population, the BS can behave as a leader; for example, it can collect information of agents and assign roles to agents. This facilitates design of population protocols, however in some applications we cannot use a BS. Symmetric property of protocols is related to the power of symmetry breaking in the population. Asymmetric protocols may include transitions that make agents with the same states enter different states. This requires a mechanism to break symmetry among agents and its implementation is sometimes difficult with low-performance agents such as molecular robots.

Manuscript received March 23, 2018.
Manuscript revised July 27, 2018.
Manuscript publicized October 30, 2018.
[†]The authors are with Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.
[††]The author is with College of Information Science, National Institute of Technology, Nara College, Koriyama-shi, 639–1080 Japan.
[*]The conference version of this paper is published in the 21st International Conference on Principles of Distributed Systems [1].
a) E-mail: yasumi.hiroto.yf9@is.naist.jp
b) E-mail: f-oosita@is.naist.jp
c) E-mail: yamaguti@info.nara-k.ac.jp
d) E-mail: kounoe@is.naist.jp
DOI: 10.1587/transinf.2018FCP0009

Table 1 Our contributions: The number of states to solve the uniform bipartition problem.

BS	initial states	symmetry	upper bound	lower bound
single	designated	asymmetric	3	3
		symmetric	3	3
	arbitrary	asymmetric	4	4
		symmetric	4	4
no	designated	asymmetric	3 [20]	3
		symmetric	4 [23]	4
	arbitrary	asymmetric	unsolvable	
		symmetric	unsolvable	

Symmetric protocols do not include such transitions. The assumption of initial states is related to the requirement of initialization and the fault-tolerant property. If a protocol requires a designated initial state, we need some mechanism to initialize agent states before executing protocols. On the other hand, when the protocol allows arbitrary initial states, initialization of agents other than the BS is not necessary. In addition, even if agents enter arbitrary states due to transient faults, the system can eventually reach the desired configuration by initializing the BS. If a protocol allows arbitrary initial states and does not require a BS, the protocol is self-stabilizing because it can work from arbitrary initial configurations.

In addition to the above assumptions, we require some fairness assumption on interactions of agents. This is because, if some agents do not join any interaction, no protocol can solve the uniform bipartition problem. In this paper, we adopt global fairness, which is a common assumption to ensure the progress of protocols [4], [7]–[9], [13], [14], [16], [20], [22], [23]. We give the definition of global fairness in Sect. 2.

For each combination of assumptions, we completely clarify solvability of the uniform bipartition problem and, if solvable, show the tight upper and lower bounds on the number of states. Our contributions are given in Table 1.

First, we consider the case of a single BS. For designated initial states, we give a symmetric protocol with three states and prove impossibility of asymmetric protocols with two states. That is, three states are necessary and sufficient for designated initial states. For arbitrary initial states, we give a symmetric protocol with four states and prove impossibility of asymmetric protocols with three states. That is, four states are necessary and sufficient for arbitrary initial states. These results show that only one additional state is required to treat arbitrary initial states.

Next, we consider the case of no BS. For designated initial states, no asymmetric protocol with two states exists (this is clearly derived from the result of a single BS). Since an asymmetric protocol with three states is given in [20], three states are necessary and sufficient for asymmetric protocols. For symmetric protocols, we prove impossibility with three states. Since a symmetric protocol with four states is obtained by a general transformer in [23], four states are necessary and sufficient for symmetric protocols. For arbitrary initial states, we prove that no protocol exists even if the protocol can use any number of states. This implies that a BS is necessary for protocols with arbitrary initial states.

1.3 Related Works

The population protocol model was introduced by Angluin et al. [2], [22]. They regard initial states of agents as an input to the system, and resultant states of them as an output from the system. Following this definition, they clarified the class of computable predicates in the population protocol model.

In addition to such computability researches, many algorithmic problems have been considered in the population protocol model. For example, they include leader election [5]–[12], counting [13]–[16], and majority [5], [17]–[19]. These problems are considered under various assumptions of a population with or without a base station, symmetric or asymmetric protocols, designated or arbitrary initial states. The leader election problem has been thoroughly studied for both designated and arbitrary initial states. For designated initial states, many researches aim to minimize the time and space complexity [5], [6], [9]. For arbitrary initial states, many papers have developed self-stabilizing and loosely-stabilizing protocols [7], [8], [10]–[12]. Cai et al. [8] proposed a self-stabilizing leader election protocol with knowledge of n, and proved that knowledge of n is necessary to construct a self-stabilizing leader election protocol, where n is the number of agents. To overcome the requirement of knowledge of n, Sudo et al. [12] proposed a concept of loose stabilization and gave a loosely-stabilizing leader election protocol. The complexity and the requirement on communication graphs are improved later [10], [11]. The counting problem aims to count the number of agents and it has been studied under assumptions of a single BS and arbitrary initial states. After the first protocol was proposed in [15], the space complexity was gradually minimized [14], [16]. In [13], a time and space optimal protocol was proposed. The majority problem is also a fundamental problem in the population protocol model. In this problem, each agent initially has a color x or y, and the goal is to decide which color gets a majority. For the majority problem, many protocols have been proposed [5], [17]–[19]. Recently an asymptotically space-optimal protocol for c colors ($c > 2$) has been proposed in [19].

As a similar problem to the uniform bipartition problem, a group composition problem is studied in [20], [21]. Delporte-Gallet et al. [20] proposed a protocol to divide a population into g groups of almost the same size. The protocol is asymmetric, assumes designated initial states, and works under global fairness in the model of no BS. When $g = 2$, the protocol solves the uniform bipartition problem with three states. However, the paper does not consider other setting. Lamani et al. [21] studied a problem that divides a population into groups of designated sizes. Although the proposed protocols assume arbitrary initial states, they also assume that $n/2$ pairs of agents make interactions at the same time and that agents know n. In addition, the protocol requires n states, that is, it is not a constant-space protocol.

2. Definitions

2.1 Population Protocol Model

A population A is defined as a collection of pairwise interacting agents. A protocol is defined as $P = (Q, \delta)$, where Q is a set of possible states of agents and δ is a set of transitions on Q. Each transition in δ is described in the form $(p, q) \to (p', q')$, which means that, when an agent in state p and an agent in state q interact, they change their states to p' and q', respectively. We consider only deterministic protocols, that is, for every pair $(p, q) \in Q \times Q$, there exists at most one pair $(p', q') \in Q \times Q$ such that transition $(p, q) \to (p', q')$ is in δ. If transition $(p, q) \to (p', q')$ satisfies $p = q$ and $p' \neq q'$, the transition is asymmetric. We assume state changes by asymmetric transitions are decided deterministically, that is, when a_i and a_j in states p change their states by transition $(p, p) \to (p', q')$, 1) a_i and a_j always enter p' and q', respectively, or 2) a_i and a_j always enter q' and p', respectively. A transition is symmetric if it is not asymmetric. For protocol $P = (Q, \delta)$, P is symmetric if every transition in δ is symmetric, and P is asymmetric if every transition in δ is symmetric or asymmetric. Note that a symmetric protocol is also asymmetric.

A global state of a population is called a configuration. A configuration is defined as a vector of (local) states of all agents. We define $s(a, C)$ as the state of agent a at configuration C. When C is clear from the context, we simply write $s(a)$. If configuration C' is obtained from configuration C by a single transition of a pair of agents, we say $C \to C'$. For configurations C and C', if there is a sequence of configurations $C = C_0, C_1, \ldots, C_k = C'$ that satisfies $C_i \to C_{i+1}$ for any i $(0 \leq i < k)$, we say C' is reachable from C, denoted by $C \xrightarrow{*} C'$.

If an infinite sequence of configurations $E = C_0, C_1, C_2, \ldots$ satisfies $C_i \to C_{i+1}$ for any i $(i \geq 0)$, E is an execution of a protocol. An execution E is globally fair if, for every pair of configurations C and C' such that $C \to C'$, C' occurs infinitely often when C occurs infinitely often. Intuitively, global fairness represents that, when the current configuration is C, the system can transit, with a positive probability, to any configuration C' such that $C \to C'$ holds. This implies that, if the system reaches configuration C infinitely many times, the system infinitely many times transits to any C' such that $C \to C'$ holds. If C occurs infinitely often, C' satisfying $C \to C'$ occurs infinitely often, and consequently C'' satisfying $C' \to C''$ also occurs infinitely often. This implies that, under global fairness, if C occurs infinitely often, every configuration C^* reachable from C also occurs infinitely often. Note that global fairness does not put a condition on a finite sequence of interactions. From this property, in some impossibility proofs, we construct a globally fair execution such that some artificial sequence of interactions make the uniform bipartition problem unsolvable.

In this paper, we consider two models, one with a single BS (base station) and one with no BS. In the model with a single BS, we assume that a single agent called a BS exists in A. The BS is distinguishable from other non-BS agents while non-BS agents are identical and cannot be distinguished. That is, state set Q is divided into state set Q_b of a BS and state set Q_p of non-BS agents. The BS can be as powerful as needed, in contrast with resource-limited non-BS agents. That is, we focus on the number of states $|Q_p|$ for non-BS agents and do not care the number of states $|Q_b|$ for the BS. In addition, even if we consider protocols with arbitrary initial states, we assume that the BS has a designated initial state while all non-BS agents have arbitrary initial states. If we consider protocols with designated initial states, all non-BS agents have the same designated initial states and the BS has another designated initial state. In the model with no BS, no BS exists and all agents are identical. In this case, they all have the same designated initial states or arbitrary initial states. In both models, no agent knows the total number of agents in the initial configuration.

2.2 Uniform Bipartition Problem

Let A_p be a set of all non-BS agents. Let $f : Q_p \to \{red, blue\}$ be a function that maps a state of a non-BS agent to *red* or *blue*. We define a color of $a \in A_p$ as $f(s(a))$. We say agent $a \in A_p$ is *red* if $f(s(a)) = red$ and agent $a \in A_p$ is *blue* if $f(s(a)) = blue$.

Configuration C is stable if there is a partition $\{R, B\}$ of A_p that satisfies the following condition: 1) $||R| - |B|| \leq 1$, and 2) for every C^* such that $C \xrightarrow{*} C^*$, each agent in R is *red* and each agent in B is *blue* at C^*.

An execution $E = C_0, C_1, C_2, \ldots$ solves the uniform bipartition problem if there is a stable configuration C_t in E. If each execution E of protocol P solves the uniform bipartition problem, we say protocol P solves the uniform bipartition problem. The main objective of this paper is to minimize the number of states for non-BS agents. Since the BS is powerful, we do not care the number of states for the BS. When protocol P requires x states for non-BS agents, we say P is a protocol with x states.

For simplicity, we use agents only to refer to non-BS agents in the following sections. To refer to the BS, we always use the BS (not an agent).

3. Uniform Bipartition Protocols with a Single BS

In this section, we consider the uniform bipartition problem under the assumption of a single BS. Recall that the BS is distinguishable from other non-BS agents, and we do not care the number of states for the BS.

3.1 Protocols with Designated Initial States

In this subsection, we consider protocols with designated initial states. We give a simple symmetric protocol with three states, and then prove that there exists no asymmetric protocol with two states. This implies that, in this case, three states are sufficient for asymmetric or symmetric protocols.

3.1.1 A Protocol with Three States

In this protocol, the state set of (non-BS) agents is $Q_p = \{initial, red, blue\}$, and we set $f(initial) = f(red) = red$ and $f(blue) = blue$. The designated initial state of all agents is *initial*. The idea of the protocol is to assign states *red* and *blue* to agents alternately when agents interact with the BS. To realize this, the BS has a state set $Q_b = \{b_{red}, b_{blue}\}$, and its initial state is b_{red}. The protocol consists of the following two transitions.

1. $(b_{red}, initial) \rightarrow (b_{blue}, red)$
2. $(b_{blue}, initial) \rightarrow (b_{red}, blue)$

That is, when the BS in state b_{red} (resp., b_{blue}) and a non-BS agent in state *initial* interact, the BS changes the state of the non-BS agent to *red* (resp., *blue*) and the state of itself to b_{blue} (resp., b_{red}). When two non-BS agents interact, no state transition occurs. Clearly, all non-BS agents evenly transit to state *red* or *blue*, and the difference in the numbers of *red* and *blue* agents is at most one. Note that the protocol contains no asymmetric transition and works correctly if every non-BS agent interacts with the BS. Therefore, we have the following theorem.

Theorem 1: In the model with a single BS, there exists a symmetric protocol with three states and designated initial states that solves the uniform bipartition problem under global fairness.

3.1.2 Impossibility with Two States

Next, we show three states are necessary to construct an asymmetric protocol under global fairness. This implies that, in this case, three states are necessary for asymmetric or symmetric protocols under global fairness because a symmetric protocol is also asymmetric. That is, three states are necessary and sufficient in this case.

Theorem 2: In the model with a single BS, no asymmetric protocol with two states and designated initial states solves the uniform bipartition problem under global fairness.

Proof : For contradiction, assume that such a protocol *Alg* exists. Without loss of generality, we assume $Q_p = \{s_1, s_2\}, f(s_1) = red, f(s_2) = blue$, and that the designated initial state of all agents is s_1. Let n be an even number that is at least four. We consider the following three cases.

First, for population A of a single BS and n (non-BS) agents $a_1, a_2, \ldots, a_n$, consider a globally fair execution $E = C_0, C_1, \ldots$ of *Alg*. According to the definition, there exists a stable configuration C_t. That is, after C_t, the state of each agent does not change even if the BS and agents in states s_1 and s_2 interact in any order.

Next, for population A' of a single BS and $n + 2$ agents $a_1, a_2, \ldots, a_{n+2}$, we define an execution $E' = C'_0, C'_1, \ldots, C'_t, C'_{t+1}, \ldots$ of *Alg* as follows.

- From C'_0 to C'_t, the BS and n agents $a_1, a_2, \ldots, a_n$ interact in the same order as the execution E.
- After C'_t, the BS and $n+2$ agents interact so as to satisfy global fairness.

Since the BS and agents $a_1, \ldots, a_n$ change their states similarly to E from C'_0 to C'_t, there are $n/2 + 2$ agents in state s_1 and $n/2$ agents in state s_2 at C'_t. Moreover, the state of the BS at C'_t is the same as the state of the BS at C_t. However, since the difference in the numbers of *red* and *blue* agents is two, C'_t is not a stable configuration. Consequently, after C'_t, some *red* or *blue* agent changes its state in execution E'.

Lastly, we define execution $E'' = C''_0, C''_1, \ldots$ for population A as follows. First, we make agents transit similarly to E and reach stable configuration $C''_t (= C_t)$ in E''. After that we apply interactions in E' to execution E''. That is, we make agents interact as follows after C''_t in E'': 1) when the BS and an agent in state $s \in \{s_1, s_2\}$ interact at $C'_u \rightarrow C'_{u+1}$ $(u \geq t)$ in E', the BS and an agent in state s interact at $C''_u \rightarrow C''_{u+1}$ in E'', and 2) when two agents in states $s \in \{s_1, s_2\}$ and $s' \in \{s_1, s_2\}$ interact at $C'_u \rightarrow C'_{u+1}$ $(u \geq t)$ in E', two agents in states s and s' interact at $C''_u \rightarrow C''_{u+1}$ in E''. We can realize such interactions because, after stable configuration C''_t, at least two agents are in s_1 and at least two agents are in s_2. After C''_t, since interactions occur similarly to E', some *red* or *blue* agent changes its state similarly to E'. After such a state change occurs, we make agents interact so that E'' satisfies global fairness. This implies that, in globally fair execution E'', an agent changes its color after stable configuration C''_t. This is a contradiction. □

3.2 Protocols with Arbitrary Initial States

In this subsection, we consider protocols with arbitrary initial states. As a result, we give a symmetric protocol with four states, and prove impossibility of protocols with three states. That is, we show that four states are necessary and sufficient to construct a (symmetric or asymmetric) protocol in this case. Recall that, since a BS is powerful, the BS can start the protocol from a designated initial state.

3.2.1 A Symmetric Protocol with Four States

Here we show a symmetric protocol with four states under global fairness. In this protocol, each (non-BS) agent x has two variables $rb_x \in \{red, blue\}$ and $mark_x \in \{0, 1\}$. Variable rb_x represents the color of agent x. That is, for state s of agent x, $f(s) = red$ holds if $rb_x = red$ and $f(s) = blue$ holds if $rb_x = blue$. We define $\#red$ as the number of *red* agents and $\#blue$ as *blue* agents. We explain the role of variable $mark_x$ later.

The basic strategy of the protocol is that the BS counts *red* and *blue* agents by counting protocol *Count* [14] and changes colors of agents so that the numbers of *red* and *blue* agents become equal. Protocol *Count* is a symmetric protocol that counts the number of non-BS agents from arbitrary initial states under global fairness. Protocol *Count* uses only two states for each non-BS agent. We use variable $mark_x$ to maintain the state of protocol *Count*. In protocol *Count*, the BS has variable $Count.out$ that eventually outputs the number of agents. More concretely, $Count.out$ initially has value 0, gradually increases one by one, eventually equals to the number of agents, and stabilizes. The following lemma explains the characteristic of protocol *Count*.

Lemma 1 ([14]): Let n be the number of non-BS agents. In the initial configuration, $Count.out = 0$ holds. When $Count.out < n$, $Count.out$ eventually increases by one under global fairness. When $Count.out = n$, $Count.out$ never changes and stabilizes.

To count *red* and *blue* agents, the BS executes two instances of protocol *Count* in parallel to the main procedure of the uniform bipartition protocol. We denote by $Count_{red}$ and $Count_{blue}$ instances of protocol *Count* to count *red* and *blue* agents, respectively. The BS executes $Count_{red}$ when it interacts with a *red* agent. That is, the BS updates variables of $Count_{red}$ at the BS and the *red* agent by applying a transition of protocol $Count_{red}$. By this behavior, the BS executes $Count_{red}$ as if the population contains only *red* agents. Therefore, after the BS initializes its own variables of $Count_{red}$, it can correctly count the number of *red* agents by $Count_{red}$ (i.e., $Count_{red}.out$ eventually stabilizes to $\#red$) as long as a set of *red* agents does not change. Similarly, the BS executes $Count_{blue}$ when it interacts with a *blue* agent, and counts the number of *blue* agents. The straightforward approach to use the counting protocols is to adjust colors of agents after $Count_{red}.out$ and $Count_{blue}.out$ stabilize. However, the BS cannot know whether the outputs have stabilized or not. For this reason, the BS maintains estimated numbers of *red* and *blue* agents, and it changes colors of agents when the difference in the estimated numbers of *red* and *blue* agents is two. Note that, since the counting protocols assume that a set of counted agents does not change, the BS must restart $Count_{red}$ and $Count_{blue}$ from the beginning when the BS changes colors of some agents.

We explain the details of this procedure. The BS records the estimated numbers of *red* and *blue* agents in variables $C^*_{rb}[red]$ and $C^*_{rb}[blue]$, respectively. In the beginning of execution, these variables are identical to outputs of $Count_{red}$ and $Count_{blue}$. If the difference between $C^*_{rb}[red]$ and $C^*_{rb}[blue]$ becomes two, the BS immediately changes colors of agents. At the same time, the BS updates $C^*_{rb}[red]$ and $C^*_{rb}[blue]$ to reflect the change of colors. After the BS changes colors of some agents, it restarts $Count_{red}$ and $Count_{blue}$ from the beginning by initializing its own variables of the counting protocols. Since the counting protocols allow arbitrary initial states of non-BS agents, the BS can correctly count *red* and *blue* agents after that. Note that the BS does not initialize $C^*_{rb}[red]$ and $C^*_{rb}[blue]$ because it knows such numbers of *red* and *blue* agents exist. If the output of $Count_{red}$ and $Count_{blue}$ exceeds $C^*_{rb}[red]$

Algorithm 1 Uniform bipartition protocol

Variables at BS:

$C^*_{rb}[c] (c \in \{red, blue\})$: the estimated number of c agents, initialized to 0

$Variables$: variables of $Count_c (c \in \{red, blue\})$

Variables at a mobile agent x:

$rb_x \in \{red, blue\}$: color of the agent, initialized arbitrarily

$mark_x \in \{0, 1\}$: a variable of $Count_c (c \in \{red, blue\})$, initialized arbitrarily

when a mobile agent x interacts with BS **do**
 update $mark_x$ and variables of $Count_{rb_x}$ at BS by applying a transition of $Count_{rb_x}$
 if $C^*_{rb}[rb_x] < Count_{rb_x}.out$ **then**
  $C^*_{rb}[rb_x] \leftarrow Count_{rb_x}.out$
 end if
 if $C^*_{rb}[rb_x] - C^*_{rb}[\overline{rb_x}] = 2$ **then**
  $C^*_{rb}[rb_x] \leftarrow C^*_{rb}[rb_x] - 1$
  $C^*_{rb}[\overline{rb_x}] \leftarrow C^*_{rb}[\overline{rb_x}] + 1$, $rb_x \leftarrow \overline{rb_x}$
  reset variables of $Count_{red}$ and $Count_{blue}$ at BS
 end if
end when

---

and $C^*_{rb}[blue]$, the BS updates $C^*_{rb}[red]$ and $C^*_{rb}[blue]$, respectively. After that, if the difference between $C^*_{rb}[red]$ and $C^*_{rb}[blue]$ becomes two, the BS changes colors of agents. By repeating this behavior, the BS adjusts colors of agents.

The pseudocode of this protocol is given in Algorithm 1. We define $\overline{red} = blue$ and $\overline{blue} = red$. Recall that variable $mark_x$ is a two-state variable of counting protocols $Count_{red}$ and $Count_{blue}$. Since the BS restarts the counting protocols whenever it changes colors of agents, the BS keeps a set of *red* (resp., *blue*) agents unchanged until it restarts $Count_{red}$ (resp., $Count_{blue}$). In addition, each agent is involved in either $Count_{red}$ or $Count_{blue}$ at the same time. Hence it requires only a single variable $mark_x$ to execute $Count_{red}$ and $Count_{blue}$. When two non-BS agents interact, no state transition occurs in this protocol and counting protocols. When the BS and a *red* agent interact, they update $mark_x$ and variables of $Count_{red}$ at the BS by applying a transition of $Count_{red}$. This means that they execute $Count_{red}$ in parallel to the main procedure of the uniform bipartition protocol. After that, if $Count_{red}.out$ is larger than $C^*_{rb}[red]$, $C^*_{rb}[red]$ is updated with $Count_{red}.out$. If the difference between $C^*_{rb}[red]$ and $C^*_{rb}[blue]$ becomes two, the *red* agent changes its color to *blue* and the BS updates $C^*_{rb}[red]$ and $C^*_{rb}[blue]$. After updating, the BS resets variables of $Count_{red}$ and $Count_{blue}$, and restarts counting. When the BS and a *blue* agent interact, they behave similarly.

Lemma 2: In any configuration, $C^*_{rb}[red] \leq \#red$, $C^*_{rb}[blue] \leq \#blue$ and $|C^*_{rb}[red] - C^*_{rb}[blue]| \leq 1$ hold.

Proof : We prove by induction on the index $k \geq 0$ of a configuration in an execution $C_0, C_1, C_2, \ldots, C_k, \ldots$. At the initial configuration C_0, the lemma holds. Let us assume that the lemma holds for configuration C_k and prove it for configuration C_{k+1}. From this assumption, $C^*_{rb}[red] \leq \#red$, $C^*_{rb}[blue] \leq \#blue$ and $|C^*_{rb}[red] - C^*_{rb}[blue]| \leq 1$ hold at C_k. Assume that, when C_k transits to C_{k+1}, the BS and agent

x interact. If $Count_{rb_x}.out$ becomes larger than $C^*_{rb}[rb_x]$, the BS updates $C^*_{rb}[rb_x]$ by $C^*_{rb}[rb_x] \leftarrow Count_{rb_x}.out$ (line 3). Note that, in this case, $C^*_{rb}[rb_x]$ increases by one from Lemma 1. In addition, $C^*_{rb}[red] \leq \#red$ and $C^*_{rb}[blue] \leq \#blue$ still hold. Recall that $|C^*_{rb}[red] - C^*_{rb}[blue]| \leq 1$ held before this update and $C^*_{rb}[rb_x]$ increases by one. Consequently, at this moment (before line 5), $|C^*_{rb}[rb_x] - C^*_{rb}[\overline{rb_x}]| \leq 1$ or $C^*_{rb}[rb_x] - C^*_{rb}[\overline{rb_x}] = 2$ holds. Next, we consider lines 5 to 9. If $C^*_{rb}[rb_x] - C^*_{rb}[\overline{rb_x}] \leq 1$ at line 5, lines 6 to 8 are not executed, and thus $C^*_{rb}[red] \leq \#red$, $C^*_{rb}[blue] \leq \#blue$ and $|C^*_{rb}[red] - C^*_{rb}[blue]| \leq 1$ hold. If $C^*_{rb}[rb_x] - C^*_{rb}[\overline{rb_x}] = 2$ at line 5, agent x changes its color from rb_x to $\overline{rb_x}$, $C^*_{rb}[rb_x]$ decreases by one, and $C^*_{rb}[\overline{rb_x}]$ increases by one. This also preserves $C^*_{rb}[red] \leq \#red$, $C^*_{rb}[blue] \leq \#blue$ and $|C^*_{rb}[red] - C^*_{rb}[blue]| \leq 1$. Therefore, the lemma holds. □

Theorem 3: Algorithm 1 solves the uniform bipartition problem. That is, in the model with a BS, there exists a symmetric protocol with four states and arbitrary initial states that solves the uniform bipartition problem under global fairness.

Proof : We define $phase = C^*_{rb}[red] + C^*_{rb}[blue]$. Initially, $phase = 0$ holds. We show that 1) $phase$ increases one by one if $phase < n$, and 2) Algorithm 1 solves the uniform bipartition problem if $phase = n$.

First consider the initial configuration. Since we assume global fairness, $Count_{red}.out$ or $Count_{blue}.out$ increases by one from Lemma 1 and at that time $phase$ increases by one.

Let us consider the transition $C \to C'$ such that $phase$ increases by one (i.e., line 4 is executed) and $phase < n$ holds at C'. We consider two cases.

- Case that lines 7 to 9 are not executed at $C \to C'$. In this case, since the BS does not change sets of *red* and *blue* agents, it can correctly continue to execute $Count_{red}$ and $Count_{blue}$. Since $phase < n = \#red + \#blue$ holds, either $\#red > C^*_{rb}[red]$ or $\#blue > C^*_{rb}[blue]$ holds. Consequently, from Lemma 1, either $Count_{red}.out > C^*_{rb}[red]$ or $Count_{blue}.out > C^*_{rb}[blue]$ holds eventually because we assume global fairness. At that time, $C^*_{rb}[red]$ or $C^*_{rb}[blue]$ increases by one and hence $phase$ increases by one.
- Case that lines 7 to 9 are executed at $C \to C'$. In this case, the BS changes sets of *red* and *blue* agents. At that time, the BS initializes its own variables of counting algorithms $Count_{red}$ and $Count_{blue}$. Since the counting algorithms work from arbitrary initial states of agents, the BS can correctly execute $Count_{red}$ and $Count_{blue}$ from the beginning under global fairness. Similarly to the first case, from Lemma 1, either $Count_{red}.out > C^*_{rb}[red]$ or $Count_{blue}.out > C^*_{rb}[blue]$ holds eventually. Then, $phase$ increases by one.

Lastly, consider the transition $C \to C'$ such that $phase$ increases by one and $phase = n$ holds at C'. From $phase = n$, $C^*_{rb}[red] + C^*_{rb}[blue] = n = \#red + \#blue$ holds, and consequently $C^*_{rb}[red] = \#red$ and $C^*_{rb}[blue] = \#blue$ hold from Lemma 2. This implies that $Count_{red}.out$ and $Count_{blue}.out$ never exceed $C^*_{rb}[red]$ and $C^*_{rb}[blue]$ after that, respectively. Therefore, $C^*_{rb}[red]$ and $C^*_{rb}[blue]$ are never updated and consequently agents never change their colors any more. Since $|\#red - \#blue| = |C^*_{rb}[red] - C^*_{rb}[blue]| \leq 1$ holds from Lemma 2, we have the theorem. □

3.2.2 Impossibility with Three States

Theorem 4: In the model with a single BS, no asymmetric protocol with three states and arbitrary initial states solves the uniform bipartition problem under global fairness.

Proof : For contradiction, assume that such a protocol Alg exists. Without loss of generality, we assume that the state set of agents is $Q_p = \{s_1, s_2, s_3\}$, $f(s_1) = f(s_2) = red$, and $f(s_3) = blue$. We consider the following three cases.

First, consider population $A = \{a_0, \ldots, a_n\}$ of a single BS and n agents such that n is even and at least 4. Assume that a_0 is a BS. Since each agent has an arbitrary initial state, we consider an initial configuration C_0 such that $s(a_i) = s_3$ holds for any $i (1 \leq i \leq n)$. Note that the BS a_0 has a designated initial state at C_0. From the definition of Alg, for any globally fair execution $E = C_0, C_1, \ldots$, there exists a stable configuration C_t. Hence, both the number of *red* agents and the number of *blue* agents are $n/2$ at C_t. After C_t, the color of agent a_i (i.e., $f(s(a_i))$) never changes for any $a_i (1 \leq i \leq n)$ even if the BS and agents interact in any order.

Next, consider population $A' = \{a'_0, \ldots, a'_{n+2}\}$ of a single BS and $n + 2$ agents. Assume that agent a'_0 is a BS. We consider an initial configuration C'_0 such that $s(a'_i) = s_3$ holds for any i $(1 \leq i \leq n + 2)$. From this initial configuration, we define an execution $E' = C'_0, C'_1, \ldots, C'_t, \ldots$ using the execution E as follows.

- For $0 \leq u < t$, when a_i and a_j interact at $C_u \to C_{u+1}$, a'_i and a'_j interact at $C'_u \to C'_{u+1}$.
- For $t \leq u$, an interaction occurs at $C'_u \to C'_{u+1}$ so that E' satisfies global fairness.

Since the BS and agents $a_1, \ldots, a_n$ change their states similarly to E from C'_0 to C'_t, $s(a'_i) = s(a_i)$ holds for $1 \leq i \leq n$. Hence, there exist $n/2$ *red* agents and $n/2 + 2$ *blue* agents at C'_t. Consequently C'_t is not a stable configuration. This implies that there exists a stable configuration $C'_{t'}$ for some $t' > t$. Clearly at least one *blue* agent becomes *red* from C'_t to $C'_{t'}$. That is, for some configuration $C'_{t^*} (t \leq t^* < t')$, an agent in state s_3 transits to state s_1 or s_2 at $C_{t^*} \to C_{t^*+1}$. Assume that t^* is the smallest value that satisfies the condition.

Finally, for A we define an execution $E'' = C''_0, C''_1, \ldots$ using executions E and E' as follows.

- Let $C''_u = C_u$ for $0 \leq u \leq t$. That is, E'' reaches stable configuration C''_t in similarly to E.
- For $t \leq u \leq t^*$, we define an execution so that interaction at $C'_u \to C'_{u+1}$ also occurs at $C''_u \to C''_{u+1}$.

Concretely, when a'_i and a'_j interact at $C'_u \rightarrow C'_{u+1}$, we define $a_{i'}$ and $a_{j'}$ as follows and they interact at $C''_u \rightarrow C''_{u+1}$. If $i \le n$, let $i' = i$. Otherwise, since $s(a'_i) = s_3$ holds at C'_u (because no agent in state s_3 changes its state from C'_t to C'_{t^*}), choose $i'(\le n)$ such that both $s(a_{i'}) = s_3$ and $i' \ne j$ hold. Similarly, if $j \le n$, let $j' = j$. Otherwise choose $j'(\le n)$ such that both $s(a_{j'}) = s_3$ and $j' \ne i'$ hold. Such i' and j' exist since at least two agents in state s_3 exist (because $n \ge 4$ holds and no agent in state s_3 changes its state from C'_t to C'_{t^*}).

- After $t^* < u$, an interaction occurs at $C''_u \rightarrow C''_{u+1}$ so that E'' satisfies global fairness.

Clearly, for $t \le u \le t^*$ and $i \le n$, $s(a_i)$ at C''_u is equal to $s(a'_i)$ at C'_u. Additionally, at $C''_{t^*} \rightarrow C''_{t^*+1}$, an agent in state s_3 transits to s_1 or s_2 as well as $C'_{t^*} \rightarrow C'_{t^*+1}$. This means that the agent changes its color at $C''_{t^*} \rightarrow C''_{t^*+1}$. That is, an agent changes its color after stable configuration C''_t in globally fair execution E''. This is a contradiction. □

Remark 5: Note that, in the proof of Theorem 4, we consider a protocol with $Q_p = \{s_1, s_2, s_3\}$, $f(s_1) = f(s_2) = red$, and $f(s_3) = blue$, and assume that every agent is in state s_3 at the initial configuration of E, E', and E''. This means, even if we consider a protocol with three states and designated initial states, there exists no protocol such that the designated initial state does not have the same color as any other state. This fact holds even if the number of states is larger than three.

On the other hand, Sect. 3.1.1 gives a protocol with three states and designated initial states. In the protocol, the state set of agents is $Q_p = \{initial, red, blue\}$, we set $f(initial) = f(red) = red$ and $f(blue) = blue$, and the designated initial state is *initial*. This implies that there exists a protocol if the designated initial state (i.e., *initial*) has the same color as one of other states (i.e., *red*). □

4. Uniform Bipartition Protocols with No BS

In this section, we consider the uniform bipartition problem under the assumption of no BS. That is, all agents are identical.

4.1 Protocols with Designated Initial States

In this subsection, we consider protocols with designated initial states. Since we consider the model with no BS, all agents have the same initial state in the initial configuration.

4.1.1 Asymmetric Protocols

First, we consider asymmetric protocols in this case. Since three states are necessary in the model with a BS from Theorem 2, three states are also necessary in the model with no BS. In addition, Delporte-Gallet et al. [20] gives a protocol with three states. This implies that three states are necessary and sufficient in this case.

Here, we briefly explain the protocol proposed in [20]. In this protocol, the state set of agents is $Q_p = \{initial, red, blue\}$, and we set $f(initial) = f(red) = red$ and $f(blue) = blue$. The designated initial state of all agents is *initial*. The protocol consists of a single asymmetric transition $(initial, initial) \rightarrow (red, blue)$. In this protocol, when two agents in state *initial* interact, one agent transits to *red* and the other transits to *blue*. This implies that the number of agents in state *red* is always the same as the number of agents in state *blue*. Eventually all agents (possibly except one agent) transit to state *red* or *blue*. From $f(initial) = red$, the difference in the numbers of *red* and *blue* agents is at most one. Note that the protocol works correctly if every pair of agents interacts once.

Theorem 6 ([20]): In the model with no BS, there exists an asymmetric protocol with three states and designated initial states that solves the uniform bipartition problem under global fairness.

4.1.2 Symmetric Protocols

Next, we consider symmetric protocols in this case. For this setting, we show a protocol with four states and impossibility with three states. These results show that, in this case, four states are necessary and sufficient to construct a symmetric protocol under global fairness.

(1) A protocol with four states under global fairness

We can easily obtain a symmetric protocol with four states by a scheme proposed in [23]. The scheme transforms an asymmetric protocol with α states to a symmetric protocol with at most 2α states. By applying the scheme to an asymmetric protocol in Sect. 4.1.1 and deleting unnecessary states, we can obtain a symmetric protocol with four states.

For self-containment, we briefly explain the obtained protocol. Since no symmetric protocol solves the uniform bipartition problem for a population of two agents, we assume that a population consists of at least three agents. In this protocol, the state set of agents is $Q_p = \{initial, initial', red, blue\}$, and we set $f(initial) = f(initial') = f(red) = red$ and $f(blue) = blue$. The designated initial state of all agents is *initial*. The protocol consists of the following seven transitions.

1. $(initial, initial) \rightarrow (initial', initial')$
2. $(initial', initial') \rightarrow (initial, initial)$
3. $(initial, initial') \rightarrow (red, blue)$
4. $(initial, red) \rightarrow (initial', red)$
5. $(initial, blue) \rightarrow (initial', blue)$
6. $(initial', red) \rightarrow (initial, red)$
7. $(initial', blue) \rightarrow (initial, blue)$

The main behavior of the protocol is similar to the previous asymmetric protocol with three states. However, since asymmetric transition $(initial, initial) \rightarrow (red, blue)$ is not

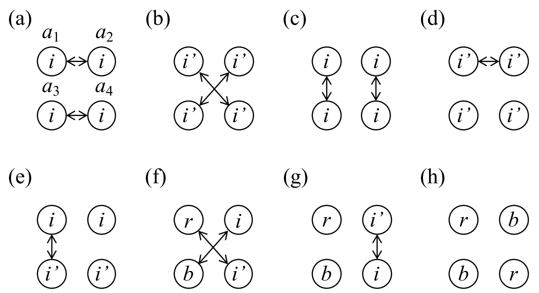

Fig. 1 An example execution of the protocol. Symbols i, i', r, and b represent states *initial*, *initial′*, *red*, and *blue*, respectively. Arrows represent interactions of agents.

allowed in symmetric protocols, the scheme in [23] introduces a new state *initial′*. Transition 3 implies that, when agents in states *initial* and *initial′* interact, they become *red* and *blue*, respectively. In addition, agents in states *initial* and *initial′* become *initial′* and *initial* respectively when they interact with some agents (except for interaction between two agents in states *initial* and *initial′*). From global fairness, if at least two agents are in state *initial* or *initial′*, some two agents eventually enter states *initial* and *initial′*. After that, if the two agents interact, they enter states *red* and *blue*. Note that, since $f(initial) = f(initial') = red$ holds, the protocol solves the problem even if the number of agents is odd and an agent with state *initial* or *initial′* remains forever.

Figure 1 shows an example execution of the protocol for a population of four agents. Initially all agents are in state *initial* (Fig. 1 (a)). After interactions (a_1, a_2) and (a_3, a_4), all agents enter state *initial′* (Fig. 1 (b)). Similarly, after interactions (a_1, a_4), (a_2, a_3), (a_1, a_3), and (a_2, a_4), all agents have the same state (Fig. 1 (c) and (d)). If these interactions happen infinite times, all agents keep the same state and never achieve the uniform bipartition. However, under the global fairness, such interactions do not happen infinite times. This is because, if some configuration C occurs infinite times, every configuration reachable from C should occur. This implies that eventually interactions (a_1, a_2) and (a_1, a_3) happen in this order from a configuration in Fig. 1 (d). Then, a_1 and a_3 enter states *red* and *blue*, respectively (Fig. 1 (e) and (f)). After that, in a similar way, the remaining agents eventually enter *red* and *blue* like Fig. 1 (g) and (h).

Theorem 6 and correctness of the scheme in [23] derives the following theorem.

Theorem 7: In the model with no BS, when the number of agents is at least three, there exists a symmetric protocol with four states and designated initial states that solves the uniform bipartition problem under global fairness.

(2) Impossibility with three states

Theorem 8: In the model with no BS, no symmetric protocol with three states and designated initial states solves the uniform bipartition problem under global fairness.

Proof : For contradiction, assume that such a protocol *Alg* exists. Without loss of generality, we assume that the state set of agents is $Q_p = \{s_1, s_2, s_3\}$, $f(s_1) = f(s_2) = red$, and $f(s_3) = blue$. Consider population $A = \{a_1, \ldots, a_n\}$ of n agents such that n is even and at least 6. First, assume that the designated initial state of all agents is s_3. Clearly, *Alg* has transition $(s_3, s_3) \to (s_i, s_i)$ for some $i \neq 3$. However, since $n/2$ agents in state s_3 exist at a stable configuration, some agents change their states from s_3 to s_i at the stable configuration. This implies that agents change their colors. Therefore, a designated initial state is s_1 or s_2.

Next, assume that the designated initial state of all agents is s_1 (Case of s_2 is the same). Since *Alg* is a symmetric protocol and all the initial states are s_1, *Alg* includes $(s_1, s_1) \to (s_i, s_i)$ for some $i \neq 1$. This implies that all agents can transit to state s_i from the initial configuration. Hence, *Alg* also includes $(s_i, s_i) \to (s_j, s_j)$ for some $j \neq i$. When $i = 3$, since $n/2$ *blue* agents exist at a stable configuration and they are in state s_3, the *blue* agents become *red* by transition $(s_3, s_3) \to (s_j, s_j)$. Therefore, $i \neq 3$ holds.

The remaining case is $i = 2$. If $j = 3$, that is, *Alg* includes $(s_2, s_2) \to (s_3, s_3)$, *red* agents (i.e., agents in state s_1 or s_2) change their colors at a stable configuration because *Alg* includes $(s_1, s_1) \to (s_2, s_2)$ and $(s_2, s_2) \to (s_3, s_3)$. This implies $j = 1$. In this case, *Alg* includes $(s_2, s_2) \to (s_1, s_1)$. Since some agents should transit to state s_3, *Alg* includes $(s_1, s_2) \to (s_k, s_l)$ such that k or l is 3. At a stable configuration, there exist $n/2$ agents with states s_1 or s_2. However, these agents can transit to state s_3 from transitions $(s_1, s_2) \to (s_k, s_l)$, $(s_2, s_2) \to (s_1, s_1)$, and $(s_1, s_1) \to (s_2, s_2)$. This is a contradiction. □

4.2 Protocols with Arbitrary Initial States

In this subsection, we consider protocols with arbitrary initial states. We show that, in this case, no protocol solves the uniform bipartition problem. That is, to allow agents to start from arbitrary initial states, a single BS is necessary.

Theorem 9: In the model with no BS, no asymmetric protocol with arbitrary initial states solves the uniform bipartition problem under global fairness.

Proof : For contradiction, assume that such a protocol *Alg* exists. Assume that n is even and at least 4. We consider the following three cases.

First, for population $A = \{a_1, \ldots, a_n\}$ of n agents, consider a globally fair execution $E = C_0, C_1, \ldots$ of *Alg*. From the definition of *Alg*, there exists a stable configuration C_t. Hence, both the number of *red* agents and the number of *blue* agents are $n/2$ at C_t. After C_t, the color of agent a_i (i.e., $f(s(a_i))$) never changes for any a_i ($1 \leq i \leq n$) even if agents interact in any order.

Next, for population $A' = \{a'_i | f(s(a_i, C_t)) = red\}$ of $n/2$ agents, consider an execution $E' = C'_0, C'_1, \ldots$ of *Alg* from the initial configuration C'_0 such that $s(a'_i, C'_0) = s(a_i, C_t)$ holds for any i ($1 \leq i \leq n/2$). Since all agents are *red* at

C_0', some agents must change their colors to reach a stable configuration.

Lastly we consider execution E'' for population A as follows. First agents interact similarly to E and reach the same stable configuration as C_t. Then, $n/2$ *red* agents interact similarly to E'. From the definition of E', some agents change their colors. After that, agents interact to satisfy global fairness. This implies that, in globally fair execution E'', some agents change their colors after a stable configuration. This is a contradiction. □

5. Conclusion

In this paper, we completely clarify solvability of the uniform bipartition problem under global fairness and minimum requirements of agent space under various assumptions. This paper leaves many open problems:

- Is it possible to extend our results to the uniform k-partition problem, which divides a population into k groups of the same size, for arbitrary k? Note that we can easily construct a uniform k-partition protocol for $k = 2^h$ by repeating the described uniform bipartition protocol h times. When we assume designated initial states, protocols in Sects. 3.1.1, 4.1.1, and 4.1.2 guarantee that each agent never changes its state after it enters *red* or *blue*. Hence, after each agent becomes *red* or *blue* in the i-th protocol (i.e., the protocol for 2^i-partition) for $i < h$, it can start the $(i+1)$-th protocol (i.e., the protocol for 2^{i+1}-partition). When we assume a single BS and arbitrary initial states, the BS can control the execution of h protocols. That is, if the BS changes a color of an agent in the i-th protocol, it can restart the i'-th protocol for each $i' \geq i+1$ by initializing variables of the i'-th protocol on the BS. By repeating this behavior, the population eventually stabilizes to a uniform 2^h-partition.
  On the other hand, it is difficult to extend the protocol to the case of $k \neq 2^h$. As described in Sect. 1, we can approximately achieve the uniform k-partition by regrouping $k'(= 2^h > k)$ groups into k groups with almost the same size. However, to exactly achieve the uniform k-partition, we require a protocol specific to the uniform k-partition.
- What is the relation between the uniform bipartition problem and other problems such as counting, leader election, and majority?
- What is the time complexity of the uniform bipartition problem under probabilistic fairness? The uniform bipartition problem has a close relationship to computation of function $f(n) = n/2$. The time complexity of $n/2$ computation has been studied in [24], [25]. Is it possible to derive the time complexity of the uniform bipartition problem from the results?
- Is it possible to characterize other initial configurations that can achieve the uniform bipartition with a small number of states? We considered two extremes as initial configurations: a designated initial configuration, where all agents have the same state, and an arbitrary initial configuration, where all agents have arbitrary states. We can consider initial configurations between the two extremes, such as initial configuration where one agent has a unique leader state and other agents have other arbitrary states. Is it possible to achieve the uniform bipartition with a small number of states from such initial configurations?

References

[1] H. Yasumi, F. Ooshita, K. Yamaguchi, and M. Inoue, "Constant-space population protocols for uniform bipartition," the 21st International Conference on Principles of Distributed Systems, 2017.
[2] D. Angluin, J. Aspnes, Z. Diamadi, M.J. Fischer, and R. Peralta, "Computation in networks of passively mobile finite-state sensors," Distrib. Comput., vol.18, no.4, pp.235–253, 2006.
[3] S. Murata, A. Konagaya, S. Kobayashi, H. Saito, and M. Hagiya, "Molecular robotics: A new paradigm for artifacts," New Generat. Comput., vol.31, no.1, pp.27–45, 2013.
[4] J. Aspnes and E. Ruppert, "An introduction to population protocols," Middleware for Network Eccentric and Mobile Applications, pp.97–120, 2009.
[5] D. Alistarh, J. Aspnes, D. Eisenstat, R. Gelashvili, and R.L. Rivest, "Time-space trade-offs in population protocols," Proc. 28th Annual ACM-SIAM Symposium on Discrete Algorithms, pp.2560–2579, 2017.
[6] D. Alistarh and R. Gelashvili, "Polylogarithmic-time leader election in population protocols," Proc. 42nd International Colloquium on Automata, Languages, and Programming, vol.9135, pp.479–491, 2015.
[7] D. Angluin, J. Aspnes, M.J. Fischer, and H. Jiang, "Self-stabilizing population protocols," International Conference On Principles Of Distributed Systems, vol.3974, pp.103–117, Springer, 2005.
[8] S. Cai, T. Izumi, and K. Wada, "How to prove impossibility under global fairness: On space complexity of self-stabilizing leader election on a population protocol model," Theor. Comput. Syst., vol.50, no.3, pp.433–445, 2012.
[9] D. Doty and D. Soloveichik, "Stable leader election in population protocols requires linear time," Proc. International Symposium on Distributed Computing, vol.9363, pp.602–616, 2015.
[10] T. Izumi, "On space and time complexity of loosely-stabilizing leader election," Proc. International Colloquium on Structural Information and Communication Complexity, vol.9439, pp.299–312, 2015.
[11] Y. Sudo, T. Masuzawa, A.K. Datta, and L.L. Larmore, "The same speed timer in population protocols," Proc. International Conference on Distributed Computing Systems, pp.252–261, 2016.
[12] Y. Sudo, J. Nakamura, Y. Yamauchi, F. Ooshita, H. Kakugawa, and T. Masuzawa, "Loosely-stabilizing leader election in a population protocol model," Theor. Comput. Sci., vol.444, pp.100–112, 2012.
[13] J. Aspnes, J. Beauquier, J. Burman, and D. Sohier, "Time and space optimal counting in population protocols," Proc. International Conference on Principles of Distributed Systems, pp.13:1–13:17, 2016.
[14] J. Beauquier, J. Burman, S. Claviere, and D. Sohier, "Space-optimal counting in population protocols," Proc. International Symposium on Distributed Computing, vol.9363, pp.631–646, 2015.
[15] J. Beauquier, J. Clement, S. Messika, L. Rosaz, and B. Rozoy, "Self-stabilizing counting in mobile sensor networks with a base station," Proc. International Symposium on Distributed Computing, pp.63–76, 2007.
[16] T. Izumi, K. Kinpara, T. Izumi, and K. Wada, "Space-efficient self-

stabilizing counting population protocols on mobile sensor networks," Theor. Comput. Sci., vol.552, pp.99–108, 2014.
[17] D. Alistarh, R. Gelashvili, and M. Vojnović, "Fast and exact majority in population protocols," Proc. 2015 ACM Symposium on Principles of Distributed Computing, pp.47–56, 2015.
[18] D. Angluin, J. Aspnes, and D. Eisenstat, "A simple population protocol for fast robust approximate majority," Distrib. Comput., vol.21, no.2, pp.87–102, 2008.
[19] L. Gasieniec, D. Hamilton, R. Martin, P.G. Spirakis, and G. Stachowiak, "Deterministic population protocols for exact majority and plurality," Proc. International Conference on Principles of Distributed Systems, pp.14:1–14:14, 2016.
[20] C. Delporte-Gallet, H. Fauconnier, R. Guerraoui, and E. Ruppert, "When birds die: Making population protocols fault-tolerant," Distributed Computing in Sensor Systems, vol.4026, pp.51–66, 2006.
[21] A. Lamani and M. Yamashita, "Realization of periodic functions by self-stabilizing population protocols with synchronous handshakes," Proc. International Conference on Theory and Practice of Natural Computing, vol.10071, pp.21–33, 2016.
[22] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert, "The computational power of population protocols," Distrib. Comput., vol.20, no.4, pp.279–304, 2007.
[23] O. Bournez, J. Chalopin, J. Cohen, and X. Koegler, "Playing with population protocols," Electronic Proceedings in Theoretical Computer Science, vol.1, pp.3–15, 2009.
[24] D. Angluin, J. Aspnes, and D. Eisenstat, "Fast computation by population protocols with a leader," Distrib. Comput., vol.21, no.3, pp.183–199, 2008.
[25] A. Belleville, D. Doty, and D. Soloveichik, "Hardness of computing and approximating predicates and functions with leaderless population protocols," Proc. 44th International Colloquium on Automata, Languages, and Programming, pp.141:1–141:14, 2017.

Hiroto Yasumi received his B.E. degree in information engineering from National Institute of Technology, Nara College, Japan, in 2018. He is now a student of Graduate School of Science and Technology, Nara Institute of Science and Technology (NAIST). His research interests include distributed algorithms. He is a student member of IEICE.

Fukuhito Ooshita received the M.E. and D.I. degrees in computer science from Osaka University in 2002 and 2006. He had been an assistant professor in the Graduate School of Information Science and Technology at Osaka University during 2003–2015. He is now an associate professor of Graduate School of Information Science, Nara Institute of Science and Technology (NAIST). His research interests include parallel algorithms and distributed algorithms. He is a member of ACM, IEEE, IEICE, and IPSJ.

Ken'ichi Yamaguchi received his B.E. degree in information engineering from National Institute of Technology, Nara College, Japan, in 1999 and the M.E. and Ph.D. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 2001 and 2003, respectively. His research interests include VLSI CAD, design for testability, and high-level synthesis.

Michiko Inoue received her B.E., M.E, and Ph.D degrees in Computer Science from Osaka University in 1987, 1989, and 1995 respectively. She worked at Fujitsu Laboratories Ltd. from 1989 to 1991. She is a Professor of Graduate School of Science and Technology, Nara institute of Science and Technology (NAIST). Her research interests include distributed algorithms, graph theory and dependability of digital systems. She is a member of Science Council of Japan, the Information Processing Society of Japan (IPSJ), and Japanese Society for Artificial Intelligence, and a senior member of IEEE and the Institute of Electronics, Information and Communication Engineers (IEICE).