International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 9, Number 1, pages 97–110, January 2019

A Population Protocol for Uniform k-partition under Global Fairness

Hiroto Yasumi
Nara Institute of Science and Technology, yasumi.hiroto.yf9@is.naist.jp
Takayama 8916-5, Ikoma 630-0192, JAPAN

Naoki Kitamura
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555, JAPAN

Fukuhito Ooshita
Nara Institute of Science and Technology, f-oosita@is.naist.jp
Takayama 8916-5, Ikoma 630-0192, JAPAN

Taisuke Izumi
Nagoya Institute of Technology, t-izumi@nitech.ac.jp
Gokiso-cho, Showa-ku, Nagoya 466-8555, JAPAN

Michiko Inoue
Nara Institute of Science and Technology, kounoe@is.naist.jp
Takayama 8916-5, Ikoma 630-0192, JAPAN

Received: July 25, 2018
Accepted: September 14, 2018
Communicated by Akihiro Fujiwara

Abstract

In this paper, we consider a uniform k-partition problem in a population protocol model. The uniform k-partition problem divides a population into k groups of the same size. For this problem, we give a symmetric protocol with designated initial states under global fairness. The proposed protocol requires $3k - 2$ states for each agent. Since any protocol for the uniform k-partition problem requires $\Omega(k)$ states to indicate a group, the space complexity of the proposed protocol is asymptotically optimal.

Keywords: population protocol, uniform k-partition, distributed protocol

1 Introduction

1.1 Background and Our Contribution

A population protocol model [5] abstracts computation carried out by many mobile devices. Such devices are called agents and a set of agents is called a population. In the population protocol model, computation is proceeded by repeating the pairwise interactions of agents. If an interaction occurs between two agents, the states of the agents are updated. The population protocol model can be used for modeling many kinds of mobile networks. For example, a network of the sensors attached to wild birds is useful to observe the ecosystem. In this system, a pairwise interaction occurs when two sensors (i.e. birds) approach to each other. Sensors collect and process data based on their interactions. Another promising application is a network of molecular robots [20], which can be deployed to a human body for the diagnosis of its physical condition. To realize these systems, many protocols have been studied as building blocks in the population protocol model [10]. For example, leader election protocols [4, 13, 17, 21, 22, 23], counting protocols [9, 11, 12], majority protocols [6, 16], and so on.

In this paper, we focus on the uniform k-partition problem, which divides a population into k groups of the same size. The uniform k-partition problem has many applications. It can be used for reducing the energy consumption of the whole system by switching on some groups and switching off the others. In another example, we can assign different tasks to different groups and make agents execute multiple tasks at the same time. It is also possible to use uniform k-partition protocols for attaining fault-tolerance [14].

As prior work, Yasumi et al. [25] focused on the uniform k-partition problem for $k = 2$ (called uniform bipartition) and proposed several space-optimal protocols in various settings. In particular, it is proved that four states are necessary and sufficient to solve the uniform bipartition problem by a symmetric protocol under global fairness. Symmetric protocol is the restricted subclass of the protocols where any interaction among the two agents with the same state must result in the (other) same state. Global fairness is an assumption on schedules of interactions (the formal definition is given in Section 2). By repeating the uniform bipartition protocol h times, we can construct a uniform k-partition protocol for $k = 2^h$. However, it is difficult to extend the protocol to the case of $k \neq 2^h$. This is because the protocol strongly depends on nature of pairwise interactions. That is, in the protocol, when one agent becomes a member of one group by an interaction, the partner becomes a member of another group at the same time. This simple mechanism guarantees that each group contains the same number of agents. However, it is obviously impossible to divide k agents ($k > 2$) into k different groups only by a single interaction, and thus the strategy of the bipartition protocol is not easily extended to the general k-partition case. In the case of allowing less uniformity, Delporte-Gallet et al. [14] proposes a protocol solving the uniform k-partition problem approximately. This protocol guarantees that each group contains at least $n/(2k)$ agents, where n is the number of agents. This protocol requires $k(k+3)/2$ states under global fairness.

Our Contribution In this paper, we propose a protocol that solves the uniform k-partition problem for any k ($k \geq 2$). This protocol is symmetric and works under global fairness. Recall that, in symmetric protocols, when two agents in the same state interact, they transit to the same state. Such protocols do not require a mechanism to break symmetry among agents and hence can be applied to various systems. This protocol requires $3k-2$ states for each agent. This space complexity is asymptotically optimal because clearly any uniform k-partition protocol requires $\Omega(k)$ states to indicate a group of an agent. We evaluate the time complexity of the protocol by simulations. From the simulation results, we can observe that the time complexity increases exponentially with k but not exponentially with n.

1.2 Related Works

The population protocol model was introduced in [5, 7]. The class of computable predicates in this model was clarified by the researches.

In addition to such computability researches, many algorithmic problems have been considered in the population protocol model: leader election [1, 2, 8, 13, 15, 17, 21, 22, 23], counting [9, 11, 12, 18], and majority [1, 3, 6, 16]. The leader election problem has been studied for both designated and arbitrary initial states. For designated initial states, the main research topic is to minimize the time and space complexity of leader election protocols [1, 2, 15]. For arbitrary initial states, many researches have developed self-stabilizing and loosely-stabilizing protocols [8, 13, 17, 21, 22, 23]. The counting problem aims to count the number of agents in the population. After the first protocol was proposed in [12], the space complexity was gradually minimized [11, 18]. In [9], 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 [1, 3, 6, 16]. Recently an asymptotically space-optimal protocol for c colors ($c > 2$) has been proposed in [16].

As a similar problem to the uniform k-partition problem, Lamani et al. [19] studied a group composition 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 simultaneously and that all agents know n. Therefore the protocol does not work in our setting.

After publishing the conference version of this paper, some of the authors extended the result to the R-generalized partition problem, where the protocol divides all agents into k groups whose sizes follow a given ratio R [24].

2 Definitions

2.1 Population Protocol Model

A population A consists of 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 denoted by $(p, q) \to (p', q')$, which means that, when an agent in state p and an agent in state q interact, they update their states to p' and q', respectively. Transition $(p, q) \to (p', q')$ is asymmetric if both $p = q$ and $p' \neq q'$ hold; otherwise, the transition is symmetric. Protocol $P = (Q, \delta)$ is symmetric if every transition in δ is symmetric. Protocol $P = (Q, \delta)$ is deterministic if, for any pair of states $(p, q) \in Q \times Q$, at most one transition $(p, q) \to (p', q')$ exists in δ. We consider only deterministic symmetric protocols in this paper.

A global state of a population is called a configuration. A configuration is defined as a vector of (local) states of all agents. We describe $C \to C'$ if configuration C' is obtained from C by a single transition of a pair of agents. For configurations C and C', if there exists a sequence of configurations $C = C_0, C_1, \ldots, C_m = C'$ that satisfies $C_i \to C_{i+1}$ for any i ($0 \leq i < m$), we say C' is reachable from C, denoted by $C \xrightarrow{*} C'$. An infinite sequence of configurations $E = C_0, C_1, C_2, \ldots$ is called an execution of a protocol if $C_i \to C_{i+1}$ holds for any i ($i \geq 0$). 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. This implies that, under global fairness, if C occurs infinitely often, every configuration C^* reachable from C also occurs infinitely often.

In this paper, we assume that a protocol has designated initial states, that is, the state of every agent is a designated initial state $s_0 \in Q$ in the initial configuration. We denote by n the number of agents in a population. No agent knows n in the initial configuration. If $n = 2$ holds, two agents cannot transit to different states in symmetric protocols and thus cannot solve the uniform k-partition problem. Hence, we assume $n \geq 3$.

2.2 Uniform k-Partition Problem

Let $f : Q \to \{1, 2, \ldots, k\}$ be a function that maps a state of an agent to an integer $i (1 \leq i \leq k)$. Let $s(a)$ be a state of agent a. We say agent $a \in A$ belongs to the i-th group if $f(s(a)) = i$ holds.

Configuration C is stable if there is a partition $\{G_1, G_2, \ldots, G_k\}$ of A that satisfies the following condition:

1. $||G_i| - |G_j|| \leq 1$ for any i and j, and
2. For all C^* such that $C \xrightarrow{*} C^*$, each agent in G_i belongs to the i-th group at C^*.

An execution $E = C_0, C_1, C_2, \ldots$ solves the uniform k-partition problem if there is a stable configuration C_t in E. If each execution E of protocol P solves the uniform k-partition problem,

Algorithm 1 Uniform k-partition protocol

A state set

$Q = I \cup G \cup M \cup D$ where
$I = \{initial, initial'\}$,
$G = \{g_1, g_2, \ldots, g_k\}$,
$M = \{m_2, m_3, \ldots, m_{k-1}\}$, and
$D = \{d_1, d_2, \ldots, d_{k-2}\}$.

A mapping function to groups

$f(ini) = 1$ holds for any $ini \in I$.
$f(g_i) = i$ holds for any $g_i \in G$.
$f(m_i) = i$ holds for any $m_i \in M$.
$f(d_i) = 1$ holds for any $d_i \in D$.

Transition rules

1. $(initial, initial) \rightarrow (initial', initial')$
2. $(initial', initial') \rightarrow (initial, initial)$
3. $(d_i, ini) \rightarrow (d_i, \overline{ini})$ ($d_i \in D$ and $ini \in I$)
4. $(g_i, ini) \rightarrow (g_i, \overline{ini})$ ($g_i \in G$ and $ini \in I$)
5. $(initial, initial') \rightarrow (g_1, m_2)$
6. $(ini, m_i) \rightarrow (g_i, m_{i+1})$ ($ini \in I$ and $2 \leq i \leq k-2$)
7. $(ini, m_{k-1}) \rightarrow (g_{k-1}, g_k)$ ($ini \in I$)
8. $(m_i, m_j) \rightarrow (d_{i-1}, d_{j-1})(2 \leq i, j \leq k-1)$
9. $(d_i, g_i) \rightarrow (d_{i-1}, initial)(2 \leq i \leq k-2)$
10. $(d_1, g_1) \rightarrow (initial, initial)$

we say protocol P solves the uniform k-partition problem. The main objective of this paper is to minimize the number of states. When protocol P requires x states, we say P is a protocol with x states.

3 Uniform k-partition protocol

In this section, we propose a symmetric uniform k-partition protocol with designated initial states under global fairness. The summary of the protocol is given in Algorithm 1.

In this protocol, a set of agent states is divided into four subsets, i.e., $Q = I \cup G \cup M \cup D$, where $I = \{initial, initial'\}$, $G = \{g_1, g_2, \ldots, g_k\}$, $M = \{m_2, m_3, \ldots, m_{k-1}\}$, and $D = \{d_1, d_2, \ldots, d_{k-2}\}$. The designated initial state of agents is *initial*, that is, the state of every agent is *initial* in the initial configuration. State g_i in G indicates that the agent belongs to the i-th group, that is, $f(g_i) = i$ holds for any $g_i \in G$. For other state s, we define $f(s)$ as follows:

- $f(ini) = 1$ holds for any $ini \in I$.
- $f(d_i) = 1$ holds for any $d_i \in D$.
- $f(m_i) = i$ holds for any $m_i \in M$.

We say an agent is free if its state is in I. We define $\overline{initial} = initial'$ and $\overline{initial'} = initial$.

We will describe the details of the protocol in Sections 3.1 and 3.2. In the basic strategy (Section 3.1), the protocol makes k agents enter states $g_1, g_2, \ldots, g_k$ by using states in M as intermediate states. However, this strategy may increase the number of agents in some groups beyond n/k. In Section 3.2, we overcome such a situation by using states in D.

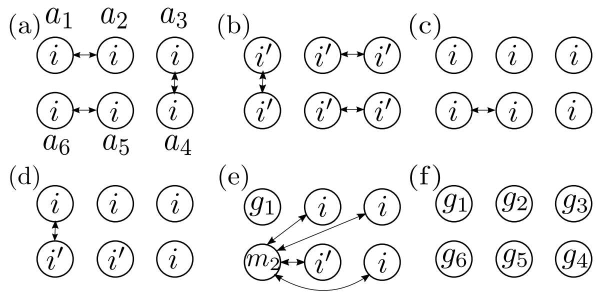


Fig. 1: An example of k-partition

3.1 Basic strategy

The basic strategy of the protocol is as follows: First two free agents transit to states g_1 and m_2. After that, for each i ($2 \le i \le k-2$), when an agent in state m_i and a free agent interact, they transit to states m_{i+1} and g_i, respectively. Lastly, when an agent in state m_{k-1} and a free agent interact, they transit to states g_k and g_{k-1}. By this behavior, k free agents can change their states to $g_1, g_2, \ldots, g_k$. That is, the size of each group is increased by one. To realize this, the protocol includes the following transitions.

1. $(initial, initial) \to (initial', initial')$
2. $(initial', initial') \to (initial, initial)$
3. $(d_i, ini) \to (d_i, \overline{ini})$ ($d_i \in D$ and $ini \in I$)
4. $(g_i, ini) \to (g_i, \overline{ini})$ ($g_i \in G$ and $ini \in I$)
5. $(initial, initial') \to (g_1, m_2)$
6. $(ini, m_i) \to (g_i, m_{i+1})$ ($ini \in I$ and $2 \le i \le k-2$)
7. $(ini, m_{k-1}) \to (g_{k-1}, g_k)$ ($ini \in I$)

First we explain transitions 1 to 5, which make two free agents transit to states g_1 and m_2. Recall that all agents are in state *initial* in the initial configuration. Since we consider symmetric protocols, two agents in state *initial* cannot transit to states g_1 and m_2 at one interaction. This is the reason why we introduce state *initial*$'$. Each agent in state *initial* (resp., *initial*$'$) transits to *initial*$'$ (resp., *initial*) when it interacts with an agent in a state in $I \cup D \cup G$ (except for interaction between one in state *initial* and one in state *initial*$'$). Transition 5 implies that, when agents in states *initial* and *initial*$'$ interact, they become g_1 and m_2, respectively. From global fairness, if at least two free agents and no agents in a state in M exist, two free agents eventually enter states *initial* and *initial*$'$, respectively, and then enter states g_1 and m_2 by an interaction. Transition 6 implies that, when a free agent and an agent in state m_i interact, they become g_i and m_{i+1}, respectively. By these transitions, free agents transit to states $g_1, \ldots, g_{k-2}$ one by one. After that, from transition 7, when a free agent and an agent in state m_{k-1} interact, they become g_{k-1} and g_k, respectively. From this behavior, the size of each group is increased by one.

Figure 1 is an example execution of the protocol for a population of six agents. Initially all agents are in state *initial* (Fig. 1 (a)). After interactions (a_1, a_2), (a_3, a_4), and (a_5, a_6), all agents enter state *initial*$'$ (Fig. 1 (b)). After interactions (a_1, a_6), (a_2, a_3), and (a_4, a_5), all agents enter state *initial* (Fig. 1 (c)). If such interactions happen infinitely, the protocol never solves the uniform k-partition problem. However, under the global fairness, such interactions do not occur infinitely. This is because, if some configuration C occurs infinitely often, every configuration reachable from C should occur. That is, eventually interactions (a_5, a_6) and (a_1, a_6) happen in this order from such a configuration (Fig. 1 (d) and (e)). Then, a_1 and a_6 enter states g_1 and m_2, respectively (Fig. 1

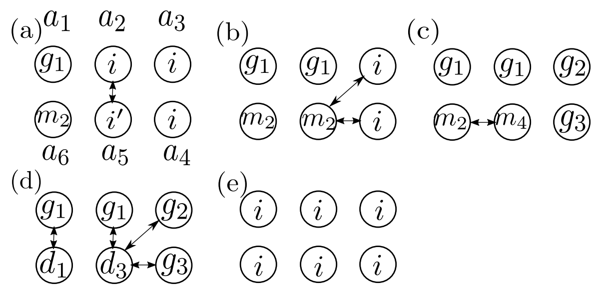


Fig. 2: Another example of k-partition

(e)). After that, if interactions (a_6, a_2), (a_6, a_3), (a_6, a_4), and (a_6, a_5) occur in this order, agent a_6 changes its state from m_2 to m_3, m_4, m_5, and g_6, and agents a_2, a_3, a_4, and a_5 enter g_2, g_3, g_4, and g_5, respectively (Fig. 1 (f)).

3.2 A problem of the basic strategy and its solution

However, in the protocol of the basic strategy, $\lceil n/k \rceil$ or more agents in state m_1 can appear. In this case, the above transitions do not achieve a uniform k-partition. For example, in the case of $n = 12$ and $k = 4$, if four agents enter state m_1, agents can transit to states g_1, g_2, m_3, g_1, g_2, m_3, g_1 ,g_2, m_3, g_1, g_2, m_3. To solve this problem, we introduce states in D and add the following transitions.

8. $(m_i, m_j) \rightarrow (d_{i-1}, d_{j-1})(2 \leq i, j \leq k-1)$
9. $(d_i, g_i) \rightarrow (d_{i-1}, \mathit{initial})(2 \leq i \leq k-2)$
10. $(d_1, g_1) \rightarrow (\mathit{initial}, \mathit{initial})$

By transition 8, when two agents in states in m_i and m_j interact, they transit to states in d_{i-1} and d_{j-1}, respectively. Intuitively, an agent in state d_i makes agents in $g_1, g_2, \ldots, g_i$ go back to state *initial*. Recall that an agent in state m_{i+1} can enter state d_i and an agent in state m_{i+1} has made agents in states $g_1, g_2, \ldots, g_i$. This means an agent in state d_i initializes agents that it makes enter states $g_1, g_2, \ldots, g_i$. More concretely, an agent in a state in D works as follows:

- For $2 \leq i \leq k-2$, when agents in states d_i and g_i interact, they become d_{i-1} and *initial* by transition 9, respectively.
- After that, from transition 10, when agents in states d_1 and g_1 interact, they become *initial*.

Figure 2 is an example that shows the impact of states in D. Similarly to Fig. 1, agents can transit to a configuration in Fig. 2 (a). If interactions (a_2, a_5), (a_3, a_5), and (a_4, a_5) occur in this order from Fig. 2 (a), agents transit to a configuration in Fig. 2 (c). In this configuration, transitions of the basic strategy (transitions 1 to 7) are not applied. However, transition 8 can be applied, that is, interaction (a_5, a_6) eventually occurs. By the interaction, a_5 and a_6 enter states d_3 and d_1, respectively (Fig. 2 (d)). After that, interactions (a_1, a_6), (a_4, a_5), (a_3, a_5) and (a_2, a_5) happen, and then all agents enter state *initial* (Fig. 2 (e)).

Clearly, agents can repeatedly enter state g_i and go back to *initial* many times. However, after an agent enters state g_k, one set of agents in states $g_1, \ldots, g_k$ never goes back to *initial*. Thus, if there are h agents in state g_k, the number of agents in state g_i is at least h for each i. In addition, when there are h agents in state g_k and $n - kh \geq k$ holds, there is an execution that makes some agent enter state g_k. This implies that, from the global fairness, some agent eventually enters state g_k. When $n - kh = r < k$ holds, there is an execution that makes the remaining agents transit to $g_1, g_2, \ldots, m_r$. From the global fairness, the remaining agents eventually enter these states. In this configuration, agents achieve a uniform k-partition and after that all agents never change their states.

4 Correctness

In this section, we prove the correctness of the proposed protocol. If $k = 2$, the protocol is exactly the same as a uniform bipartition protocol in [25]. Thus, the protocol solves the uniform k-partition problem for $k = 2$. In the rest of this section, we assume that $k \geq 3$ holds.

First, we define the notations to consider the number of states at a configuration. We denote by $\#ini$ the number of free agents (i.e., agents in states *initial* or *initial'*). We denote by $\#g_x$, $\#m_p$, and $\#d_q$ the numbers of agents in state g_x, m_p, and d_q, respectively ($1 \leq x \leq k$, $2 \leq p \leq k-1$, $1 \leq q \leq k-2$).

The first lemma gives invariants that hold for any configuration reachable from the initial configuration C_0. In the following, when configuration C is reachable from C_0, we simply say C is reachable.

Lemma **1.** *For any reachable configuration C, $\#g_x = \sum_{p=x+1}^{k-1} \#m_p + \sum_{q=x}^{k-2} \#d_q + \#g_k$ holds for any x ($1 \leq x \leq k$) at C.*

Proof. First we intuitively explain the invariants. Let us fix x. An agent in state m_p ($2 \leq p \leq k-1$) has made $p-1$ agents enter $g_1, g_2, \ldots, g_{p-1}$. Hence, for each agent in state m_p with $p > x$, there exists an agent in state g_x that corresponds to the agent. Consequently, there exist $\sum_{p=x+1}^{k-1} \#m_p$ agents in state g_x that correspond to agents in states in M. Since an agent in state d_q has changed its state from m_{q+1} to d_q, it has made q agents enter $g_1, g_2, \ldots, g_q$. Hence, for each agent in state d_q with $q \geq x$, there exists an agent in state g_x that corresponds to the agent. Consequently, there exist $\sum_{q=x}^{k-2} \#d_q$ agents in state g_x that correspond to agents in states in D. An agent in state g_k has made $k-1$ agents enter $g_1, g_2, \ldots, g_{k-1}$. Hence, there exist $\#g_k$ agents in state g_x that correspond to agents in state g_k. Therefore, we have the above invariants.

We prove the lemma formally by induction. First let us consider the initial configuration. Since $\#g_x = 0$, $\#m_p = 0$, and $\#d_q = 0$ hold for any x, p, and q ($1 \leq x \leq k$, $2 \leq p \leq k-1$, $1 \leq q \leq k-2$), the lemma holds.

Next, assume that the lemma holds at some configuration C. We show that, for any C' satisfying $C \to C'$, the lemma holds at C'. Clearly, if transition 1, 2, 3, or 4 occurs in $C \to C'$, the lemma holds at C' because $\#g_x$, $\#m_p$, and $\#d_q$ do not change for any x, p, and q ($1 \leq x \leq k$, $2 \leq p \leq k-1$, $1 \leq q \leq k-2$). Hence, we consider the remaining six transitions.

First, consider the case of transition 5. This transition increases $\#g_1$ and $\#m_2$ by one, and consequently it affects the formula of $x = 1$. Since the left and right sides of the formula increase by one, the lemma holds in this case.

Consider the case of transition 6. This transition increases $\#g_i$ and $\#m_{i+1}$ by one, and decreases $\#m_i$ by one. Consequently, it affects the formula of $x \leq i$. For $x < i$, since $\sum_{p=x+1}^{k-1} \#m_p$ and $\#g_x$ do not change, the left and right sides of the formula do not change. For $x = i$, both $\#g_i$ and $\sum_{p=i+1}^{k-1} \#m_p$ increase by one, the left and right sides of the formula increase by one. Hence, the lemma holds in this case.

Consider the case of transition 7. This transition increases $\#g_{k-1}$ and $\#g_k$ by one, and decreases $\#m_{k-1}$ by one. Consequently, it affects the formula of $x \leq k$. For $x < k-1$, since $\sum_{p=x+1}^{k-1} \#m_p$ decreases and $\#g_k$ increases by one, the left and right sides of the formula do not change. For $x = k-1$, both $\#g_k$ and $\#g_{k-1}$ increase by one and $\#m_{k-1}$ is not included in $\sum_{p=x+1}^{k-1} \#m_p$, the left and right sides of the formula increase by one. For $x = k$, the formula always holds. Hence, the lemma holds in this case.

Consider the case of transition 8. This transition increases $\#d_{i-1}$ and $\#d_{j-1}$ by one, and decreases $\#m_i$ and $\#m_j$ by one. Consequently, it affects the formula of $x \leq \max\{i, j\} - 1$. Since this transition increases $\sum_{q=x}^{k-2} \#d_q$ and decreases $\sum_{p=x+1}^{k-1} \#m_p$ by the same number for any $x \leq \max\{i, j\} - 1$, the lemma holds in this case.

Consider the case of transition 9. This transition increases $\#d_{i-1}$ by one, and decreases $\#d_i$ and $\#g_i$ by one. Consequently, it affects the formula of $x \leq i$. For $x < i-1$, since $\sum_{q=x}^{k-2} \#d_q$ and $\#g_x$ do not change, the left and right sides of the formula do not change. For $x = i$, both $\#g_i$ and $\sum_{q=x}^{k-2} \#d_q$ decrease by one, the left and right sides of the formula decrease by one. Hence, the lemma holds in this case.

Finally, consider the case of transition 10. Since this transition decreases $\#d_1$ and $\#g_1$ by one, it affects only formula of $x = 1$. Clearly, the left and right sides of the formula decrease by one. Hence, the lemma holds in this case. □

The invariants in Lemma 1 explain some properties of the proposed protocol. For example, $\#g_x \geq \#g_k$ holds for any x $(1 \leq x \leq k)$. This means the number of agents in each group is at least $\#g_k$. Since $\#g_k$ is never decreased from the protocol, the number of agents in each group is never decreased below $\#g_k$ after that. By Lemmas 2 to 4, we prove that $\#g_k$ eventually becomes $\lfloor n/k \rfloor$. That is, the number of agents in each group eventually becomes $\lfloor n/k \rfloor$.

Lemma **2.** *Let $\mathcal{C}_1$ be a set of all reachable configurations such that $\#ini \geq k$ holds. For any configuration C in $\mathcal{C}_1$, there exists C' such that $C \xrightarrow{*} C'$ holds and $\#g_k$ at C' is increased by one from C.*

Proof. If there exist no agents in state *initial* at C, there exist at least three agents in state *initial′* exist (because of $k \geq 3$). Consequently two of them enter state *initial* by interacting each other (transition 2). Similarly, if there exist no agents in state *initial′* at C, some agents can enter state *initial′* (transition 1). Hence, there exists a reachable configuration from C where at least one agent in state *initial′* and at least one agent in state *initial*. Let a_1 and a_2 be agents in state *initial′* and *initial*, respectively. After a_1 and a_2 interact, they become m_2 and g_1, respectively (transition 5). At this moment, there exist at least $k-2$ agents in state *initial* or *initial′*. After that, these $k-2$ agents can interact with a_2 one by one. As a result, these $k-2$ agents enter $g_2, g_3, \ldots, g_{k-1}$, and a_2 enters g_k (transitions 6 and 7). Therefore, $\#g_k$ is increased by one from C. □

Lemma **3.** *Let $\mathcal{C}_2$ be a set of all reachable configurations such that $\#ini < k$ and $n - k \cdot \#g_k \geq k$ hold. For any configuration C in $\mathcal{C}_2$, there exists C' such that $C \xrightarrow{*} C'$ holds and $\#g_k$ at C' is increased by one from C.*

Proof. We prove that, from C, there exists a transition such that 1) $\#g_k$ is increased by one or 2) $\#ini$ is increased. In the former case, the lemma directly holds. In the latter case, since $\#g_k$ is not increased, $n - k \cdot \#g_k \geq k$ still holds. Consequently, we can repeatedly apply this claim, and eventually $\#ini$ exceeds k or $\#g_k$ is increased by one. If $\#ini$ exceeds k, $\#g_k$ is eventually increased from Lemma 2. Therefore, the lemma holds.

To prove the above claim, we divide $\mathcal{C}_2$ into the following four sets of configurations $\mathcal{C}_d$, $\mathcal{C}_{m2}$, $\mathcal{C}_{m1}$, and $\mathcal{C}_{m0}$.

- $\mathcal{C}_d$ is a set of configurations (in $\mathcal{C}_2$) such that $\#d_q > 0$ holds for some q $(1 \leq q \leq k-2)$.
- $\mathcal{C}_{m2}$ is a set of configurations (in $\mathcal{C}_2$) such that $d_q = 0$ holds for any q $(1 \leq q \leq k-2)$ and $\sum_{p=2}^{k-1} \#m_p \geq 2$ holds.
- $\mathcal{C}_{m1}$ is a set of configurations (in $\mathcal{C}_2$) such that $d_q = 0$ holds for any q $(1 \leq q \leq k-2)$ and $\sum_{p=2}^{k-1} \#m_p = 1$ holds.
- $\mathcal{C}_{m0}$ is a set of configurations (in $\mathcal{C}_2$) such that $d_q = 0$ holds for any q $(1 \leq q \leq k-2)$ and $\sum_{p=2}^{k-1} \#m_p = 0$ holds.

First consider a configuration $C \in \mathcal{C}_d$. Let q be an integer such that $d_q > 0$ holds in C. From Lemma 1, $\#g_q > 0$ holds. Consequently, when agents in states d_q and g_q interact, at least one of them enters *initial* by transition 9 or 10. Thus, $\#ini$ is increased.

Next consider a configuration $C \in \mathcal{C}_{m2}$. From the definition of $\mathcal{C}_{m2}$, there exist two distinct agents a_i and a_j whose states are m_i and m_j, respectively. When a_i and a_j interact, they enter states d_{i-1} and d_{j-1} by transition 8, respectively. This configuration belongs to $\mathcal{C}_d$, and thus $\#ini$ is eventually increased.

Consider a configuration $C \in \mathcal{C}_{m1}$. Let i be an integer such that $\#m_i = 1$ holds. From Lemma 1, $\#g_x = 1 + \#g_k$ holds for $x \leq i-1$ and $\#g_x = \#g_k$ holds for $x \geq i$. Since a population consists of one agent in state m_i and agents in states $g_x (1 \leq x \leq k)$, *initial*, and *initial′*, we have

$\#ini = n - 1 - \sum_{x=1}^{k} \#g_x = n - k \cdot \#g_k - i \geq k - i$. Let a be the agent in state m_i and a_i, a_{i+1}, …, a_{k-1} be agents in state *initial* or *initial*$'$. If a interacts with a_i, a_{i+1}, …, a_{k-1} in this order, a_i, a_{i+1}, …, a_{k-1} transit to g_i, g_{i+1}, …, g_{k-1}, respectively and a transits to g_k. Thus, $\#g_k$ is increased by one.

Finally, consider a configuration $C \in \mathcal{C}_{m0}$. In this case, $\sum_{q=1}^{k} \#g_q + \#ini = n$ holds. From Lemma 1, $\#g_x = \#g_k$ holds for any x $(1 \leq x \leq k)$. That is, $\sum_{x=1}^{k} \#g_x + \#ini = k \cdot \#g_k + \#ini = n$ holds. Hence, $\#ini = n - k \cdot \#g_k \geq k$ holds. This means no configuration is in $\mathcal{C}_{m0}$.

Therefore, the lemma holds. □

Lemma **4.** *For any execution* $E = C_0, C_1, \ldots$, *there exists* C_t *such that* $n - k \cdot \#g_k < k$ *holds.*

Proof. First, we show that, when $n - k \cdot \#g_k \geq k$ holds at a configuration C_i, $\#g_k$ is increased by one at C_j for some j $(j > i)$. For contradiction, assume that such C_j does not exist. Since $\#g_k$ is never decreased from the protocol, $\#g_k$ is never changed and $n - k \cdot \#g_k \geq k$ continuously holds after C_i. Since the number of such configurations is finite, some configuration C'_i occurs infinitely often after C_i in E. From Lemmas 2 and 3, there exists C'_j such that $C'_i \xrightarrow{*} C'_j$ and $\#g_k$ in C'_j is increased by one from C'_i. That is, there exists a sequence of configurations $C'_1, C'_2, \ldots, C'_l$ such that $C'_i = C'_1 \rightarrow C'_2 \rightarrow \cdots \rightarrow C'_l = C'_j$ holds. From global fairness, since $C'_i = C'_1$ occurs infinitely often, C'_2 occurs infinitely often. Similarly, $C'_3, \ldots, C'_l = C'_j$ occur infinitely often. That is, $\#g_k$ at C'_j is increased by one from C_i. This is a contradiction. Thus, if $n - k \cdot \#g_k \geq k$ holds, $\#g_k$ is eventually increased by one. Therefore, the lemma holds. □

Note that, since $n \geq \sum_{x=1}^{k} \#g_x \geq k \cdot \#g_k$ holds from Lemma 1, $n - k \cdot \#g_k < k$ derives $\#g_k = \lfloor n/k \rfloor$. Hence, Lemma 4 implies that $\#g_k = \lfloor n/k \rfloor$ eventually holds. This implies that the number of agents in each group eventually becomes $\lfloor n/k \rfloor$ or $\lfloor n/k \rfloor + 1$ from Lemma 1. Let $r = n - k \cdot \lfloor n/k \rfloor$. If $r = 0$ holds, the uniform k-partition has been solved. If $r \geq 1$ holds, there exist r remaining agents. Lemma 5 shows resultant states of the remaining agents. If $r = 1$ holds, the one remaining agent is in state *initial* or *initial*$'$. If $r \geq 2$ holds, r agents enter states $g_1, g_2, \ldots, g_{r-1}$ and m_r.

Lemma **5.** *Assume that* $r = n - k \cdot \lfloor n/k \rfloor > 0$ *holds. Let* $\mathcal{C}_3$ *be a set of reachable configurations such that* $n - k \cdot \#g_k < k$ *holds (i.e.,* $\#g_k = \lfloor n/k \rfloor$*). For any configuration* $C \in \mathcal{C}_3$*, there exists* C' *such that 1)* $C \xrightarrow{*} C'$ *holds, 2)* $\#g_x = \lfloor n/k \rfloor + 1$ *holds for any* x $(1 \leq x \leq r - 1)$*, 3)* $\#g_x = \lfloor n/k \rfloor$ *holds for any* x $(r \leq x \leq k)$*, and 4)* $\#ini = 1$ *holds if* $r = 1$ *and* $\#m_r = 1$ *holds if* $r \geq 2$*.*

Proof. From Lemma 1, $\#g_x \geq \#g_k = \lfloor n/k \rfloor$ holds for any x $(1 \leq x \leq k)$ at C. Let $A' \subset A$ be a set of agents that include $\lfloor n/k \rfloor$ agents in state g_x at C for any x $(1 \leq x \leq k)$, and let $A_r = A - A'$.

Let us consider the case of $r = 1$. In this case A_r does not contain an agent in state m_p for any p because otherwise A_r also contains agents in state g_r $(r \leq p - 1)$ from Lemma 1. Similarly, A_r does not contain an agent in state d_q for any q. Hence, A_r contains one agent in state *initial* or *initial*$'$. Thus, if $r = 1$, the lemma holds.

In the following, we assume $r \geq 2$. Similarly to Lemma 3, we can prove that r agents in A_r transit to $g_1, g_2, \ldots, g_{r-1}$ and m_r. That is, we can easily observe the following facts. If all agents in A_r are in *initial* or *initial*$'$, they can transit to $g_1, g_2, \ldots, g_{r-1}$ and m_r by interacting one by one. If an agent in state d_q exists in A_r for some q, it eventually transits to *initial*. If two agents in states m_i and m_j exist in A_r for some i and j, they can transit to d_{i-1} and d_{j-1}. If A_r contains exactly one agent in state m_p for some p, A_r contains $p-1$ agents in states $g_1, g_2, \ldots, g_{p-1}$ and $r - p$ agents in states *initial* and *initial*$'$. In this case, agents in state m_p, *initial*, and *initial*$'$ can transit to $g_p, g_{p+1}, \ldots, g_{r-1}$ and m_r.

Since A' contains $\lfloor n/k \rfloor$ agents in state g_x for every x and A_r contains r agents in states $g_1, g_2, \ldots, g_{r-1}$ and m_r, the lemma holds. □

Lemma 5 proved that a configuration specified in the lemma is reachable from a configuration specified in Lemma 4. Thus, similarly to Lemma 4, we can obtain the following lemma.

Lemma **6.** *Assume that $r = n - k \cdot \lfloor n/k \rfloor > 0$ holds. For any execution $E = C_0, C_1, \ldots$, there exists C_t such that 1) $\#g_x = \lfloor n/k \rfloor + 1$ holds for any x $(1 \le x \le r-1)$, 2) $\#g_x = \lfloor n/k \rfloor$ holds for any x $(r \le x \le k)$, and 3) $\#ini = 1$ holds if $r = 1$ and $\#m_r = 1$ holds if $r \ge 2$.*

Let $r = n - k \cdot \lfloor n/k \rfloor$. From Lemmas 4 and 6, a population eventually reaches a configuration C^* such that 1) $\#g_x = \lfloor n/k \rfloor + 1$ holds for any x $(1 \le x \le r-1)$, 2) $\#g_x = \lfloor n/k \rfloor$ holds for any $x \ge r$, and 3) $\#ini = 1$ holds if $r = 1$ and $\#m_r = 1$ holds if $r \ge 2$. Since $f(g_x) = x$ holds for x $(1 \le x \le k)$, $f(m_p) = p$ holds for p $(2 \le p \le k-1)$, and $f(ini) = 1$ holds for $ini \in \{initial, initial'\}$, the number of agents in each group is $\lfloor n/k \rfloor$ or $\lfloor n/k \rfloor + 1$. In addition, no transition can happen at C^*. This implies that C^* is stable. Therefore, we have the following theorem.

Theorem **1.** *The proposed protocol solves the uniform k-partition problem. That is, there exists a symmetric protocol with $3k-2$ states and designated initial states that solves the uniform k-partition problem under global fairness.*

5 Simulation Results

In this section, we discuss the time complexity of the proposed protocol by simulations. We evaluate the time complexity by the total number of interactions until a population reaches a stable configuration. In the simulations, we construct an execution by selecting two agents uniformly at random in each configuration and making them interact. Note that, if we construct an infinite execution by this way, the execution satisfies global fairness with probability 1. For all simulation settings, we conduct a simulation 100 times and show the average values as the results.

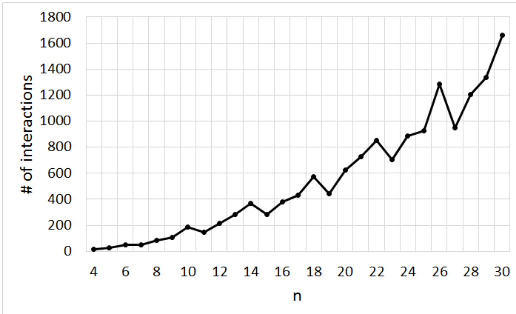
5.1 Varying the population size n

Figure 3 shows the number of interactions for $k \in \{4, 6, 8\}$ with changing the population size (i.e., the number of agents) n. As n increases, the number of interactions tends to increase. However, the number of interactions sometimes decreases when n increases. We can observe that such a phenomenon is repeated with a period of a length of k. That is, $n \bmod k$ influences the number of interactions.

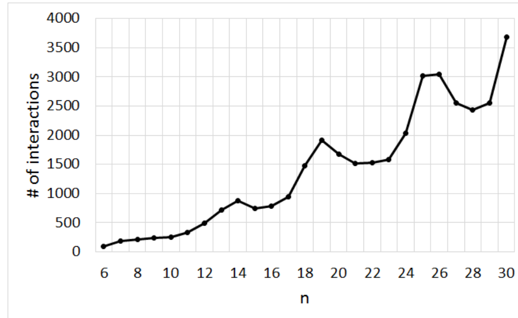
To observe the details of executions, we focus on the number of interactions required to construct one set of agents in states $g_1, g_2, \ldots, g_k$. We refer to this construction by grouping. Recall that, once an agent enters state g_k, the set of agents never goes back to *initial*. Let NI_i be the number of interactions required to construct the i-th set of agents in states $g_1, g_2, \ldots, g_k$. We define $NI_0 = 0$. We count $NI'_i = NI_i - NI_{i-1}$, i.e., the number of interactions to achieve the i-th grouping. We show the results in Figure 4. In this figure, we show NI'_1 at the bottom of the figure (denoted by 1st-grouping), NI'_2 at the second to the bottom (denoted by 2nd-grouping), and so on. Figure 4 shows that $NI'_1 < NI'_2 < \cdots$ holds except for the last part (i.e., transitions of the remaining $n \bmod k$ agents). This is because, as the execution proceeds, the number of agents not in a group decreases and consequently agents require more interactions to achieve the grouping. In addition, we can observe that, for any positive integer c, when $n = c \cdot k + 2, c \cdot k + 3, \ldots, c \cdot k + (k+1)$ holds, the number of interactions to achieve the $(c+1)$-th grouping (shown in the top of each graph) increases steeply with n. In addition, the number of interactions for the $(c+1)$-th grouping accounts for more than half of the total number of interactions for $n = c \cdot k + k$ and $n = c \cdot k + (k+1)$. These facts influence juggy forms of graphs in Figure 3.

Hereafter, to prevent the effect of $n \bmod k$, we execute simulations for the case that $n \bmod k = 0$ holds.

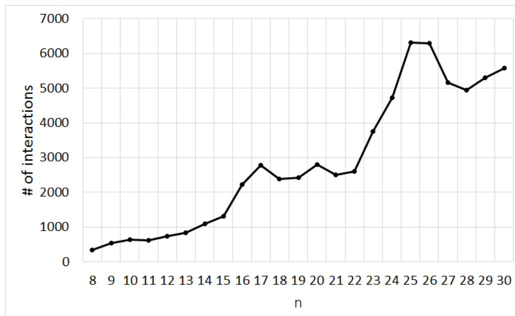
Figure 5 shows the number of interactions for $k \in \{3, 4, 5, 6\}$ with changing the population size n. We consider $n = 120 \cdot n'$ for $n' \in \{1, 2, \ldots, 8\}$ so that $n \bmod k = 0$ holds. Figure 5 shows that, as n increases, the number of interactions also increases. The number of interactions seems to increase more than linearly but less than exponentially with n.



(a) $k = 4$

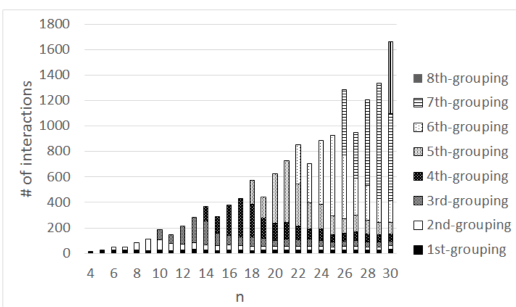


(b) $k = 6$

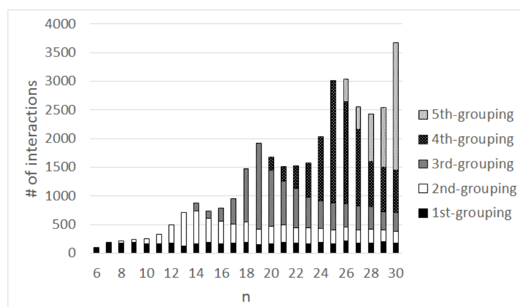


(c) $k = 8$

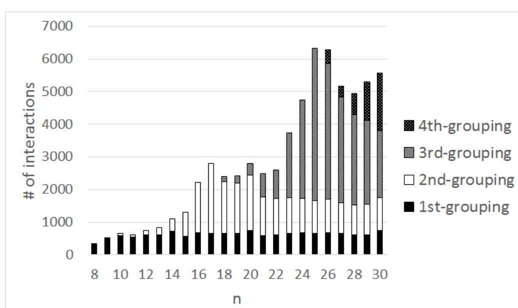
Fig. 3: The number of interactions for $k \in \{4, 6, 8\}$ with changing the population size n



(a) $k = 4$



(b) $k = 6$



(c) $k = 8$

Fig. 4: The number of interactions to achieve the i-th grouping

5.2 Varying the number of groups k

The logarithmic graph in Figure 6 shows the number of interactions for $n = 960$ with changing k. To avoid the effect of $n \bmod k$, we show the results only for the case that $n \bmod k = 0$ holds. Figure

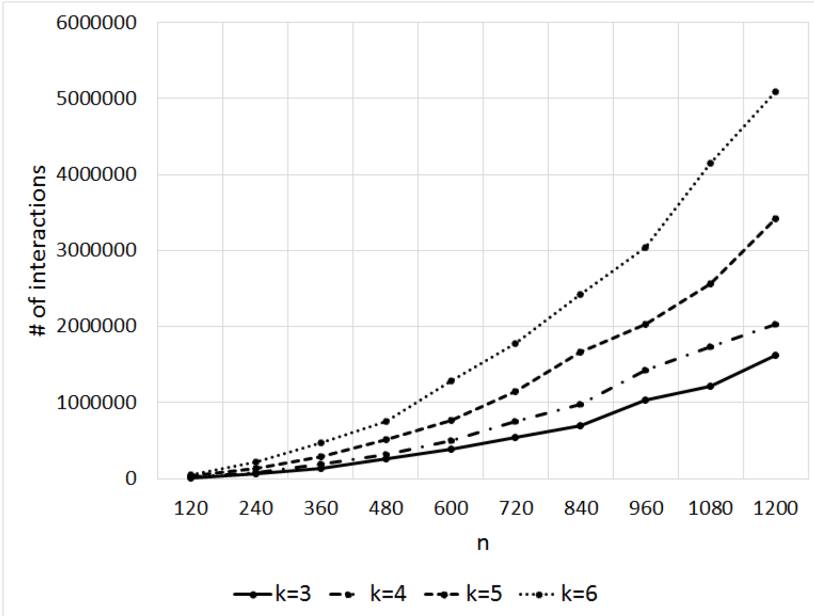


Fig. 5: The number of interactions for $k \in \{3, 4, 5, 6\}$ with changing the population size n

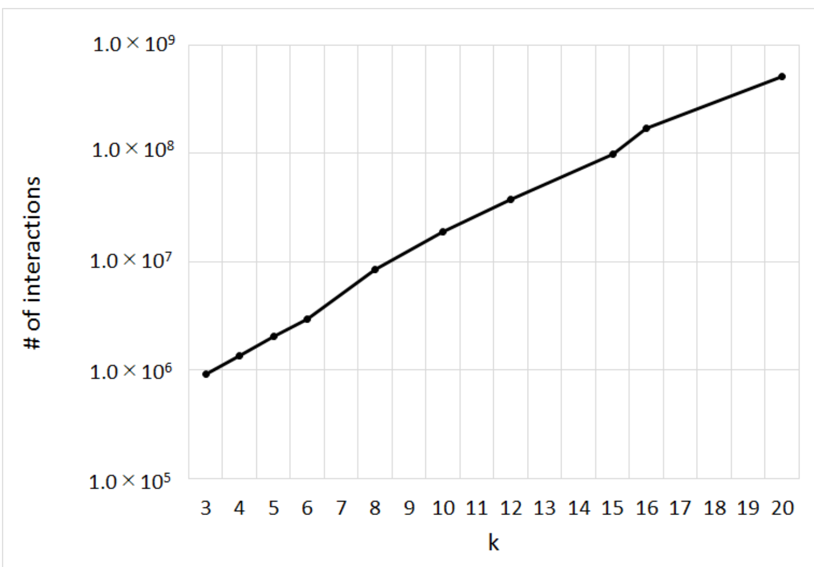


Fig. 6: The number of interactions for $n = 960$ with changing k

6 shows that the number of interactions seems to increase exponentially with k. This is because, to create a set of groups including agents with states g_1 to g_k, a m_2-state agent interacts $k-2$ free agents (i.e., agents with state *initial* or *initial*$'$) without interacting other m-state agents. Since interaction of *initial* and *initial*$'$ agents creates a m-state agent, a non-negligible number of m-state agents exist. Hence, the possibility that an agent interacts $k-2$ free agents without interacting m-state agents becomes exponentially small when k becomes large. This increases the number of interactions exponentially with k.

6 Conclusion

In this paper, we proposed a symmetric population protocol with $3k - 2$ states and designated initial states that solves the uniform k-partition problem under global fairness. Since $\Omega(k)$ states are necessary for any uniform k-partition protocol, the proposed protocol is asymptotically space-optimal. We evaluated the time complexity of the protocol by simulations. From the simulation results, we can observe that the time complexity increases exponentially with k but not exponentially with n. Some open questions are the following:

- What is the tight lower bound for space of the uniform k-partition protocol? Although our protocol is asymptotically space-optimal, it is important to develop a (non-asymptotically) space-optimal protocol for low-performance devices.
- What is the relation between the uniform k-partition problem and other problems such as counting, leader election, and majority?
- What is the time complexity of the uniform k-partition problem under probabilistic fairness? Is there a protocol such that the time complexity is polynomial of n and k?

Acknowledgements

This research was supported by Japan Science and Technology Agency (JST) SICORP and JSPS KAKENHI Grant Number 18K11167.

References

[1] Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L Rivest. Time-space trade-offs in population protocols. In *Proc. of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2560–2579, 2017.

[2] Dan Alistarh and Rati Gelashvili. Polylogarithmic-time leader election in population protocols. In *Proc. of the 42nd International Colloquium on Automata, Languages, and Programming*, pages 479–491, 2015.

[3] Dan Alistarh, Rati Gelashvili, and Milan Vojnović. Fast and exact majority in population protocols. In *Proc. of the 2015 ACM Symposium on Principles of Distributed Computing*, pages 47–56, 2015.

[4] Dana Angluin, James Aspnes, Melody Chan, Michael J Fischer, Hong Jiang, and René Peralta. Stably computable properties of network graphs. In *Proc. of International Conference on Distributed Computing in Sensor Systems*, pages 63–74, 2005.

[5] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4):235–253, 2006.

[6] Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, 2008.

[7] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.

[8] Dana Angluin, James Aspnes, Michael J Fischer, and Hong Jiang. Self-stabilizing population protocols. In *International Conference On Principles Of Distributed Systems*, pages 103–117. Springer, 2005.

[9] James Aspnes, Joffroy Beauquier, Janna Burman, and Devan Sohier. Time and space optimal counting in population protocols. In *Proc. of International Conference on Principles of Distributed Systems*, pages 13:1–13:17, 2016.

[10] James Aspnes and Eric Ruppert. An introduction to population protocols. In *Middleware for Network Eccentric and Mobile Applications*, pages 97–120, 2009.

[11] Joffroy Beauquier, Janna Burman, Simon Claviere, and Devan Sohier. Space-optimal counting in population protocols. In *Proc. of International Symposium on Distributed Computing*, pages 631–646, 2015.

[12] Joffroy Beauquier, Julien Clement, Stephane Messika, Laurent Rosaz, and Brigitte Rozoy. Self-stabilizing counting in mobile sensor networks with a base station. In *Proc. of International Symposium on Distributed Computing*, pages 63–76, 2007.

[13] Shukai Cai, Taisuke Izumi, and Koichi Wada. How to prove impossibility under global fairness: On space complexity of self-stabilizing leader election on a population protocol model. *Theory of Computing Systems*, 50(3):433–445, 2012.

[14] Carole Delporte-Gallet, Hugues Fauconnier, Rachid Guerraoui, and Eric Ruppert. When birds die: Making population protocols fault-tolerant. *Distributed Computing in Sensor Systems*, pages 51–66, 2006.

[15] David Doty and David Soloveichik. Stable leader election in population protocols requires linear time. In *Proc. of International Symposium on Distributed Computing*, pages 602–616, 2015.

[16] Leszek Gasieniec, David Hamilton, Russell Martin, Paul G Spirakis, and Grzegorz Stachowiak. Deterministic population protocols for exact majority and plurality. In *Proc. of International Conference on Principles of Distributed Systems*, pages 14:1–14:14, 2016.

[17] Taisuke Izumi. On space and time complexity of loosely-stabilizing leader election. In *Proc. of International Colloquium on Structural Information and Communication Complexity*, pages 299–312, 2015.

[18] Tomoko Izumi, Keigo Kinpara, Taisuke Izumi, and Koichi Wada. Space-efficient self-stabilizing counting population protocols on mobile sensor networks. *Theoretical Computer Science*, 552:99–108, 2014.

[19] Anissa Lamani and Masafumi Yamashita. Realization of periodic functions by self-stabilizing population protocols with synchronous handshakes. In *Proc. of International Conference on Theory and Practice of Natural Computing*, pages 21–33, 2016.

[20] Satoshi Murata, Akihiko Konagaya, Satoshi Kobayashi, Hirohide Saito, and Masami Hagiya. Molecular robotics: A new paradigm for artifacts. *New Generation Computing*, 31(1):27–45, 2013.

[21] Yuichi Sudo, Toshimitsu Masuzawa, Ajoy K Datta, and Lawrence L Larmore. The same speed timer in population protocols. In *Proc. of International Conference on Distributed Computing Systems*, pages 252–261, 2016.

[22] Yuichi Sudo, Junya Nakamura, Yukiko Yamauchi, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Loosely-stabilizing leader election in a population protocol model. *Theoretical Computer Science*, 444:100–112, 2012.

[23] Yuichi Sudo, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Loosely-stabilizing leader election on arbitrary graphs in population protocols without identifiers nor random numbers. In *Proc. of International Conference on Principles of Distributed Systems*, pages 14:1–14:16, 2015.

[24] Tomoki Umino, Naoki Kitamura, and Taisuke Izumi. Differentiation in population protocols. *6th workshop on biological distributed algorithms(BDA)*, 2018.

[25] Hiroto Yasumi, Fukuhito Ooshita, Ken'ichi Yamaguchi, and Michiko Inoue. Constant-space population protocols for uniform bipartition. *the 21st International Conference on Principles of Distributed Systems*, 2017.