Uniform Partition in Population Protocol Model Under Weak Fairness

Hiroto Yasumi
Nara Institute of Science and Technology, Ikoma city, Nara, Japan
yasumi.hiroto.yf9@is.naist.jp

Fukuhito Ooshita
Nara Institute of Science and Technology, Ikoma city, Nara, Japan
f-oosita@is.naist.jp

Michiko Inoue
Nara Institute of Science and Technology, Ikoma city, Nara, Japan
kounoe@is.naist.jp

Abstract

We focus on a uniform partition problem in a population protocol model. The uniform partition problem aims to divide a population into k groups of the same size, where k is a given positive integer. In the case of $k = 2$ (called uniform bipartition), a previous work clarified space complexity under various assumptions: 1) an initialized base station (BS) or no BS, 2) weak or global fairness, 3) designated or arbitrary initial states of agents, and 4) symmetric or asymmetric protocols, except for the setting that agents execute a protocol from arbitrary initial states under weak fairness in the model with an initialized base station. In this paper, we clarify the space complexity for this remaining setting. In this setting, we prove that P states are necessary and sufficient to realize asymmetric protocols, and that $P + 1$ states are necessary and sufficient to realize symmetric protocols, where P is the known upper bound of the number of agents. From these results and the previous work, we have clarified the solvability of the uniform bipartition for each combination of assumptions. Additionally, we newly consider an assumption on a model of a non-initialized BS and clarify solvability and space complexity in the assumption. Moreover, the results in this paper can be applied to the case that k is an arbitrary integer (called uniform k-partition).

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Concurrent algorithms

Keywords and phrases population protocol, uniform k-partition, distributed protocol

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2019.8

Related Version A full version of the paper is available at `http://arxiv.org/abs/1911.04678`

Funding This work was supported in part by JSPS KAKENHI Grant Number 18K11167 and JST SICORP Grant Number JPMJSC1806.

1 Introduction

1.1 The Background

A population protocol model [6, 9] is an abstract model for devices with heavily limited computation and communication capability. The devices are represented as passively moving agents, and a set of agents is called a population. In this model, if two agents approach, an interaction happens between them. At the time of the interaction, the two agents update their states. By repeating such interactions, agents proceed with computation. The population protocol model has many application examples such as sensor networks and molecular robot networks. For example, one may construct a network to investigate the ecosystem by attaching sensors to a flock of wild small animals such as birds. In this system,

Table 1 The minimum number of states to solve the uniform bipartition problem under global fairness.

BS	initial states of agents	symmetry	upper bound	lower bound	paper
initialized BS	designated	asymmetric	3	3	[32]
		symmetric	3	3	
	arbitrary	asymmetric	4	4	
		symmetric	4	4	
non-initialized BS	designated	asymmetric	3	3	
		symmetric	3	3	
	arbitrary	asymmetric	unsolvable		this paper
		symmetric	unsolvable		
no BS	designated	asymmetric	3	3	[32]
		symmetric	4	4	
	arbitrary	asymmetric	unsolvable		
		symmetric	unsolvable		

Table 2 The minimum number of states to solve the uniform bipartition problem under weak fairness.

BS	initial states of agents	symmetry	upper bound	lower bound	paper
initialized BS	designated	asymmetric	3	3	[32]
		symmetric	3	3	
	arbitrary	asymmetric	P	P	this paper
		symmetric	$P+1$	$P+1$	
non-initialized BS	designated	asymmetric	3	3	[32]
		symmetric	3	3	
	arbitrary	asymmetric	unsolvable		this paper
		symmetric	unsolvable		
no BS	designated	asymmetric	3	3	[32]
		symmetric	unsolvable		
	arbitrary	asymmetric	unsolvable		
		symmetric	unsolvable		

sensors exchange information with each other when two sensors approach sufficiently close. By repeating such information exchange, the system eventually grasps the entire environment of the flock. Another example is a system of molecular robots [26]. In this system, a large number of robots cooperate in a human body to achieve an objective (e.g. carrying medicine). To realize such systems, various fundamental protocols have been proposed in the population protocol model [11]. For example, there are leader election protocols [5, 14, 15, 17, 23, 28], counting protocols [10, 12, 13], majority protocols [7, 20], naming protocols [16], and so on.

In this paper, we study a uniform k-partition problem, which divides a population into k groups of the same size, where k is a given positive integer. The uniform k-partition problem has some applications. For example, we can save the battery by switching on only some groups. Another example is to execute multiple tasks by assigning different tasks to each group simultaneously. Protocols for the uniform k-partition problem can be used to attain fault-tolerance [18].

As a previous work, Yasumi et al. [32, 33] studied space complexity of uniform partition when the number of partitions is two (called uniform bipartition). In the paper, they considered four types of assumptions: 1) an initialized base station (BS) or no BS, 2) designated or arbitrary initial states of agents, 3) asymmetric or symmetric protocols, and 4)

Table 3 The minimum number of states to solve the uniform k-partition problem.

fairness	BS	initial states of agents	symmetry	upper bound	lower bound
weak fairness	single	arbitrary	asymmetric	P	P
			symmetric	$P+1$	$P+1$
global fairness	no	designated	symmetric	$3k-2$ [30]	k (truism)

global or weak fairness. A BS is a special agent that is distinguishable from other agents and has powerful capability. An initialized BS means that the BS has a designated initial state in the initial configuration. The BS enables us to construct efficient protocols, though it is sometimes difficult to implement. The assumption of initial states bear on the requirement of initialization and the fault-tolerant property. If a protocol requires designated initial states, it is necessary to initialize all agents to execute the protocol. Alternatively, if a protocol solves the problem with arbitrary initial states, we do not need to initialize agents other than the BS. This implies that, when agents transit to arbitrary states by transient faults, the protocol can reach the desired configuration by initializing the BS. Symmetry of protocols is related to the power of symmetry breaking in the population. Asymmetric protocols may include asymmetric transitions that make agents with the same states transit to different states. This needs a mechanism to break symmetry among agents and its implementation is not easy with heavily limited devices. Symmetric protocols do not include such asymmetric transitions. Fairness is an assumption of interaction patterns. Though weak fairness guarantees only that every pair of agents interact infinitely often, global fairness makes a stronger assumption on the order of interactions.

For most combinations of assumptions, Yasumi et al. [32] clarified the solvability of the uniform bipartition problem and the minimum number of states to solve the problem. Tables 1 and 2 show the solvability of the uniform bipartition. These tables show the number of states to solve the uniform bipartition problem under various assumptions, where P is the known upper bound of the number of agents. The remaining case for an initialized BS and no BS is a protocol with an initialized BS and arbitrary initial states under weak fairness. For this case, they proved only that $P-2$ states are necessary. In this paper, we will give tight lower and upper bounds of the number of states for this case. In addition, recently Burman et al. [16] have considered the case with a non-initialized BS, which is distinguished from other agents but has an arbitrary initial state, for a naming problem. Because Yasumi et al. [32] did not consider the case, we also consider the case in this paper.

For the general case of an arbitrary number of partitions, Yasumi et al. [32] proposed a symmetric protocol with no BS and designated initial sates under global fairness. The protocol uses $3k-2$ states for an agent to construct k groups of the same size. However, no protocol has been proposed for other combinations of assumptions.

1.2 Our Contributions

Our main contribution is to clarify the solvability of the uniform bipartition problem with arbitrary initial states under weak fairness in the model with an initialized BS. A previous work [32] proved only that $P-2$ states are necessary for each agent, where P is the known upper bound of the number of agents. In this paper, we improve the lower bound from $P-2$ states to P states for asymmetric protocols and from $P-2$ states to $P+1$ states for symmetric protocols. Additionally, we propose an asymmetric protocol with P states, and obtain a symmetric protocol with $P+1$ states by a scheme proposed in [12].

Another contribution is to clarify the solvability in case of a non-initialized BS for the uniform bipartition problem. For designated initial states, the protocol with an initialized BS, which is proposed in [32] can still work even if the BS is non-initialized. In this paper, we prove the impossibility with arbitrary initial states in case of non-initialized BS.

By combining these results with the previous work [32], we have clarified the tight upper and lower bounds on the number of states for an agent to solve the uniform bipartition problem for all combinations of assumptions (see Tables 1 and 2).

For the case of an initialized BS, arbitrary initial states, and weak fairness, it is interesting to compare these results with those of naming protocols [16]. A naming protocol aims to assign different states to all agents, and hence it can be regarded as a uniform P-partition protocol (the size of each group is zero or one). Burman et al. [16] prove that, to realize naming protocols in the same setting, P states are necessary and sufficient for asymmetric protocols and $P+1$ states are necessary and sufficient for symmetric protocols. That is, naming protocols have the same space complexity as uniform k-partition protocols. Clearly naming protocols (or uniform P-partition protocols) require P states to assign different states to P agents. Interestingly uniform bipartition protocols still require P states in this setting. On the other hand, the number of states is reduced to three or four when we assume designated initial states or global fairness.

Protocols proposed in this paper are available for the uniform k-partition problem, where k is a given integer. That is, P states and $P+1$ states are sufficient to realize asymmetric and symmetric protocols, respectively, to solve the uniform k-partition problem from arbitrary initial states under weak fairness in the model with an initialized BS. Since the uniform bipartition is a special case of the uniform k-partition, the lower bound for the uniform bipartition problem can be applied to the uniform k-partition problem. That is, P states and $P+1$ states are necessary to realize asymmetric and symmetric protocols, respectively, under the assumption. Therefore, we have clarified the tight upper and lower bounds of the number of states for the uniform k-partition problem under the assumption (see Table 3).

Due to space constraints, we have omitted some proofs. See [31] for the full version of this paper.

1.3 Related Works

The population protocol model was first introduced in [6, 8]. In those papers, the class of computable predicates in this model was studied. After that, many fundamental tasks have been studied such as leader election, counting, and majority. Those problems have been studied under various assumptions such as existence of a base station, fairness, symmetry of protocols, and initial states of agents. Many researchers have considered the leader election problem for both designated and arbitrary initial states. For designated initial states, leader election protocols have been studied intensively to minimize the time and space complexity [1, 3, 14, 15, 19, 21, 22, 27]. Alistarh et al. [3] proposed an algorithm that solves the problem in polylogarithmic stabilization time with polylogarithmic states. In [19], it was clarified that $\Omega(n)$ parallel time is necessary (i.e., $\Omega(n^2)$ interactions are necessary) to solve the problem with probability 1. After that, many researchers focused on solving the problem with high probability and shrink the time and space complexity [14, 15, 21, 22, 27]. On the other hand, for arbitrary initial states, self-stabilizing and loosely-stabilizing protocols are proposed [9, 17, 23, 28]. The counting problem, which aims to count the number of agents in the population, was introduced by [13]. After that, some researchers have studied the protocol to minimize the space complexity of the counting protocols [12, 24]. In [10], a time and space optimal protocol was proposed. The majority problem is also a fundamental problem that aims to decide majority of initial states. For the majority problem, many protocols have been studied [1, 2, 4, 7, 15, 20]. Although there are some difference in the model (existence of failure, deterministic or probabilistic solution, and so on), these works also aim to minimize the time and space complexity. Moreover, in recent years, Burman et al. [16] proposed a naming protocol which assigns a different state (called name) to each agent. In the paper, they completely clarify the solvability of the naming protocol under various assumptions.

The uniform k-partition problem and a similar problem have been considered in [16, 25, 29, 30]. Lamani et al. [25] studied a group composition problem, which aims to divide a population into groups of designated sizes. They assume that half of agents make interactions at the same time and that all agents know n. Therefore the protocol does not work in our setting. In [30], Yasumi et al. proposed a uniform k-partition protocol that requires $3k-2$ states without the BS under global fairness. Moreover, some of the authors extended the result of [30] to the R-generalized partition problem, where the protocol divides all agents into k groups whose sizes follow a given ratio R [29]. Since they assume designated initial states and global fairness, the protocol does not work in our setting. In addition, Delporte-Gallet et al. [18] proposed a protocol solving the k-partition problem with less uniformity. This protocol guarantees that each group includes at least $n/(2k)$ agents, where n is the number of agents. This protocol requires $k(k+3)/2$ states under global fairness.

2 Definitions

2.1 Population Protocol Model

A population is a collection of pairwise interacting agents, denoted by A. A protocol $\mathcal{P}(Q,\delta)$ consists of Q and δ, 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 with state p and an agent with state q interact, they transit their states to p' and q', respectively. Transition $(p,q)\to(p',q')$ is null if both $p=p'$ and $q=q'$ hold. We omit null transitions in descriptions of algorithms. Transition $(p,q)\to(p',q')$ is asymmetric if both $p=q$ and $p'\neq q'$ hold; otherwise, the transition is symmetric. Protocol $\mathcal{P}(Q,\delta)$ is symmetric if every transition in δ is symmetric. Protocol $\mathcal{P}(Q,\delta)$ is asymmetric if every transition in δ is symmetric or asymmetric. Protocol $\mathcal{P}(Q,\delta)$ is deterministic if, for any pair of states $(p,q)\in Q\times Q$, exactly one transition $(p,q)\to(p',q')$ exists in δ. We consider only deterministic protocols in this paper. A global state of a population is called a configuration, defined as a vector of (local) states of all agents. A state of agent a in configuration C, is denoted by $s(a,C)$. Moreover, when C is clear from the context, we simply denote $s(a)$. Transition of configurations is described in the form $C\to C'$, which means that 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 $C=C_0,C_1,\ldots,C_m=C'$ such that $C_i\to C_{i+1}$ holds for any i $(0\le i<m)$, 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 an execution of a protocol if $C_i\to C_{i+1}$ holds for any i $(i\ge 0)$. An execution E is weakly-fair if every pair of agents a and a' interacts infinitely often. An execution segment is a subsequence of an execution.

In this paper, we assume that a single BS exists in A. The BS is distinguishable from other non-BS agents, although non-BS agents 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 has unlimited resources, 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. For this reason, we say a protocol uses x states if $|Q_p|=x$ holds. Throughout the paper, we assume that non-BS agents have arbitrary initial states. On the other hand, as for the BS, we consider two cases, an initialized BS and a non-initialized BS. When we assume an initialized BS, the BS has a designated initial state while all non-BS agents have arbitrary initial states. When we assume a non-initialized BS, the BS also has an arbitrary initial state. 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). In the initial configuration, the BS and non-BS agents do not know the number of agents, but they know the upper bound P of the number of agents.

2.2 Uniform k-Partition Problem

Let A_p be a set of all non-BS agents. Let $f : Q_p \to \{color_1, color_2, \dots, color_k\}$ be a function that maps a state of a non-BS agent to $color_i (1 \le i \le k)$. We define a color of $a \in A_p$ as $f(s(a))$. We say agent $a \in A_p$ belongs to the i-th group if $f(s(a)) = color_i$ holds.

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

1. $||G_i| - |G_j|| \le 1$ 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^* (i.e., at C^*, any agent a in G_i satisfies $f(s(a)) = color_i$).

An execution $E = C_0, C_1, C_2, \dots$ solves the uniform k-partition problem if E includes a stable configuration C_t. If every weakly-fair execution E of protocol $\mathcal{P}$ solves the uniform k-partition problem, we say protocol $\mathcal{P}$ solves the uniform k-partition problem under weak fairness.

3 Impossibility Results for Initialized BS and Weak Fairness

In this section, we give impossibility results of asymmetric and symmetric protocols for the uniform bipartition problem (i.e., $k = 2$). Clearly these impossibility results can be applied to the uniform k-partition problem for $k > 2$. Recall that, for an initialized BS, we assume weak fairness and arbitrary initial states.

Since we consider the case of $k = 2$, function f is defined as $f : Q_p \to \{color_1, color_2\}$. In this section, we assign colors *red* and *blue* to $color_1$ and $color_2$, respectively, and we define f as function $f : Q_p \to \{red, blue\}$ that maps a state of a non-BS agent to *red* or *blue*. We say agent $a \in A_p$ is *red* (resp., *blue*) if $f(s(a)) = red$ (resp., $f(s(a)) = blue$) holds. We say s is a c-state if $f(s) = c$ holds. For $c \in \{red, blue\}$, we define c-agent as an agent that has a c-state. We define $\overline{red} = blue$ and $\overline{blue} = red$.

3.1 Common Properties of Asymmetric and Symmetric Protocols

First, we show basic properties that hold for both asymmetric and symmetric protocols. Let *Alg* be a protocol that solves the uniform bipartition. Recall that P is the known upper bound of the number of agents. Hence, *Alg* must solve the uniform bipartition when the actual number of agents is at most P. In the remainder of this subsection, we consider the case that the actual number of agents is $P - 2$.

Lemma 1 shows that, in any execution for $P - 2$ agents, eventually all agents continue to keep different states.

▶ **Lemma 1.** *In any weakly-fair execution* $E = C_0, C_1, C_2, \ldots$ *of* Alg *with* $P-2$ *agents and an initialized BS, there exists a configuration* C_h *such that 1)* C_h *is a stable configuration, and, 2) all agents have different states at* $C_{h'}$ *for any* $h' \geq h$.

Proof. (*Sketch*) For contradiction, we assume that there exist two agents with the same state s in a stable configuration of some execution E with $P-2$ agents. Next, consider an execution with P agents such that two additional agents have s as their initial states and other agents behave similarly to E. In the execution, two additional agents do not join the interactions until $P-2$ agents converge to a stable configuration in E. At that time, two of the $P-2$ agents have state s and additional two agents also have state s. We can prove that, from this configuration, $P-2$ agents cannot recognize the two additional agents and hence they make the same behavior as in E. In addition, the two additional agents can keep state s. Since the numbers of *red* and *blue* agents are balanced without the two additional agents and the two additional agents have the same state, the uniform bipartition problem cannot be solved. This is a contradiction. ◀

In the next lemma, we prove that there exists a configuration C such that, in any configuration reachable from C, all agents have different states. In addition, we also show that the system reaches C in some execution.

▶ **Definition 2.** *Configuration* C *is strongly-stable if 1)* C *is stable, and, 2) for any configuration* C' *with* $C \xrightarrow{*} C'$, *all agents have different states at* C'.

▶ **Lemma 3.** *When the number of agents other than the BS is* $P-2$, *there exists an execution of* Alg *that includes a strongly-stable configuration.*

Proof. (*Sketch*) For contradiction, we assume that such execution does not exist. First, consider a weakly-fair execution E of Alg. By Lemma 1, after some configuration C_t in E, all agents have different states. From the assumption, C_t is not strongly-stable. That is, there exists a configuration C_u reachable from C_t such that two agents have the same state. Hence, we can construct another weakly-fair execution E' of Alg such that E' is similar to E until C_t and C_u occurs after that. By Lemma 1, after some configuration $C_{t'}$ in E', all agents have different states. Observe that $C_{t'}$ occurs after C_u. From the assumption, there exists a configuration $C_{u'}$ reachable from $C_{t'}$ such that two agents have the same state. Hence, similarly to E', we can construct another weakly-fair execution E'' of Alg such that E'' is similar to E' until $C_{t'}$ and $C_{u'}$ occurs after that. By repeating this construction, we can construct a weakly-fair execution such that two agents have the same state infinitely often. From Lemma 1, this is a contradiction. ◀

3.2 Impossibility of Asymmetric Protocols

Here we show the impossibility of asymmetric protocols with $P-1$ states.

▶ **Theorem 4.** *In the model with an initialized BS, there is no asymmetric protocol that solves the uniform bipartition problem with* $P-1$ *states from arbitrary initial states under weak fairness, if* P *is an even integer.*

To prove the theorem by contradiction, we assume that such protocol Alg_{asym} exists. Let $Q_p = \{s_1, s_2, \ldots, s_{P-1}\}$ be a state set of agents other than the BS. Let $Q_{blue} = \{s \in Q_p \mid f(s) = blue\}$ be a set of blue states and $Q_{red} = \{s \in Q_p \mid f(s) = red\}$ be a set of red states. Without loss of generality, we assume that $|Q_{blue}| < |Q_{red}|$ holds. Recall that Lemmas 1 and 3 can be applied to both symmetric and asymmetric algorithms. Hence, the properties of the lemmas hold even in Alg_{asym}. In this proof, based on the properties, we construct an execution of P agents such that the BS does not recognize the difference from the execution of $P-2$ agents. We show contradiction by proving that this execution does not achieve uniform bipartition.

By Lemma 1, clearly Alg_{asym} requires $P/2-1$ *blue* states and $P/2-1$ *red* states. Consequently, we have the following two corollaries.

▶ **Corollary 5.** *$|Q_{blue}| = P/2-1$ and $|Q_{red}| = P/2$ hold.*

▶ **Corollary 6.** *For any weakly-fair execution of Alg_{asym} with $P-2$ agents and an initialized BS, any strongly-stable configuration includes all states in Q_{blue}.*

To prove the main theorem, we focus on the following weakly-fair execution of Alg_{asym} with $P-2$ agents.

▶ **Definition 7.** *Consider a population $A = \{a_0, a_1, \dots, a_{P-2}\}$ of $P-2$ agents and an initialized BS, where a_0 is the BS. We define $E_\alpha = C_0, C_1, C_2, \dots$ as a weakly-fair execution of Alg_{asym} for population A that satisfies the following conditions.*

- *E_α includes a strongly-stable configuration C_t, and,*
- *for any $u \geq 0$, agents that interact at $C_{t+2u} \to C_{t+2u+1}$ also interact at $C_{t+2u+1} \to C_{t+2(u+1)}$.*

Note that, in E_α, the system reaches a strongly-stable configuration C_t (this is possible from Lemma 3), and after C_t agents always repeat the same interaction twice.

▶ **Definition 8.** *We define Q_t as a set of states that appear after C_t in E_α.*

Note that, since C_t is strongly-stable, Q_t includes at least $P-2$ states. This implies that Q_t includes all states in Q_p or does not include one state in Q_p. From Corollary 6, $Q_{blue} \subset Q_t$ holds.

The following lemmas give key properties of Alg_{asym} to prove Theorem 4. We will present proofs of these lemmas later.

▶ **Lemma 9.** *For any distinct states p and q ($p \neq q$) such that $p \in Q_{blue}$ and $q \in Q_t$ hold, transition rule $(p, q) \to (p', q')$ satisfies the following conditions.*

- *If $q \in Q_{red}$ or $q \in Q_b$ (i.e., q is a state of the BS) holds, $p' = p$ holds.*
- *If $q \in Q_{blue}$ holds, either $(p', q') = (p, q)$ or $(p', q') = (q, p)$ holds.*

▶ **Lemma 10.** *There is a non-empty state set $Q^* \subseteq Q_{blue}$ that satisfies the following conditions.*

- *For any state $p \in Q^*$, transition rule $(p, p) \to (p', q')$ satisfies $p' \in Q^*$ and $q' \in Q^*$.*
- *Assume that, in a configuration C, there exists a subset of agents A^* such that all agents in A^* have states in Q^* and $|A^*| = |Q^*|+1$ holds. In this case, for any agent $a_r \in A^*$ and any state $q \in Q^*$, there exists an execution segment such that 1) the execution segment starts from C, 2) a_r has state q at the last configuration, 3) only agents in A^* join interactions, and 4) all agents in A^* have states in Q^* at the last configuration.*

Lemma 10 means that, if $|Q^*|+1$ agents have states in Q^*, we can make an arbitrary agent with a state in Q^* transit to an arbitrary state in Q^*. Using these lemmas, we show the theorem by constructing a weakly-fair execution of Alg_{asym} with P agents that cannot be distinguished from execution E_α.

Proof of Theorem 4

Consider a population $A' = \{a'_0, \ldots, a'_P\}$ of P agents and an initialized BS, where a'_0 is the BS. Let C'_0 be an initial configuration such that initial states of $a'_0, \ldots, a'_P$ are $s(a_0, C_0), \ldots, s(a_{P-2}, C_0)$, s^*, s^*, where s^* is a state in Q^*.

For A' we construct an execution $E_\beta = C'_0, C'_1, \ldots, C'_t, \ldots$ using execution E_α as follows.

- For $0 \le u \le t-1$, when a_i and a_j interact at $C_u \to C_{u+1}$ in E_α, a'_i and a'_j interact at $C'_u \to C'_{u+1}$ in E_β.

Clearly, $s(a'_i, C'_t) = s(a_i, C_t)$ holds for any i $(0 \le i \le P-2)$. Since $s(a'_{P-1}, C'_t) = s(a'_P, C'_t) = s^*$ holds, the difference in the numbers of *red* and *blue* agents remains two and consequently C'_t is not a stable configuration.

To construct the remainder of E_β, first let us consider the characteristics of C'_t. Let $A_q \subseteq A$ be a set of agents that have states in Q^* at C_t, and let $\bar{A}_q = A - A_q$. Since all agents have different states and all states in Q_{blue} appear in C_t by Corollary 6, we have $|A_q| = |Q^*|$ from $Q^* \subseteq Q_{blue}$. Let $A'_q \subseteq A'$ be a set of agents that have states in Q^* at C'_t, and let $\bar{A}'_q = A' - A'_q$. Note that, for $i \le P-2$, $a_i \in A_q$ holds if and only if $a'_i \in A'_q$ holds. Since a'_{P-1} and a'_P are also in A'_q, we have $|A'_q| = |Q^*| + 2$. In the following, we construct the remainder of execution E_β that includes infinitely many configurations similar to E_α. we define similarity of configurations in E_β and E_α as follows.

▶ **Definition 11.** *We say configuration C'_u $(u \ge t)$ in E_β is similar to C_v $(v \ge t)$ in E_α if the following conditions hold:*
- *For any agent $a_i \in A_q$, $s(a_i, C_v) \in Q^*$ holds.*
- *For any agent $a'_i \in A'_q$, $s(a'_i, C'_u) \in Q^*$ holds.*
- *For any agent $a'_i \in \bar{A}'_q$ (i.e., $a_i \in \bar{A}_q$), $s(a'_i, C'_u) = s(a_i, C_v)$ holds.*

Let us focus on an execution segment $e = C_{t+2u}, C_{t+2u+1}, C_{t+2(u+1)}$ of E_α for any $u \ge 0$, and consider a configuration C'_x of A' such that C'_x is similar to C_{t+2u}. From now, we explain the way to construct an execution segment $e' = C'_x, \ldots, C'_y$ of E_β that guarantees that C'_y is similar to $C_{t+2(u+1)}$. Since C'_t is similar to C_t, we can repeatedly apply this construction and construct an infinite execution E_β. As a result, for any $u \ge 0$, E_β includes a configuration C' that is similar to C_{t+2u}. Since C' includes $P-1$ *red* agents and $P+1$ *blue* agents, E_β does not include a stable configuration. Note that E_β is not necessarily weakly-fair, but later we explain the way to construct a weakly-fair execution from E_β.

Consider configuration C'_x that is similar to C_{t+2u}. Assume that, in E_α, agents a_i and a_j interact in $C_{t+2u} \to C_{t+2u+1}$. Recall that a_i and a_j also interact in $C_{t+2u+1} \to C_{t+2(u+1)}$. We construct execution segment e' as follows:

- Case that $a_i \in A_q \wedge a_j \in A_q$ holds. Since $s(a_i, C_{t+2u}) \in Q^* \subseteq Q_{blue}$ and $s(a_j, C_{t+2u}) \in Q^* \subseteq Q_{blue}$ hold, $s(a_i, C_{t+2(u+1)}) \in Q^*$ and $s(a_j, C_{t+2(u+1)}) \in Q^*$ also hold from Lemma 10 (the first condition) and Lemma 9. Since other agents do not change their states, C'_x is similar to $C_{t+2(u+1)}$. Hence, in this case, we consider that the constructed execution segment e' is empty.
- Case that either $a_i \in A_q \wedge a_j \in \bar{A}_q$ or $a_i \in \bar{A}_q \wedge a_j \in A_q$ holds. Without loss of generality, we assume that $a_i \in A_q \wedge a_j \in \bar{A}_q$ holds. In this case, $s(a'_i, C'_x) \in Q^*$ is not necessarily equal to $\alpha = s(a_i, C_{t+2u}) \in Q^*$. Hence, in the execution segment e', we first make some agent $a'_r \in A'_q$ enter state α by interactions among agents in A'_q. By Lemma 10 (the second condition) and $|A'_q| = |Q^*| + 2$, such interactions exist and all agents in A'_q have states in Q^* after the interactions. Let C'_z be the resultant configuration.

Clearly C'_z is similar to C_{t+2u} and $s(a'_r, C'_z) = s(a_i, C_{t+2u}) \wedge s(a'_j, C'_z) = s(a_j, C_{t+2u})$ holds. After that, a'_r and a'_j interact twice. We regard the resultant configuration as C'_y (i.e., the last configuration of the constructed execution segment e'). Clearly both $s(a'_r, C'_y) = s(a_i, C_{t+2(u+1)})$ and $s(a'_j, C'_y) = s(a_j, C_{t+2(u+1)})$ hold. Since C'_z is similar to C_{t+2u} and $s(a'_j, C'_y) = s(a_j, C_{t+2(u+1)})$, it is sufficient to prove $s(a'_r, C'_y) \in Q^*$ to guarantee that C'_y is similar to $C_{t+2(u+1)}$. Observe that $s(a_j, C_{t+2u}) \notin Q^*$. This is because, since C_{t+2u} is strongly-stable, all agents have different states and agents in A_q occupy all states in Q^* (the first condition of Definition 11). Hence, $s(a'_r, C'_z) = s(a_i, C_{t+2u}) \in Q^*$ is not equal to $s(a'_j, C'_z) = s(a_j, C_{t+2u}) \notin Q^*$. Consequently, from $s(a'_r, C'_z) \in Q^* \subseteq Q_{blue}$, $s(a'_r, C'_y) = s(a'_r, C'_z) \in Q^*$ holds by Lemma 9. Therefore, C'_y is similar to $C_{t+2(u+1)}$.

- Case that $a_i \in \bar{A}_q \wedge a_j \in \bar{A}_q$ holds. In this case, since $s(a'_i, C'_x) = s(a_i, C_{t+2u})$ and $s(a'_j, C'_x) = s(a_j, C_{t+2u})$ hold, a'_i and a'_j simply interact twice consecutively. We regard the resultant configuration as C'_y (i.e., the last configuration of the constructed execution segment e'). Clearly, since a'_i and a'_j change their states similarly to a_i and a_j, C'_y is similar to $C_{t+2(u+1)}$.

Now we have constructed infinite execution E_β, but E_β is not necessarily weakly-fair. In the following, we construct a weakly-fair execution E_γ of population A' by slightly modifying E_β. To guarantee that E_γ is weakly-fair, for any pair of agents (a'_i, a'_j), a'_i and a'_j should interact infinite number of times in E_γ. For pair of agents (a'_i, a'_j) with $a'_i \in \bar{A}'_q$ and $a'_j \in \bar{A}'_q$, a'_i and a'_j interact infinite number of times in E_β because E_α is weakly-fair and a'_i interacts with a'_j in E_β when a_i interacts with a_j in E_α. For pair of agents (a'_i, a'_j) with $a'_i \in A'_q$ and $a'_j \in A'_q$, we can arbitrarily add interactions of them because, by Lemma 10 (the first condition) and Lemma 9 (the second condition), they keep their states in Q^* and consequently do not influence similarity of configurations.

Hence, we consider the remaining pair (a'_i, a'_j), that is, either $a'_i \in A'_q \wedge a'_j \in \bar{A}'_q$ or $a'_i \in \bar{A}'_q \wedge a'_j \in A'_q$ holds. Without loss of generality, we assume that a'_i is in A'_q and a'_j is in $\bar{A}'_q$. Since E_α is weakly-fair, a_j interacts with an agent in A_q infinite number of times in E_α. Recall that these interactions correspond to interactions of a'_j and a'_r in E_β, and a'_r can be arbitrarily selected from A'_q. For this reason, we can choose a'_r in a round-robin manner so that a'_j interacts with any agent in A'_q infinite number of times. For example, when a_j and an agent in A_q first interact (after C_t), we choose an agent in A'_q as a'_r, and then in the next interaction of a_j and an agent in A_q we can choose another agent in A'_q as a'_r. By this construction, a'_j can interact with any agent a'_i in A'_q infinite number of times.

From this way, we can construct a weakly-fair execution E_γ similarly to E_β. However, for any $u \geq 0$, E_γ includes a configuration C'' that is similar to C_{t+2u}. Since C'' includes $P-1$ *red* agents and $P+1$ *blue* agents, E_γ does not include a stable configuration. This is a contradiction.

The Proof Sketch of Lemma 9

Consider the case that transition $(p, q) \to (p', q')$ occurs at a strongly-stable configuration with $P-2$ agents. By Corollary 6, since any strongly-stable configuration includes all states in Q_{blue}, $(p, q) \to (p', q')$ can occur at the configuration.

First, consider the case that $q \in Q_{red}$ or $q \in Q_b$ holds. For contradiction, assume that $p' \neq p$ holds. By Corollary 6, since an agent with p' exists in the strongly-stable configuration, two agents with p' exist after transition $(p, q) \to (p', q')$. By the definition of strongly-stable configuration, this is a contradiction.

Next, consider the case that $q \in Q_{blue}$ holds. For contradiction, assume that $(p', q') \neq (p, q)$ and $(p', q') \neq (q, p)$ hold. By the definition of stable configuration, p' and q' are *blue*.

Hence, by Corollary 6, since an agent with any state in Q_{blue} exists in the strongly-stable configuration, two agents with the same state in Q_{blue} exist after transition $(p, q) \to (p', q')$. By the definition of strongly-stable configuration, this is a contradiction.

The Proof Sketch of Lemma 10

First, to show the proof sketch, we give some definitions.

▶ **Definition 12.** *For states q and q', we say $q \rightsquigarrow q'$ if there exists a sequence of states $q = q_0, q_1, \cdots, q_k = q'$ such that, for any $i (0 \le i < k)$, transition rule $(q_i, q_i) \to (q_{i+1}, x_i)$ or $(q_i, q_i) \to (x_i, q_{i+1})$ exists for some x_i.*

▶ **Definition 13.** *For states q and q', we say $q \overset{*}{\rightsquigarrow} q'$ if $x \rightsquigarrow q'$ holds for any x such that $q \rightsquigarrow x$ holds.*

Note that, in these definitions, we consider only interactions of agents with the same state. We say two agents are homonyms if they have the same state. Intuitively, $q \rightsquigarrow q'$ means that an agent with state q can transit to q' by only interactions with homonyms. Also, $q \overset{*}{\rightsquigarrow} q'$ means that, even if an agent with state q transits to any state x by interactions with homonyms, it can still transit from x to q' by interactions with homonyms.

Let $Q_{p^*} = \{q \mid p^* \overset{*}{\rightsquigarrow} q\}$. In this proof, we show that Q_{p^*} satisfies the conditions of Q^* of Lemma 10. Clearly, if homonyms with states in Q_{p*} interact, they transit to states in Q_{p*}. This implies that Q_{p^*} satisfies the first condition of Q^* of the lemma. To prove the second condition, we first show that, when $|Q_{p*}|$ agents have states in Q_{p*} initially, for any $s \in Q_{p*}$, there exists an execution such that only homonyms in the $|Q_{p*}|$ agents interact and eventually some agent transits to state s. To show this, we define a potential function $\Phi(C, s)$ for configuration C and state $s \in Q_{p*}$. Intuitively, $\Phi(C, s)$ represents how far configuration C is from a configuration that includes an agent with state s. To define $\Phi(C, s)$, we define $DtQ(s_i, s)$ as follows.

▶ **Definition 14.** *$DtQ(s_i, s)$ is a function that satisfies the following property.*
- *If $s_i = s$ holds, $DtQ(s_i, s) = 0$ holds.*
- *If $s_i \neq s$ and $s_i \in Q_{p*}$ holds, $DtQ(s_i, s) = \min\{DtQ(s_j^1, s), DtQ(s_j^2, s)\} + 1$ holds when transition rule $(s_i, s_i) \to (s_j^1, s_j^2)$ exists.*
- *If $s_i \notin Q_{p^*}$ holds, $DtQ(s_i, s) = \infty$ holds.*

Intuitively, $DtQ(s_i, s)$ gives the minimum number of interactions to transit from state s_i to state s when allowing only interactions with homonyms. Note that, for any $s_i \in Q_{p*}$, s_i can transit to s because $s_i \rightsquigarrow p \rightsquigarrow s$ holds.

▶ **Definition 15.** *Consider configuration C such that $z = |Q_{p^*}|$ agents $a_1, \ldots, a_z$ have states in Q_{p^*}. In this case, we define potential function $\Phi(C, s)$ as a multi set $\{DtQ(s(a_1, C), s)$, $DtQ(s(a_2, C), s)$, $DtQ(s(a_3, C), s)$, $\ldots$, $DtQ(s(a_z, C), s)\}$.*

▶ **Definition 16.** *For distinct $\Phi(C_1, s)$ and $\Phi(C_2, s)$, we define a comparative operator of them as follows: Let i be the minimum integer such that the number of i-elements is different in $\Phi(C_1, s)$ and $\Phi(C_2, s)$. If the number of i-elements in $\Phi(C_1, s)$ is smaller than $\Phi(C_2, s)$, we say $\Phi(C_1, s) < \Phi(C_2, s)$.*

From now, we show that there exists an execution such that some agent transits to s. Let C be a configuration with $|Q_{p^*}|$ agents such that all agents have states in Q_{p*} and there does not exist an agent with s in C. Since $|Q_{p^*}|$ agents have states in Q_{p*} in C and there does not exist an agent with s in C, there exist homonyms in C. When homonyms with a state in Q_{p*} interact, they transit to states in Q_{p*}. These imply that, when homonyms interact at $C \to C'$, either an agent with s or homonyms with a state in Q_{p*} exist in C'. Thus, for contradiction, assume that there exists an infinite execution segment $e = C_0, C_1, C_2, \ldots$ with $|Q_{p^*}|$ agents such that only homonyms interact and any agent never has s in e, where C_0 is a configuration such that all agents have states in Q_{p*}. For e, $\Phi(C_0, s) > \Phi(C_1, s) > \Phi(C_2, s) > \cdots$ holds. This is because, since any $p \in Q_{p^*}$ satisfies $p \rightsquigarrow p^* \rightsquigarrow s$, $DtQ(s(a, C_i), s) > DtQ(s(a, C_{i+1}), s)$ holds for at least one agent a that interacts at $C_i \to C_{i+1}$. Hence, eventually some agent has s in e. By the definition of e, this is a contradiction.

From now, we prove the second condition of Lemma 10. Let A^* be a set of agents such that $|A^*| = |Q_{p^*}| + 1$, and assume that all agents in A^* have states in Q_{p^*}. The existence of the above execution implies that, for any agent $a_r \in A^*$, we can make some agent in $A^* - \{a_r\}$ transits to state $s(a_r)$ by interactions among $A^* - \{a_r\}$. Then, we can make an interaction with homonyms between a_r and an agent with $s(a_r)$. After that, since a_r has a state in Q_{p^*}, all agents in A^* keep states in Q_{p^*}. Hence, in the same way, by making interaction repeatedly between a_r and an agent with $s(a_r)$, a_r can transit to any $q \in Q_{p^*}$ because any $p \in Q_{p^*}$ satisfies $p \rightsquigarrow p^* \rightsquigarrow q$. Therefore, Q_{p^*} satisfies the second condition and thus the lemma holds.

3.3 Impossibility of Symmetric Protocols

In this subsection, we show the impossibility of symmetric protocols with P states. To prove this impossibility, we use ideas of the impossibility proof for the naming protocol [16]. This work shows that, in the model with an initialized BS, there is no symmetric naming protocol with P states from arbitrary initial states under weak fairness. We apply the proof of [16] to the uniform bipartition but, since the treated problem is different, we need to make a non-trivial modification.

▶ **Theorem 17.** *In the model with an initialized BS, there is no symmetric protocol that solves the uniform bipartition problem with P states from arbitrary initial states under weak fairness, if P is an even integer.*

In the case of naming protocols [16], the impossibility proof proves that a unique state (called sink state) always exists. However, in the case of uniform bipartition protocols, sometimes no sink state exists. To treat this situation, we additionally define a sink pair, which is a pair of two states that has a similar property of a sink state. We show that either a sink state or a sink pair exists, and, we prove that there is no symmetric protocol in both cases.

4 Possibility Results for Initialized BS and Weak Fairness

In this section, we propose both asymmetric and symmetric protocols for the uniform k-partition problem. The asymmetric protocol requires P states and the symmetric protocol requires $P + 1$ states. By impossibility results, these protocols are space-optimal.

4.1 An Asymmetric Protocol

In this subsection, we show a P-state asymmetric protocol for the uniform k-partition problem. The idea of the protocol is to assign states 0, 1, $\ldots$, $n - 1$ to n agents one by one and then regard an agent with state s as a member of the $(s \mod k)$-th group. One may

Algorithm 1 Asymmetric uniform k-partition protocol.

A variable at BS

M: The state that the BS assigns next, initialized to 0

A variable at a mobile agent *a*:

$S_a \in \{0, 1, 2, \ldots, P-1\}$: The agent state, initialized arbitrarily. Agent a belongs to the ($S_a \mod k$)-th group.

while a mobile agent a interacts with BS **do**
  if $M \leq S_a$ **then**
    $S_a = M$
    $M = M + 1$
  end if
end while
while two mobile agent a and b interact **do**
  if $S_a = S_b$ and $S_a < P - 1$ **then**
    $S_a = S_a + 1$
  end if
end while

think that, to implement this idea, we can directly use a naming protocol [16], where the naming protocol assigns different states to agents by using P states if $n \leq P$ holds. Actually, if $n = P$ holds, the naming protocol assigns states 0, 1, …, $P-1$ to P agents one by one and hence it achieves uniform k-partition. However, if $n < P$ holds, the naming protocol does not always achieve uniform k-partition. For example, in the case of $(n-1)k < P$, the naming protocol may assign states 0, k, $2k$, …, $(n-1)k$ to n agents one by one, which implies that all agents are in the 0-th group.

Algorithm 1 shows a P-state asymmetric protocol for the uniform k-partition problem. In the protocol, the BS assigns states 0, 1, …, $n-1$ to n agents one by one. To do this, the BS maintains variable M, which represents the state the BS will assign next. The BS sets $M = 0$ initially, and increments M whenever it assigns M to an agent. Consider an interaction between the BS and an agent with state x. If x is smaller than M, the BS judges that it has already assigned a state to the agent, and hence it does not update the state. If x is M or larger, the BS assigns state M to the agent and increments M. When the BS assigns state x to an agent, there may exist another agent with state x because of arbitrary initial states. To treat this case, when two agents with the same state x interact, one transits to state $x+1$ and the other keeps its state x. By repeating such interactions, eventually exactly one agent has state x. By this behavior, the BS eventually assigns states 0, 1, …, $n-1$ to n agents one by one, and hence the algorithm achieves uniform k-partition.

As a result, we obtain the following theorem.

▶ **Theorem 18.** *Algorithm 1 solves the uniform k-partition problem. This means that, in the model with an initialized BS, there exists an asymmetric protocol with P states and arbitrary initial states that solves the uniform k-partition problem under weak fairness, where P is the known upper bound of the number of agents.*

▶ Remark 19. Interestingly, when P is odd, Algorithm 1 solves the uniform bipartition even if the number of agent states is $P-1$. Concretely, let $S_a \in \{1, 2, 3, \ldots, P-1\}$ be a set of agent states, and initialize variable M to 1. Then, Algorithm 1 converges to a configuration such that there exist two agents with state $P-1$ (and other states are held by exactly one agent). This is because, in the algorithm, the BS assigns $P-1$ agents to $P-1$ states one by one, and, since the algorithm works under weak fairness, the remaining one agent shifts its state until state $P-1$. In the configuration, the difference in the numbers of *red* and *blue* agents is one. Moreover, every agent does not change its own state after the configuration. Hence, the uniform bipartition is solved. ◀

4.2 A Symmetric Protocol

In this subsection, we propose a $(P+1)$-state symmetric protocol for the uniform k-partition problem. We can easily obtain the protocol by a scheme proposed in [12]. In [12], a P-state symmetric protocol for the counting problem is proposed. The counting protocol assigns different states in $\{1,\ldots,n\}$ to n agents and keeps the configuration if $n < P$ holds. Hence, by regarding $P+1$ as the upper bound of the number of agents and allowing $P+1$ states, the protocol assigns different states in $\{1,...,n\}$ to n agents for any $n \le P$. This implies that, as in the previous subsection, the protocol can achieve the uniform k-partition by regarding an agent with x as a member of the $(x \mod k)$-th group.

▶ **Theorem 20.** *In the model with an initialized BS, there exists a symmetric protocol with $P+1$ states and arbitrary initial states that solves the uniform k-partition problem under weak fairness, where P is the known upper bound of the number of agents.*

5 Results for Non-initialized BS

In this section, we show the impossibility with non-initialized BS. In the proof, we use ideas of the impossibility proof for the uniform bipartition protocol [32]. This work shows that, in the model with no BS, there is no protocol for uniform bipartition problem with arbitrary initial states under global fairness.

▶ **Theorem 21.** *In the model with non-initialized BS, no protocol with arbitrary initial states solves the uniform bipartition problem under global fairness.*

Proof. For contradiction, we assume such a protocol *Alg* exists. Moreover, we assume n is even and at least 4. We consider the following two cases.

First, consider population $A = \{a_0,\ldots,a_n\}$ of n agents and a non-initialized BS, where a_0 is the BS. For A, consider an execution $E = C_0, C_1, \cdots$ 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. By the definition of a stale configuration, the color of agent a_i (i.e.,$f(s(a_i))$) never changes for any a_i $(1 \le i \le n)$ after C_t even if agents interact in any order.

Next, consider population $A' = \{a'_0\} \cup \{a'_i | f(s(a_i, C_t)) = red\}$, where a'_0 is the BS. For A', consider an execution $E' = C'_0, C'_1, \cdots$ of *Alg* from the initial configuration C'_0 such that $s(a'_i, C'_0) = s(a_i, C_t)$ holds for any $a'_i \in A'$. Note that, since we assume a non-initialized BS, even the BS can have $s(a_0, C_t)$ as its initial state. Since all agents are *red* at C'_0, some agents must change their colors to reach a stable configuration. This implies that, after C_t in execution E, agents change their colors if they interact similarly to E'. This is a contradiction. ◀

6 Conclusion

In this paper, we clarify solvability of the uniform bipartition with arbitrary initial states under weak fairness in the model with an initialized BS. Concretely, for asymmetric protocols, we show that P states are necessary and sufficient to solve the uniform k-partition problem under the assumption, where P is the known upper bound of the number of agents. For symmetric protocols, we show that $P+1$ states are necessary and sufficient under the assumption. Moreover, these upper and lower bounds can be applied to the k-partition problem under the assumption. There are some open problems as follows:

- Are there some relations 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?

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, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2221–2239. SIAM, 2018.

3 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.

4 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.

5 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.

6 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.

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

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

9 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.

10 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.

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

12 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.

13 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.

14 Petra Berenbrink, Dominik Kaaser, Peter Kling, and Lena Otterbach. Simple and efficient leader election. In *1st Symposium on Simplicity in Algorithms (SOSA 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

15 Andreas Bilke, Colin Cooper, Robert Elsaesser, and Tomasz Radzik. Population protocols for leader election and exact majority with O (logˆ 2 n) states and O (logˆ 2 n) convergence time. *arXiv preprint arXiv:1705.01146*, 2017.

16 Janna Burman, Joffroy Beauquier, and Devan Sohier. Brief Announcement: Space-Optimal Naming in Population Protocols. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 479–481. ACM, 2018.

17 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.

18 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.

19 David Doty and David Soloveichik. Stable leader election in population protocols requires linear time. *Distributed Computing*, 31(4):257–271, 2018.

20 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.

21 Leszek Gasieniec, Grzegorz Stachowiak, and Przemyslaw Uznanski. Almost logarithmic-time space optimal leader election in population protocols. *arXiv preprint arXiv:1802.06867*, 2018.

22 Leszek Gasieniec and Grzegorz Staehowiak. Fast space optimal leader election in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2653–2667. SIAM, 2018.

23 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.

24 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.

25 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.

26 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.

27 Yuichi Sudo, Fukuhito Ooshita, Taisuke Izumi, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Logarithmic Expected-Time Leader Election in Population Protocol Model. *arXiv preprint arXiv:1812.11309*, 2018.

28 Yuichi Sudo, Fukuhito Ooshita, Hirotsugu Kakugawa, Toshimitsu Masuzawa, Ajoy K Datta, and Lawrence L Larmore. Loosely-Stabilizing Leader Election with Polylogarithmic Convergence Time. In *22nd International Conference on Principles of Distributed Systems (OPODIS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

29 Tomoki Umino, Naoki Kitamura, and Taisuke Izumi. Differentiation in Population Protocols. *6th workshop on biological distributed algorithms(BDA)*, 2018.

30 Hiroto Yasumi, Naoki Kitamura, Fukuhito Ooshita, Taisuke Izumi, and Michiko Inoue. A Population Protocol for Uniform k-partition under Global Fairness. *International Journal of Networking and Computing*, 9(1):97–110, 2019.

31 Hiroto Yasumi, Fukuhito Ooshita, and Michiko Inoue. Uniform Partition in Population Protocol Model under Weak Fairness. *arXiv preprint arXiv:1911.04678*, 2019.

32 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.

33 Hiroto Yasumi, Fukuhito Ooshita, Ken'ichi Yamaguchi, and Michiko Inoue. Space-optimal population protocols for uniform bipartition under global fairness. *IEICE TRANSACTIONS on Information and Systems*, 102(3):454–463, 2019.