Uniform Bipartition in the Population Protocol Model with Arbitrary Communication Graphs

Hiroto Yasumi\textsuperscript{1}, Fukuhito Ooshita\textsuperscript{1}, Michiko Inoue\textsuperscript{1}, and Sébastien Tixeuil\textsuperscript{2}

\textsuperscript{1}Nara Institute of Science and Technology
\textsuperscript{2}Sorbonne Université

Abstract

In this paper, we focus on the uniform bipartition problem in the population protocol model. This problem aims to divide a population into two groups of equal size. In particular, we consider the problem in the context of arbitrary communication graphs. As a result, we clarify the solvability of the uniform bipartition problem with arbitrary communication graphs when agents in the population have designated initial states, under various assumptions such as the existence of a base station, symmetry of the protocol, and fairness of the execution. When the problem is solvable, we present protocols for uniform bipartition. When global fairness is assumed, the space complexity of our solutions is tight.

1 Introduction

1.1 Background

In this paper, we consider the population protocol model introduced by Angluin et al. \cite{5}. The population protocol model is an abstract model for low-performance devices. In the population protocol model, devices are represented as anonymous agents, and a population is represented as a set of agents. Those agents move passively (i.e., they cannot control their movements), and when two agents approach, they are able to communicate and update their states (this pairwise communication is called an interaction). A computation then consists of an infinite sequence of interactions.

Application domains for population protocols include sensor networks used to monitor live animals (each sensor is attached to a small animal and monitors e.g. its body temperature) that move unpredictably (hence, each sensor must handle passive mobility patterns). Another application domain is that of molecular robot networks \cite{28}. In such systems, a large number of molecular robots collectively work inside a human body to achieve goals such as transport of medicine. Since those robots are tiny, their movement is uncontrollable, and robots may only maintain extremely small memory.

In the population protocol model, many researchers have studied various fundamental problems such as leader election protocols \cite{4} (A population protocol solves leader election if starting from an initially uniform population of agents, eventually a single agent outputs leader, while all others output non-leader), counting \cite{8, 10, 11} (The counting problem consists in counting how many agents participate to the protocol; As the agents’ memory is typically constant, this number is output by a special agent that may maintain logarithmic size memory, the base station), majority \cite{6} (The majority problem aims to decide which, if any, initial state in a population is a majority),
In this paper, we focus on the uniform bipartition problem \cite{33,34,35}, whose goal is to divide a population into two stable groups of equal size (the difference is one if the population size is odd). To guarantee the stability of the group, each agent eventually belongs to a single group and never changes the group after that. Applications of the uniform bipartition include saving batteries in a sensor network by switching on only one group, or executing two tasks simultaneously by assigning one task to each group. Contrary to previous work that considered complete communication graphs \cite{33,34,35}, we consider the uniform bipartition problem over arbitrary graphs. In the population protocol model, most existing works consider the complete communication graph model (every pairwise interaction is feasible). However, realistic networks command studying incomplete communications graphs (where only a subset of pairwise interactions remains feasible) as low-performance devices and unpredictable movements may not yield a complete set of interactions. Moreover, in this paper, we assume the designated initial states (i.e., all agents share the same given initial state), and consider the problem under various assumptions such as the existence of a base station, symmetry of the protocol, and fairness of the execution. Although protocols with arbitrary initial states tolerate a transient fault, protocols with designated initial states can usually be designed using fewer states, and exhibit faster convergence times. Actually, it was shown in \cite{34} that, with arbitrary initial states, constant-space protocols cannot be constructed in most cases even assuming complete graphs.

### 1.2 Related Works

The population protocol model was proposed by Angluin et al. \cite{5}, who were recently awarded the 2020 Edsger W. Dijkstra prize in Distributed Computing for their work. While the core of the initial study was dedicated to the computability of the model, subsequent works considered various problems (e.g., leader election, counting, majority, uniform k-partition) under different assumptions (e.g., existence of a base station, fairness, symmetry of protocols, and initial states of agents).

The leader election problem was studied from the perspective of time and space efficiency. Doty and Soloveichik \cite{20} proved that $\Omega(n)$ expected parallel time is required to solve leader election with probability 1 if agents have a constant number of states. Relaxing the number of states to a polylogarithmic value, Alistarh and Gelashvili \cite{3} proposed a leader election protocol in polylogarithmic expected stabilization time. Then, Gąsieniec et al. \cite{23} designed a protocol with $O(\log \log n)$ states and $O(\log n \cdot \log \log n)$ expected time. Furthermore, the protocol of Gąsieniec et al. \cite{23} is space-optimal for solving the problem in polylogarithmic time. In \cite{29}, Sudo et al. presented a leader election protocol with $O(\log n)$ states and $O(\log n)$ expected time. This protocol is time-optimal for solving the problem. Finally, Berenbrink et al. \cite{15} proposed a time and space optimal protocol that solves the leader election problem with $O(\log \log n)$ states and $O(\log n)$ expected time. In the case of arbitrary communication graphs, it turns out that self-stabilizing leader election is impossible \cite{7} (a protocol is self-stabilizing if its correctness does not depend on its initial global state). This impossibility can be avoided if oracles are available \cite{9,18} or if the self-stabilization requirement is relaxed: Sudo et al. \cite{30} proposed a loosely stabilizing protocol for leader election (loose stabilization relates to the fact that correctness is only guaranteed for a very long expected amount of time).

The counting problem was introduced by Beauquier et al. \cite{11} and popularized the concept of a base station. Space complexity was further reduced by follow-up works \cite{10,24}, until Aspnes et al. \cite{8} finally proposed a time and space optimal protocol. On the other hand, by allowing the initialization of agents, the counting protocols without the base station were proposed for both exact counting \cite{16} and approximate counting \cite{11,16}. In \cite{11}, Alistarh et al. proposed a protocol...
Table 1: The minimum number of states to solve the uniform bipartition problem with designated initial states over complete graphs [33, 34].

| base station                      | fairness | symmetry | upper bound | lower bound |
|-----------------------------------|----------|----------|-------------|-------------|
| initialized/non-initialized base station | global   | asymmetric | 3           | 3           |
|                                   |          | symmetric  | 3           | 3           |
| weak                              |          | asymmetric | 3           | 3           |
|                                   |          | symmetric  | 3           | 3           |
| no base station                   | global   | asymmetric | 3           | 3           |
|                                   |          | symmetric  | 4           | 4           |
| weak                              |          | asymmetric | 3           | 3           |
|                                   |          | symmetric  | unsolvable  |             |

that computes an integer \( k \) such that \( \frac{1}{2} \log n < k < 9 \log n \) in \( O(\log n) \) time with high probability using \( O(\log n) \) states. After that, Berenbrink et al. [16] designed a protocol that outputs either \( \lfloor \log n \rfloor \) or \( \lceil \log n \rceil \) in \( O(\log^2 n) \) time with high probability using \( O(\log n \cdot \log \log n) \) states. Moreover, in [16], they proposed the exact counting protocol that computes \( n \) in \( O(\log n) \) time using \( \tilde{O}(n) \) states with high probability.

The majority problem was addressed under different assumptions (e.g., with or without failures [6], deterministic [22, 25] or probabilistic [2, 12, 13, 25] solutions, with arbitrary communication graphs [27], etc.). Those works also consider minimizing the time and space complexity. Recently, Berenbrink et al. [14] shows trade-offs between time and space for the problem.

To our knowledge, the uniform \( k \)-partition problem and its variants have only been considered in complete communication graphs. Lamani et al. [26] studied a group decomposition problem that aims to divide a population into groups of designated sizes. Yasumi et al. [32] proposed a uniform \( k \)-partition protocol with no base station. Umino et al. [31] extended the result to the \( R \)-generalized partition problem that aims at dividing a population into \( k \) groups whose sizes follow a given ratio \( R \). Also, Delporte-Gallet et al. [19] proposed a \( k \)-partition protocol with relaxed uniformity constraints: the population is divided into \( k \) groups such that in any group, at least \( n/(2k) \) agents exist, where \( n \) is the number of agents.

Most related to our work is the uniform bipartition solution for complete communication graphs provided by Yasumi et al. [33, 34]. For the uniform bipartition problem over complete graphs with designated initial states, Yasumi et al. [33, 34] studied space complexity under various assumptions such as: (i) an initialized base station, a non-initialized base station, or no base station (an initialized base station has a designated initial state, while a non-initialized has an arbitrary initial state), (ii) asymmetric or symmetric protocols (asymmetric protocols allow interactions between two agents with the same state to map to two resulting different states, while symmetric protocols do not allow such a behavior), and (iii) global or weak fairness (weak fairness guarantees that every individual pairwise interaction occurs infinitely often, while global fairness guarantees that every recurrently reachable configuration is eventually reached). Furthermore, they also study the solvability of the uniform bipartition problem with arbitrary initial states. Table 1 shows the upper and lower bounds of the number of states to solve the uniform bipartition with designated initial states over complete communication graphs.

There exist some protocol transformers that transform protocols for some assumptions into ones for other assumptions. In [3], Angluin et al. proposed a transformer that transforms a protocol with complete communication graphs into a protocol with arbitrary communication graphs. This transformer requires the quadruple state space and works under global fairness. In this transformer, agents exchange their states even after convergence. For the uniform bipartition problem, since
agents must keep their groups after convergence, they cannot exchange their states among different groups and thus the transformer proposed in [5] cannot directly apply to the uniform bipartition problem. Bournez et al. [17] proposed a transformer that transforms an asymmetric protocol into symmetric protocol by assuming additional states. In [17], only protocols with complete communication graphs were considered and the transformer works under global fairness. We use the same idea to construct a symmetric uniform bipartition protocol under global fairness without a base station.

### 1.3 Our Contributions

In this paper, we study the solvability of the uniform bipartition problem with designated initial states over arbitrary graphs. A summary of our results is presented in Table 2. Let us first observe that, as complete communication graphs are a special case of arbitrary communication graphs, the impossibility results by Yasumi et al. [34] remain valid in our setting. With a base station (be it initialized or non-initialized) under global fairness, we extend the three states protocol by Yasumi et al. [34] from complete communication graphs to arbitrary communication graphs. With a non-initialized base station under weak fairness, we propose a new symmetric protocol with $3P + 1$ states, where $P$ is a known upper bound of the number of agents. These results yield identical upper bounds for the easier cases of asymmetric protocols and/or initialized base station. In addition, we also show a condition of communication graphs in which the number of states in the protocol can be reduced from $3P + 1$ to constant. Concretely, we show that the number of states in the protocol can be reduced to $3l + 1$ if we assume communication graphs such that every cycle either includes the base station or its length is not a multiple of $l$, where $l$ is a positive integer greater than three.

On the other hand, with no base station under global fairness, we prove that four and five states are necessary and sufficient to solve uniform bipartition with asymmetric and symmetric protocols, respectively. In the same setting, in complete graphs, three and four states were necessary and sufficient. So, one additional state enables problem solvability in arbitrary communications graphs in this setting. With no base station under weak fairness, we prove that the problem cannot be solved, using a similar argument as in the impossibility result for leader election by Fischer and Jiang [21]. Overall, we show the solvability of uniform bipartition in a variety of settings for a population of agents with designated initial states assuming arbitrary communication graphs. In cases where the problem remains feasible, we provide upper and lower bounds with respect to the
number of states each agent maintains, and in all cases where global fairness can be assumed, our bounds are tight.

2 Definitions

2.1 Population Protocol Model

A population whose communication graph is arbitrary is represented by an undirected connected graph $G = (V, E)$, where $V$ is a set of agents, and $E \subseteq V \times V$ is a set of edges that represent the possibility of an interaction between two agents. That is, two agents $u \in V$ and $v \in V$ can interact only if $(u, v) \in E$ holds. A protocol $\mathcal{P} = (Q, \delta)$ consists of $Q$ and $\delta$, where $Q$ is a set of possible states for agents, and $\delta$ is a set of transitions from $Q \times Q$ to $Q \times Q$. Each transition in $\delta$ is denoted by $(p, q) \rightarrow (p', q')$, which means that, when an interaction between an agent $x$ in state $p$ and an agent $y$ in state $q$ occurs, their states become $p'$ and $q'$, respectively. Moreover, we say $x$ is an initiator and $y$ is a responder. When $x$ and $y$ interact as an initiator and a responder, respectively, we simply say that $x$ interacts with $y$. Transition $(p, q) \rightarrow (p', q')$ is null if both $p = p'$ and $q = q'$ hold. We omit null transitions in the descriptions of protocols. Protocol $\mathcal{P} = (Q, \delta)$ is symmetric if, for every transition $(p, q) \rightarrow (p', q')$ in $\delta$, $(q, p) \rightarrow (q', p')$ exists in $\delta$. In particular, if a protocol $\mathcal{P} = (Q, \delta)$ is symmetric and transition $(p, p) \rightarrow (p', q')$ exists in $\delta$, $p' = q'$ holds. If a protocol is not symmetric, the protocol is asymmetric. Protocol $\mathcal{P} = (Q, \delta)$ is deterministic if, for any pair of states $(p, q) \in Q \times Q$, exactly one transition $(p, q) \rightarrow (p', q')$ exists in $\delta$. 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 use $s(a)$ to denote the state of agent $a$. A transition between two configurations $C$ and $C'$ is described as $C \rightarrow C'$, and means that configuration $C'$ is obtained from $C$ by a single interaction between two agents. For two configurations $C$ and $C'$, if there exists a sequence of configurations $C = C_0, C_1, \ldots, C_m = C'$ such that $C_i \rightarrow C_{i+1}$ holds for every $i$ ($0 \leq i < m$), we say $C'$ is reachable from $C$, denoted by $C \Rightarrow C'$. An infinite sequence of configurations $\Xi = C_0, C_1, C_2, \ldots$ is an execution of a protocol if $C_i \rightarrow C_{i+1}$ holds for every $i$ ($i \geq 0$).

An execution $\Xi$ is weakly-fair if, for every pair of agents $(v, v') \in E$, $v$ (resp. $v'$) interacts with $v'$ (resp., $v$) infinitely often. An execution $\Xi$ is globally fair if, for every pair of configurations $C$ and $C'$ such that $C \rightarrow C'$, $C'$ occurs infinitely often when $C$ occurs infinitely often. Intuitively, global fairness guarantees that, if configuration $C$ occurs infinitely often, then every possible interaction in $C$ also occurs infinitely often. Then, if $C$ occurs infinitely often, $C'$ satisfying $C \rightarrow C'$ occurs infinitely often, we can deduce that $C''$ satisfying $C' \rightarrow C''$ also occurs infinitely often. Overall, with global fairness, if a configuration $C$ occurs infinitely often, then every configuration $C''$ reachable from $C$ also occurs infinitely often.

In this paper, we consider three possibilities for the base station: initialized base station, non-initialized base station, and no base station. In the model with a base station, we assume that a single agent, called a base station, exists in $V$. Then, $V$ can be partitioned into $V_b$, the singleton set containing the base station, and $V_p$, the set of agents except for the base station. The base station can be distinguished from other agents in $V$, although agents in $V_p$ cannot be distinguished. Then, the state set $Q$ can be partitioned into a state set $Q_b$ for the base station, and a state set $Q_p$ for agents in $V_p$. The base station has unlimited resources (with respect to the number of states), in

\footnote{We use this definition for the lower bound under weak fairness, but for the upper bound we use a slightly weaker version. We show that our proposed protocols for weak fairness works if, for each pair of agents $(v, v') \in E$, $v$ and $v'$ interact infinitely often (i.e., for interactions by some pair of agents $v$ and $v'$, it is possible that $v$ only becomes an initiator and $v'$ never becomes an initiator).}
contrast with other resource-limited agents (that are allowed only a limited number of states). So, when we evaluate the space complexity of a protocol, we focus on the number of states $|Q_p|$ for agents in $V_p$ and do not consider the number of states $|Q_b|$ that are allocated to the base station. In the sequel, we thus say a protocol uses $x$ states if $|Q_p| = x$ holds. When we assume an initialized base station, the base station has a designated initial state. When we assume a non-initialized base station, the base station has an arbitrary initial state (in $Q_b$), although agents in $V_p$ have the same designated initial state. When we assume no base station, there exists no base station and thus $V = V_p$ holds. For simplicity, we use agents only to refer to agents in $V_p$ in the following sections. To refer to the base station, we always use the term base station (not an agent). In the initial configuration, both the base station and the agents are not aware of the number of agents, yet they are given an upper bound $P$ of the number of agents. However, in protocols except for a protocol in Section 3.2, we assume that they are not given $P$.

2.2 Uniform Bipartition Problem

Let $f : Q_p \rightarrow \{\text{red}, \text{blue}\}$ be a function that maps a state of an agent to red or blue. We define the color of an agent $a$ as $f(s(a))$. Then, we say that agent $a$ is red (resp., blue) if $f(s(a)) = \text{red}$ (resp., $f(s(a)) = \text{blue}$) holds. If an agent $a$ has state $s$ such that $f(s) = \text{red}$ (resp., $f(s) = \text{blue}$), we call a a red agent (resp., a blue agent). For some population $V$, the number of red agents (resp., blue agents) in $V$ is denoted by $\#\text{red}(V)$ (resp., $\#\text{blue}(V)$). When $V$ is clear from the context, we simply write $\#\text{red}$ and $\#\text{blue}$.

A configuration $C$ is stable with respect to the uniform bipartition if there exists a partition $\{H_r, H_b\}$ of $V_p$ that satisfies the following conditions:

1. $||H_r| - |H_b|| \leq 1$ holds, and
2. For every configuration $C'$ such that $C \xrightarrow{*} C'$, each agent in $H_r$ (resp., $H_b$) remains red (resp., blue) in $C'$.

An execution $\Xi = C_0, C_1, C_2, \ldots$ solves the uniform bipartition problem if $\Xi$ includes a configuration $C_t$ that is stable for uniform bipartition. Finally, a protocol $\mathcal{P}$ solves the uniform bipartition problem if every possible execution $\Xi$ of protocol $\mathcal{P}$ solves the uniform bipartition problem.

3 Upper Bounds with Non-initialized Base Station

In this section, we prove some upper bounds on the number of states that are required to solve the uniform bipartition problem over arbitrary graphs with designated initial states and a non-initialized base station. More concretely, with global fairness, we propose a symmetric protocol with three states by extending the protocol by Yasumi et al. \[34\] from a complete communication graph to an arbitrary communication graph. In the case of weak fairness, we present a symmetric protocol with $3P + 1$ states, where $P$ is a known upper bound of the number of agents.

3.1 Upper Bound for Symmetric Protocols under Global Fairness

The state set of agents in this protocol is $Q_p = \{\text{initial}, \text{red}, \text{blue}\}$, and we assume that $f(\text{initial}) = f(\text{red}) = \text{red}$ and $f(\text{blue}) = \text{blue}$ hold. The designated initial state of agents is initial. The idea of the protocol is as follows: the base station assigns red and blue to agents whose state is initial alternately. As the base station cannot meet every agent (the communication graph is arbitrary), the positions of state initial are moved throughout the communication graph using transitions.
Thus, if an agent with initial state exists somewhere in the network, the base station has infinitely many chances to interact with a neighboring agent with initial state. This implies that the base station is able to repeatedly assign red and blue to neighboring agents with initial state unless no agent anywhere in the network has initial state. Since the base station assigns red and blue alternately, the uniform bipartition is completed after no agent has initial state.

To make red and blue alternately, the base station has a state set \( Q_b = \{ b_{\text{red}}, b_{\text{blue}} \} \). Using its current state, the base station decides which color to use for the next interaction with a neighboring agent with initial state. Now, to move the position of an initial state in the communication graph, if an agent with initial state and an agent with red (or blue) state interact, they exchange their states. This implies that eventually an agent adjacent to the base station has initial state and then the agent and the base station interact (global fairness guarantees that such interaction eventually happens). Transition rules of the protocol are the following (for each transition rule \((p, q) \rightarrow (p', q')\), transition rule \((q, p) \rightarrow (q', p')\) exists, but we omit the description).

1. \((b_{\text{red}}, \text{initial}) \rightarrow (b_{\text{blue}}, \text{red})\)
2. \((b_{\text{blue}}, \text{initial}) \rightarrow (b_{\text{red}}, \text{blue})\)
3. \((\text{blue}, \text{initial}) \rightarrow (\text{initial}, \text{blue})\)
4. \((\text{red}, \text{initial}) \rightarrow (\text{initial}, \text{red})\)

From these transition rules, the protocol converges when no agent has initial state (indeed, no interaction is defined when no agent has initial state).

We now prove the correctness of the protocol.

**Theorem 1.** In the population protocol model with a non-initialized base station, there exists a symmetric protocol with three states per agent that solves the uniform bipartition problem with designated initial states assuming global fairness in arbitrary communication graphs.

**Proof.** Recall that \(#\text{red}\) (resp., \(#\text{blue}\)) denotes the number of agents in state \(s\) such that \(f(s) = \text{red}\) (resp., \(f(s) = \text{blue}\)) holds, thus initially \(#\text{red} = n\) and \(#\text{blue} = 0\) hold. We show that eventually \(|#\text{red} - #\text{blue}| \leq 1\) holds and no agent ever changes its state afterwards. Transition rules 1 and 2 indicate that, if the base station has state \(b_{\text{red}}\) (resp., \(b_{\text{blue}}\)), and an agent with initial state interact, the agent state becomes red (resp., blue), and the base station state becomes \(b_{\text{blue}}\) (resp., \(b_{\text{red}}\)). By repeating these transitions, agents are assigned red and blue alternately, and the number of agents in state initial decreases. This implies that, eventually \(|#\text{red} - #\text{blue}| \leq 1\) holds by alternating transition rules 1 and 2 (transition rules 1 and 2 are never enabled simultaneously, and executing one disables it while enabling the other). Transition rules 3 and 4 indicate that, if an agent with initial state and an agent with red (or blue) state interact, they exchange their states. From the global fairness hypothesis and transition rules 3 and 4, if there exists an agent with initial state, the base station and a neighboring agent with initial state eventually interact. Thus, transition rules 1 and 2 occur repeatedly unless no agent has initial. After all agents have red or blue state, no agent ever changes its color. Recall that agents are assigned red and blue alternately and thus \(|#\text{red} - #\text{blue}| \leq 1\) holds after those assignments. Moreover, the arguments do not depend on the initial state of the base station, and the proposed protocol is symmetric. Therefore, the theorem holds.

Note that, under weak fairness, this protocol does not solve the uniform bipartition problem. This is because we can construct a weakly-fair execution of this protocol such that some agents keep
Algorithm 1 Uniform bipartition protocol with $3P + 1$ states

Variables at the base station:
$RB \in \{r, b\}$: The state that the base station assigns next

Variables at an agent $x$:
$color_x \in \{ini, r, b\}$: Color of the agent, initialized to $ini$
$depth_x \in \{\perp, 1, 2, 3, \ldots, P\}$: Depth of agent $x$ in a tree rooted at the base station, initialized to $\perp$

1: when an agent $x$ and the base station interact do
2: if $color_x = ini$ and $depth_x = 1$ then
3: $color_x \leftarrow RB$
4: $RB \leftarrow \overline{RB}$
5: end if
6: if $depth_x = \perp$ then
7: $depth_x \leftarrow 1$
8: end if
9: end when
10: when two agents $x$ and $y$ interact do
11: if $depth_y \neq \perp$ and $depth_x = \perp$ then
12: $depth_x \leftarrow depth_y + 1$
13: else if $depth_x \neq \perp$ and $depth_y = \perp$ then
14: $depth_y \leftarrow depth_x + 1$
15: end if
16: if $depth_x < depth_y$ and $color_y = ini$ then
17: $color_y \leftarrow color_x$
18: $color_x \leftarrow ini$
19: end if
20: if $depth_y < depth_x$ and $color_x = ini$ then
21: $color_x \leftarrow color_y$
22: $color_y \leftarrow ini$
23: end if
24: end when

Note: If $depth_x = \perp$ holds, $color_x = ini$ holds.

(initial state infinitely often. For example, we can make an agent keep initial by constructing an execution in the following way.

- If the agent (in initial) interacts with an agent in red or blue, the next interaction occurs between the same pair of agents.

3.2 Upper Bound for Symmetric Protocols under Weak Fairness

3.2.1 A protocol over arbitrary graphs

We obtain a symmetric protocol under the assumption by using a similar idea of the transformer proposed in [17]. The transformer simulates an asymmetric protocol on a symmetric protocol. To do this, the transformer requires additional states. Moreover, the transformer works with complete communication graphs. We show that one additional state is sufficient to transform the asymmetric uniform bipartition protocol into the symmetric protocol even if we assume arbitrary graphs.
In this protocol, every agent $x$ has variables $\text{color}_x$ and $\text{depth}_x$. Variable $\text{color}_x$ represents the color of agent $x$. That is, for an agent $x$, if $\text{color}_x = \text{ini}$ or $\text{color}_x = r$ holds, $f(s(x)) = \text{red}$ holds. On the other hand, if $\text{color}_x = b$ holds, $f(s(x)) = \text{blue}$ holds. The protocol is given in Algorithm 1.

Note that this algorithm does not care an initiator and a responder.

The basic strategy of the protocol is the following.

1. Create a spanning tree rooted at the base station. Concretely, agent $x$ assigns its depth in a tree rooted at the base station into variable $\text{depth}_x$. Variable $\text{depth}_x$ is initialized to $\perp$. Variable $\text{depth}_x$ obtains the depth of $x$ in the spanning tree as follows: If the base station and an agent $p$ with $\text{depth}_p = \perp$ interact, $\text{depth}_p$ becomes 1. If an agent $q$ with $\text{depth}_q \neq \perp$ and an agent $p$ with $\text{depth}_p = \perp$ interact, $\text{depth}_p$ becomes $\text{depth}_q + 1$. By these behaviors, for any agent $x$, eventually variable $\text{depth}_x$ has a depth of $x$ in a tree rooted at the base station.

2. Using the spanning tree, carry the initial color $\text{ini}$ toward the base station and make the base station assign $r$ and $b$ to agents one by one. Concretely, if agents $x$ and $y$ interact and both $\text{depth}_y < \text{depth}_x$ and $\text{color}_x = \text{ini}$ hold, $x$ and $y$ exchange their colors (i.e., $\text{ini}$ is carried from $x$ to $y$). Hence, since $\text{ini}$ is always carried to a smaller $\text{depth}$, eventually an agent $z$ with $\text{depth}_z = 1$ obtains $\text{ini}$. After that, the base station and the agent $z$ interact and the base station assigns $r$ or $b$ to $z$. Note that, if the base station assigns $r$ (resp., $b$), the base station assigns $b$ (resp., $r$) next.

Then, for any agent $v$, eventually $\text{color}_v \neq \text{ini}$ holds. Hence, there exist $\lceil n/2 \rceil$ red (resp., blue) agents, and $\lfloor n/2 \rfloor$ blue (resp., red) agents if variable $\text{RB}$ in the base station has $r$ (resp., $b$) as an initial value. Therefore, the protocol solves the uniform bipartition problem.

From now on, we demonstrate the correctness of the protocol. Let $\#r = |\{x|\text{color}_x = \text{ini}\}|$, $\#b = |\{x|\text{color}_x = r\}|$, and $\#\text{ini} = |\{x|\text{color}_x = b\}|$. First, we show that $\#r$, $\#b$, and $\#\text{ini}$ do not change except for an interaction between the base station and an agent with a specific state.

**Lemma 2.** For any weakly-fair execution of Algorithm 1 unless an interaction occurs between the base station and an agent $v$ such that $\text{depth}_v = 1$ and $\text{color}_v = \text{ini}$ hold, $\#r$, $\#b$, and $\#\text{ini}$ do not change.

**Proof.** For each interaction occurring in a weakly-fair execution of the protocol, we consider two cases.

First, we consider the case that the base station does not participate to the interaction. From the protocol, when agents interact, the color of agents may only change in lines 17-18 or 21-22 of the pseudocode. On lines 17-18 and 21-22, two agents $x$ and $y$ just exchange $\text{color}_x$ and $\text{color}_y$. Thus, $\#r$, $\#b$, and $\#\text{ini}$ do not change in this case.

Next, we consider the case that the base station participates the interaction. In the case, from the protocol, variable $\text{color}$ changes only if the base station interacts with an agent $v$ such that $\text{depth}_v = 1$ and $\text{color}_v = \text{ini}$ hold. Thus, the lemma holds.

We now show three basic properties of variable $\text{depth}$.

**Lemma 3.** For any agent $v$, when $\text{depth}_v \neq \perp$ holds in a configuration, $\text{depth}_v$ does not change afterwards.

**Proof.** From the protocol, variable $\text{depth}_v$ of agent $v$ changes only in line 7, 12, or 14 of the pseudocode. On lines 7, 12, and 14, $\text{depth}_v$ changes only if $\text{depth}_v = \perp$ holds. Thus, the lemma holds.
Lemma 4. For any agent $v$, $\text{depth}_v \neq \bot$ holds after some configuration in any weakly-fair execution of Algorithm 1.

Proof. By the weak fairness assumption, for any agent $v_0$ adjacent to the base station, eventually $v_0$ and the base station interact. By lines 6-7 of the pseudocode, $\text{depth}_{v_0} \neq \bot$ holds after the interaction with the base station occurs. From Lemma 3, for any agent $v$, once $\text{depth}_v \neq \bot$ holds, $\text{depth}_v \neq \bot$ holds perpetually. Hence, if agent $v$ with $\text{depth}_v = \bot$ is adjacent to $v'$ with $\text{depth}_{v'} \neq \bot$, eventually agents $v$ and $v'$ interact by the weak fairness assumption, and after the interaction occurs, $\text{depth}_v \neq \bot$ holds. This implies that, for any agent $v$, eventually $\text{depth}_v \neq \bot$ holds.

Lemma 5. For any agent $v_1$, when $\text{depth}_{v_1}$ is neither $\bot$ nor one, then $v_1$ is adjacent to an agent $v_2$ such that $\text{depth}_{v_1} = \text{depth}_{v_2} + 1$ holds. When $\text{depth}_{v_1}$ is one, $v_1$ is adjacent to the base station.

Proof. We focus on interactions such that an agent $v$ assigns a value other than $\bot$ to $\text{depth}_v$. By Lemmas 3 and 4 the number of such interactions for a given agent is exactly one. We consider two cases.

First, we consider the case when $\text{depth}_v$ becomes one by the interaction. From the protocol, $\text{depth}_v$ can become one only if the base station and $v$ interact. Hence, $v$ is adjacent to the base station.

Next, we consider the case that $\text{depth}_v$ becomes neither $\bot$ nor one by the interaction. From the protocol, if the interaction happens between $v$ and an agent $w$, $\text{depth}_v$ becomes $\text{depth}_w + 1$ by the interaction. By Lemma 3, $\text{depth}_w$ and $\text{depth}_v$ never change afterward. Thus, the lemma holds.

From the above lemmas, we show that eventually $\#\text{ini} = 0$ holds in any weakly-fair execution of the protocol.

Lemma 6. For any weakly-fair execution of Algorithm 1, $\#\text{ini} = 0$ holds after finite time.

Proof. For the purpose of contradiction, let us assume that, after some configuration $C$, $\#\text{ini} > 0$ never decreases. Let us consider a configuration $C'$ such that $\text{depth}_v \neq \bot$ holds for any agent $v$ in $C'$, and $C'$ appears after $C$. By Lemma 4, such $C'$ exists. In $C'$, let $\text{Ini} = \{x | \text{color}_x = \text{ini}\}$, and let $v_1 \in \text{Ini}$ be an agent such that $\text{depth}_{v_1} = \min\{\text{depth}_x | x \in \text{Ini}\}$ holds. By Lemma 5, either $v_1$ is adjacent to an agent $v_2$ such that $\text{depth}_{v_2} = \text{depth}_{v_1} - 1$ or $\text{depth}_{v_1} = 1$ holds.

First, we consider the case when $\text{depth}_{v_1} = 1$ holds. By Lemma 5, $v_1$ is adjacent to the base station. Since $\text{depth}_{v_1} = 1$ holds, there is not an agent $v_2'$ such that $\text{depth}_{v_2'} < \text{depth}_{v_1}$ holds. Hence, from the protocol, $\text{color}_{v_1}$ keeps $\text{ini}$ unless $v_1$ and the base station interact. Thus, eventually $v_1$ with $\text{color}_{v_1} = \text{ini}$ and the base station interact. Then, $\#\text{ini}$ decreases in this case.

Next, we consider the case that $v_1$ is adjacent to an agent $v_2$ such that $\text{depth}_{v_2} = \text{depth}_{v_1} - 1$. From the protocol, since $\text{depth}_{v_1} \neq 1$ holds, $\text{color}_{v_1}$ keeps $\text{ini}$ unless an interaction happens between $v_1$ and an agent $v_2'$ such that $\text{depth}_{v_2'} < \text{depth}_{v_1}$ holds. From the weak fairness, eventually $v_1$ and such $v_2'$ interact and $\text{color}_{v_2'}$ becomes $\text{ini}$. At that time, the smallest depth of agents with color $\text{ini}$ decreases. By repeating this behavior similarly, eventually some agent $v_h$ with $\text{depth}_{v_h} = 1$ obtains color $\text{ini}$. After that, $v_h$ and the base station interact and $\#\text{ini}$ decreases. This is a contradiction.

Next, we prove that $|\#r - \#b| \leq 1$ holds in any weakly-fair execution of the protocol.

Lemma 7. For any configuration in any weakly-fair execution of Algorithm 1 either (i) both $\#r - \#b = 1$ and $\text{RB} = b$ hold, or (ii) both $\#r - \#b = 0$ and $\text{RB} = r$ hold.
Proof. Consider an execution $Ξ = C_0, C_1, \ldots$ of the protocol. We prove the lemma by induction on the index of a configuration. In the base case ($C_0$), $#r = #b = 0$ and $RB = r$ hold and thus alternative (ii) of the lemma holds immediately.

For the induction step, assume that there exists an integer $i \geq 0$ such that the lemma holds in $C_i$. Consider an interaction at $C_i \rightarrow C_{i+1}$. We consider two cases.

First, we consider the case where, for an agent $v$ such that $\text{depth}_v = 1$ and $\text{color}_v = \text{ini}$ hold, the base station and $v$ do not interact. By Lemma 2, when the base station and such $v$ do not interact, $#r$, $#b$, and $#\text{ini}$ do not change. Moreover, from the protocol, $RB$ does not change at the interaction, and thus either alternatives (i) of (ii) of the lemma holds in this case.

Next, we consider the case where, for an agent $v$ such that $\text{depth}_v = 1$ and $\text{color}_v = \text{ini}$ hold, the base station and $v$ interact.

• In the case where $RB = r$ holds in $C_i$: After the interaction at $C_i \rightarrow C_{i+1}$, $#r$ increases by one and $RB$ becomes $b$. By the induction assumption, since $#r - #b = 0$ holds in $C_i$, $#r - #b = 1$ holds in $C_{i+1}$. Hence, alternative i) of the lemma holds.

• In the case where $RB = b$ holds in $C_i$: After the interaction at $C_i \rightarrow C_{i+1}$, $#r$ increases by one and $RB$ becomes $r$. By the induction assumption, since $#r - #b = 1$ holds in $C_i$, $#r - #b = 0$ holds in $C_{i+1}$. Hence, alternative (ii) of the lemma holds.

Thus, the lemma holds.

Using Lemmas 6 and 7, we show that the protocol solves the uniform bipartition problem.

Theorem 8. Algorithm 1 solves the uniform bipartition problem. That is, there exists a protocol with $3P + 1$ states and designated initial states that solves the uniform bipartition problem under weak fairness assuming arbitrary communication graphs with a non-initialized base station.

Proof. By Lemmas 6 and 7, $|#r - #b| \leq 1$ holds in any weakly-fair execution of the protocol and eventually $#\text{ini} = 0$ holds in the execution. Moreover, from the protocol, when there exists no agent $v$ such that $\text{color}_v = \text{ini}$, any agent $x$ does not change its $\text{color}_x$. Thus, the protocol solves the problem. Additionally, the protocol works with $2P + (P + 1)$ states. This is because, if $\text{depth}_x = \bot$ holds, $\text{color}_x = \text{ini}$ holds. That is, $\text{depth}_x$ takes $\bot, 1, 2, \ldots, P$ for $\text{ini}$, and takes $1, 2, \ldots, P$ for $r$ and $b$.

Therefore, the theorem holds.

3.2.2 A protocol with constant states over a restricted class of graphs

In this subsection, we show that the space complexity of Algorithm 1 can be reduced to constant for communication graphs such that every cycle either includes the base station or its length is not a multiple of $l$, where $l$ is a positive integer at least three.

We modify Algorithm 1 as follows. Each agent maintains the distance from the base station by computing modulo $l$ plus 1. That is, we change lines 12 and 14 in Algorithm 1 to $\text{depth}_x \leftarrow \text{depth}_y \mod l + 1$ and $\text{depth}_y \leftarrow \text{depth}_x \mod l + 1$, respectively. Now $\text{depth}_x \in \{\bot, 1, 2, 3, \ldots, l\}$ holds for any agent $x$. Then we redefine the relation $\text{depth}_x < \text{depth}_y$ in lines 16 and 20 as follows: $\text{depth}_x < \text{depth}_y$ holds if and only if either $\text{depth}_x = 1 \land \text{depth}_y = 2$, $\text{depth}_x = 2 \land \text{depth}_y = 3$, $\text{depth}_x = 3 \land \text{depth}_y = 4$, $\ldots$, $\text{depth}_x = l - 1 \land \text{depth}_y = l$, or $\text{depth}_x = l \land \text{depth}_y = 1$ holds.

We can easily observe that these modifications do not change the essence of Algorithm 1. For two agents $x$ and $y$, we say $x < y$ if $\text{depth}_x < \text{depth}_y$ holds. Each agent $x$ eventually assigns a depth of $x$ modulo $l$ plus 1 to $\text{depth}_x$, and at that time there exists a path $x_0, x_1, \ldots, x_h$ such that
Algorithm 2 Uniform bipartition protocol with four states

A state set
\[ Q = \{ r^\omega, b^\omega, r, b \} \]

A mapping to colors
\[ f(r^\omega) = f(r) = \text{red} \]
\[ f(b^\omega) = f(b) = \text{blue} \]

Transition rules
1. \((r^\omega, r^\omega) \rightarrow (r, b)\)
2. \((r^\omega, b^\omega) \rightarrow (b, b)\)
3. \((r^\omega, r) \rightarrow (r, r^\omega)\)
4. \((b^\omega, b) \rightarrow (b, b^\omega)\)
5. \((r^\omega, b) \rightarrow (r, b^\omega)\)
6. \((b^\omega, r) \rightarrow (b, r^\omega)\)

- \(x_0\) is a neighbor of the base station, \(x = x_h\) holds, and \(x_i < x_{i+1}\) holds for any \(0 \leq i < h\). In addition, there exists no cycle \(x_0, x_1, \ldots, x_h = x_0\) such that \(x_i < x_{i+1}\) holds for any \(0 \leq i < h\). This is because, from the definition of relation ‘<’, the length of such a cycle should be a multiple of \(l\), but we assume that underlying communication graphs do not include a cycle of agents in \(V_p\) whose length is a multiple of \(l\). Hence, similarly to Algorithm 1, we can carry the initial color \(ini\) toward the base station and make the base station assign \(r\) and \(b\) to agents one by one.

Corollary 9. There exists a protocol with \(3l + 1\) states and designated initial states that solves the uniform bipartition problem under weak fairness assuming arbitrary communication graphs with a non-initialized base station if, for any cycle of the communication graphs, it either includes the base station or its length is not a multiple of \(l\), where \(l\) is a positive integer at least three.

4 Upper and Lower Bounds with No Base Station

In this section, we show upper and lower bounds of the number of states to solve the uniform bipartition problem with no base station and designated initial states over arbitrary communication graphs. Concretely, under global fairness, we prove that the minimum number of states for asymmetric protocols is four, and the minimum number of states for symmetric protocols is five. Under weak fairness, we prove that the uniform bipartition problem cannot be solved without a base station using proof techniques similar to those Fischer and Jiang [21] used to show the impossibility of leader election.

4.1 Upper Bound for Asymmetric Protocols under Global Fairness

In this subsection, over arbitrary graphs with designated initial states and no base station under global fairness, we give an asymmetric protocol with four states.

We define a state set of agents as \(Q = \{ r^\omega, b^\omega, r, b \} \), and function \(f\) as follows: \(f(r^\omega) = f(r) = \text{red}\) and \(f(b^\omega) = f(b) = \text{blue}\). We say an agent has a token if its state is \(r^\omega\) or \(b^\omega\). Initially, every agent has state \(r^\omega\), that is, every agent is \(\text{red}\) and has a token. The transition rules are given in Algorithm 2 (for each transition rule \((p, q) \rightarrow (p', q')\), transition rule \((q, p) \rightarrow (q', p')\) exists, but we omit the description except for transition rule 1).
The basic strategy of the protocol is as follows. When two agents with tokens interact and one of them is red, a red agent transitions to blue and the two tokens are deleted (transition rules 1 and 2). Since \( n \) tokens exist initially and the number of tokens decreases by two in an interaction, \([n/2]\) blue agents appear and \([n/2]\) red agents remain after all tokens (except one token for the case of odd \( n \)) disappear. To make such interactions, the protocol moves a token when agents with and without a token interact (transition rules 3, 4, 5, and 6). Global fairness guarantees that, if two tokens exist, an interaction of transition rule 1 or 2 happens eventually. Therefore, the uniform bipartition is achieved by the protocol.

From now, we prove the correctness of the protocol. We define \( \#r, \#b, \#r^\omega, \#b^\omega \) as the number of agents that have state \( r, b, r^\omega, b^\omega \), respectively. Let \( \#red = \#r + \#r^\omega \) and \( \#blue = \#b + \#b^\omega \) be the number of red and blue agents, respectively. Let \( \#token = \#r^\omega + \#b^\omega \) be the number of agents with tokens.

**Lemma 10.** In any globally-fair execution of Algorithm \( \Xi \), \( \#r = \#b + 2 \times \#b^\omega \) holds in any configuration.

**Proof.** Consider an execution \( \Xi = C_0, C_1, \ldots \) of the protocol. We prove the equation by induction on the index of a configuration. The base case is the case of \( C_0 \). In this case, the equation holds because all agents have \( r^\omega \) initially. For the induction step, assume that the equation holds in \( C_i (0 \leq i) \). Consider an interaction at \( C_i \rightarrow C_{i+1} \) for each transition rule.

- **Transition rule 1:** When the transition rule 1 occurs at \( C_i \rightarrow C_{i+1} \), \#\( r \) and \#\( b \) increase by one.
- **Transition rule 2:** When the transition rule 2 occurs at \( C_i \rightarrow C_{i+1} \), \#\( b^\omega \) decreases by one, and \#\( b \) increases by two.
- **Transitions rule 3 and 4:** When the transition rule 3 or 4 occurs at \( C_i \rightarrow C_{i+1} \), \#\( r \), \#\( b \), and \#\( b^\omega \) do not change.
- **Transition rule 5:** When the transition rule 5 occurs at \( C_i \rightarrow C_{i+1} \), \#\( b \) decreases by one, and \#\( r \) and \#\( b^\omega \) increase by one.
- **Transition rule 6:** When the transition rule 6 occurs at \( C_i \rightarrow C_{i+1} \), \#\( b \) increases by one, and \#\( r \) and \#\( b^\omega \) decrease by one.

For every case, \( \#r = \#b + 2 \times \#b^\omega \) holds in \( C_{i+1} \). Therefore, the lemma holds.

**Lemma 11.** For any globally-fair execution of the Algorithm \( \Xi \), \#\( token \leq 1 \) holds after finite time.

**Proof.** First of all, there is no transition rule that increases \#\( token \). This implies that, if \#\( token \leq 1 \) holds at some configuration, \#\( token \leq 1 \) holds thereafter. Hence, for the purpose of contradiction, we assume that there exists a globally-fair execution \( \Xi \) of the protocol where \#\( token = x > 1 \) continuously holds after some configuration.

Consider a configuration \( C \) that occurs infinitely often in \( \Xi \). Note that \( C \) is stable and satisfies \#\( token = x \). Consider two agents \( a_1 \) and \( a_2 \) such that \( a_1 \) is adjacent to \( a_2 \) and one of them is red in \( C \).

Since tokens can move through a graph by swapping states (transition rules 3,4,5, and 6), a configuration \( C' \) such that \( a_1 \) and \( a_2 \) have a token is reachable from \( C \). When \( a_1 \) and \( a_2 \) interact at \( C' \rightarrow C'' \), \#\( token = x - 2 \) holds in \( C'' \). From the global fairness assumption, since \( C \) occurs infinitely often in \( \Xi \), \( C'' \) also occurs infinitely often. Since \#\( token = x \) continuously holds after some configuration, this is a contradiction.
Next, by using these lemmas, we show that Algorithm 2 solves the problem under the assumptions.

**Theorem 12.** Algorithm 2 solves the uniform bipartition problem. That is, there exists a protocol with four states and designated initial states that solves the uniform bipartition problem under global fairness over arbitrary communication graphs.

**Proof.** Since \#token is reduced only by transition rules 1 and 2, \#token is reduced by two in an interaction. This implies that, by Lemma 11, when \( n \) is even (resp., odd), \#token = 0 (resp., \#token = 1) holds after some configuration \( C \).

First, we consider the case that \( n \) is even. By Lemma 10 if \#token = 0 holds, \#r = \#b holds. Hence, since \( n = \#r + \#b + \#token \) holds, \#r = \#b = \( n/2 \) holds at \( C \). Moreover, since agents can change their colors only if \#token \( \geq 2 \) holds, they do not change their colors after \( C \). Hence \( C \) is a stable configuration, and thus the uniform bipartition is completed.

Next, we consider the case that \( n \) is odd. When \#token = 1 holds, we consider two cases. If an agent in state \( r^w \) exists at \( C \), by Lemma 10 \#r = \#b holds and thus \#red = \#r + \#r^w = \#b + 1 = \#blue + 1 \) holds. If an agent in state \( b^w \) exists at \( C \), by Lemma 10 \#r = \#b + 2 holds and thus \#red = \#r = \#b + \#b^w + 1 = \#blue + 1 \) holds. Hence, in both cases, \#red − \#blue = 1 holds at \( C \). Since agents can change their colors only if \#token \( \geq 2 \) holds, they do not change their colors after \( C \). Hence \( C \) is a stable configuration, and thus the uniform bipartition is completed. \( \square \)

### 4.2 Upper Bound for Symmetric Protocols under Global Fairness

In this subsection, with arbitrary communication graphs with designated initial states and no base station under global fairness, we give a symmetric protocol with five states.

Observe that, with designated initial states and no base station, clearly no symmetric protocol can solve the problem if the number of agents \( n \) is two (the state of the two agents is the same in the initial state, so symmetry is never broken and uniform bipartition cannot occur). Thus, we assume that \( 3 \leq n \) holds.

We define a state set of agents as \( Q = \{ r_0^w, r_1^w, b^w, r, b \} \), and function \( f \) as follows: \( f(r_0^w) = f(r_1^w) = f(r) = \text{red} \) and \( f(b^w) = f(b) = \text{blue} \). We say an agent has a token if its state is \( r_0^w \), \( r_1^w \), or \( b^w \). Initially, every agent has state \( r_0^w \), that is, every agent is \text{red} and has a token. The transition rules are given in Algorithm 3.

The idea of Algorithm 3 is similar to Algorithm 2. That is, when two agents with tokens interact and one of them is \text{red}, a \text{red} agent transitions to \text{blue}, and the two tokens are deleted. Then, eventually \( \lceil n/2 \rceil \) \text{blue} agents appear and \( \lfloor n/2 \rfloor \) \text{red} agents remain after all tokens (except one token for the case of odd \( n \)) disappear. However, to make a \text{red} agent transition to a \text{blue} agent in the first place, Algorithm 2 includes transition rule 1 that makes agents with the same states transition to different states. This implies that Algorithm 2 is not symmetric. Hence, by borrowing the technique proposed in [17], we improve Algorithm 2 so that the new protocol (Algorithm 3) makes a \text{red} agent transition to a \text{blue} agent without such a transition (and two tokens are deleted at that time). Concretely, we realize it as follows. In Algorithm 3 there are two states \( r_0^w \) and \( r_1^w \) that are \text{red} and have a token. When two agents with \( r_0^w \) interact, they transition to \( r_1^w \), and vice versa (transition rules 1 and 2). Thus, under global fairness, eventually an agent with \( r_0^w \) interacts with an agent with \( r_1^w \) and then one of them transitions to \text{blue}, and two tokens are deleted (transition rule 3). Observe that, these transitions do not affect the essence of Algorithm 2. This is because the numbers of \text{blue} agents, \text{red} agents, and tokens do not change after transition rules 1 and 2, and a \text{red} agent transitions to \text{blue}, and two tokens are deleted at transition rule 3.
Algorithm 3 Uniform bipartition protocol with five states

A state set
\[ Q = \{ r_0^\omega, r_1^\omega, b^\omega, r, b \} \]

A mapping to colors
\[ f(r_0^\omega) = f(r_1^\omega) = f(r) = \text{red} \]
\[ f(b^\omega) = f(b) = \text{blue} \]

Transition rules
1. \((r_0^\omega, r_0^\omega) \rightarrow (r_1^\omega, r_1^\omega)\)
2. \((r_1^\omega, r_1^\omega) \rightarrow (r_0^\omega, r_0^\omega)\)
3. \((r_0^\omega, r_1^\omega) \rightarrow (r, b)\)
4. \((r_0^\omega, r) \rightarrow (r, r_0^\omega)\)
5. \((b^\omega, b) \rightarrow (b, b^\omega)\)
6. \((r_0^\omega, b^\omega) \rightarrow (b, b^\omega)\)
7. \((r_1^\omega, b^\omega) \rightarrow (b, r_0^\omega)\)
8. \((b^\omega, r) \rightarrow (b, r_0^\omega)\)
9. \((r_0^\omega, b^\omega) \rightarrow (b, b)\)
10. \((r_1^\omega, b^\omega) \rightarrow (b, b)\)

We now prove the correctness of Algorithm 3 along the proof in subsection 4.1. We define \(#r\), \(#b\), \(#r_0^\omega\), \(#r_1^\omega\), \(#b^\omega\) as the number of agents that have state \(r\), \(b\), \(r_0^\omega\), \(r_1^\omega\), \(b^\omega\), respectively. Let \(#\text{red} = #r + #r_0^\omega + #r_1^\omega\) and \(#\text{blue} = #b + #b^\omega\) be the number of red and blue agents, respectively. Let \(#\text{token} = #r_0^\omega + #r_1^\omega + #b^\omega\) be the number of agents with tokens.

Recall that a mechanism of symmetry breaking (transition rules 1, 2, and 3) does not affect the essence of Algorithm 2. In particular, if \(r_0^\omega = r_1^\omega = \omega\) holds, Algorithm 3 is equal to Algorithm 2. Thus, since an equation of Lemma 10 holds in any configuration of any execution of Algorithm 3 and \(#\text{red}\) is not related to the equation, the following corollary holds.

**Corollary 13.** In any globally-fair execution of Algorithm 3, \(#r = #b + 2\times #b^\omega\) holds in any configuration.

Moreover, we show the following lemma similarly to Lemma 11.

**Lemma 14.** For any globally-fair execution of Algorithm 3, \(#\text{token} \leq 1\) holds after some configuration.

**Proof.** First of all, there is no transition rule that increase \(#\text{token}\). This implies that, if \(#\text{token} \leq 1\) holds at some configuration, \(#\text{token} \leq 1\) holds thereafter. Hence, for the purpose of contradiction, we assume that there exists a globally-fair execution \(\Xi\) of Algorithm 3 where \(#\text{token} = x > 1\) continuously holds after some configuration.

Consider a configuration \(C\) that occurs infinitely often in \(\Xi\). Note that \(C\) is stable and satisfies \(#\text{token} = x\). First, we show that at least one blue agent occurs from the initial configuration. When \(n = #r_0^\omega + #r_1^\omega\) holds, only transition rules 1, 2, and 3 can occur. From the global fairness assumption, an agent with \(r_0^\omega\) is adjacent to an agent with \(r_1^\omega\) at the same time and then they interact. By the interaction, transition rule 3 happens and thus one blue agent appears.
Since there is no transition rule that decreases the number of blue agents, at least one blue agent exists after some configuration of $\Xi$. Moreover, by Corollary 13 if there exists blue agent, there also exists red agent. Hence, in $C$, there exist at least one blue agent and at least one red agent. Consider two agents $a_1$ and $a_2$ such that $a_1$ is adjacent to $a_2$ and $a_1$ (resp., $a_2$) is red (resp., blue) in $C$.

Since tokens can move through a graph by swapping states (transition rules 4, 5, 6, 7, 8, and 9), a configuration $C'$ such that $a_1$ and $a_2$ have a token is reachable from $C$. When $a_1$ and $a_2$ interact at $C' \rightarrow C''$, $\#token = x - 2$ holds in $C''$. From the global fairness assumption, since $C$ occurs infinitely often in $\Xi$, $C''$ also occurs infinitely often. Since $\#token = x$ continuously holds after some configuration, this is a contradiction.

Finally, we show, similarly to Theorem 12 that Algorithm 3 solves the uniform bipartition problem.

**Theorem 15.** Algorithm 3 solves the uniform bipartition problem. That is, there exists a symmetric protocol with five states and designated initial states that solves the uniform bipartition problem under global fairness with arbitrary communication graphs.

**Proof.** Since $\#token$ is reduced only in transition rules 3, 10, and 11, $\#token$ is reduced by two in an interaction. This implies that, by Lemma 14, when $n$ is even (resp., odd), $\#token = 0$ (resp., $\#token = 1$) holds after some configuration $C$.

First, we consider the case where $n$ is even. By Corollary 13 if $\#token = 0$, $\#r = \#b$ holds. Hence, since $n = \#r + \#b + \#token$ holds, $\#r = \#b = n/2$ holds at $C$. Moreover, since agents can change their colors only if $\#token \geq 2$ holds, they do not change their colors after $C$. Hence $C$ is a stable configuration, and thus the uniform bipartition is completed.

Next, we consider the case that $n$ is odd. When $\#token = 1$ holds, we consider two cases. If an agent in state $r_0^0$ or $r_1^0$ exists at $C$, by Corollary 13 $\#r = \#b$ holds and thus $\#red = \#r + \#r_0^0 + \#r_1^0 = \#b + 1 = \#blue + 1$ holds. If an agent in state $b^0$ exists at $C$, by Corollary 13 $\#r = \#b + 2$ holds and thus $\#red = \#r = \#b + \#b^0 + 1 = \#blue + 1$ holds. Hence, in both cases, $\#red = \#blue = 1$ holds at $C$. Since agents can change their colors only if $\#token \geq 2$ holds, they do not change their colors after $C$. Hence $C$ is a stable configuration, and thus the uniform bipartition is completed.

### 4.3 Lower Bound for Asymmetric Protocols under Global Fairness

In this section, we show that, over arbitrary graphs with designated initial states and no base station under global fairness, there exists no asymmetric protocol with three states.

To prove this, we first show that, when the number of agents $n$ is odd and no more than $P/2$, each agent changes its own state to another state infinitely often in any globally-fair execution $\Xi$ of a uniform bipartition protocol $Alg$, where $P$ is a known upper bound of the number of agents. This proposition holds regardless of the number of states in a protocol.

After that, we prove impossibility of an asymmetric protocol with three states. The outline of the proof is as follows. For the purpose of contradiction, we assume that there exists a protocol $Alg$ that solves the problem with three states. From the above proposition, in any globally-fair execution, some agents change their state infinitely often. Now, with three states, the number of red or blue states is at least one and thus, if we assume without loss of generality that the number of blue states is one, agents with the blue state change their color eventually. This is a contradiction.

From now on, we show that, in any globally-fair execution $\Xi$ of a protocol $Alg$ solving uniform bipartition over an arbitrary communication graph such that the number of agents $n < P/2$ is odd, all agents transition their own state to another state infinitely often.
Lemma 16. Assume that there exists a uniform bipartition protocol $Alg$ with designated initial states over arbitrary communication graphs assuming global fairness. Consider a graph $G = (V, E)$ such that the number of agents $n$ is odd and no more than $P/2$. In any globally-fair execution $Ξ = C_0, C_1, \ldots$ of $Alg$ over $G$, each agent changes its state infinitely often.

Proof. The outline of the proof is as follows. First, for the purpose of contradiction, we assume that there exists an agent $v_α$ that never changes its state after some stable configuration $C_h$ in a globally-fair execution $Ξ$ over graph $G$. Let $s_α$ be a state that $v_α$ has after $C_h$. Let $v_β ∈ V$ be an agent adjacent to $v_α$ and $S_β$ be a set of states that $v_β$ has after $C_h$. Since the number of states is finite, there exists a stable configuration $C_t$ that occurs infinitely often after $C_h$. Next, let $G'_1 = (V'_1, E_1)$ and $G'_2 = (V'_2, E_2)$ be graphs that are isomorphic to $G$. Moreover, let $v'_α ∈ V'_1$ (resp., $v'_{n+β} ∈ V'_2$) be an agent that corresponds to $v_α ∈ V$ (resp., $v_β ∈ V$). We construct $G' = (V', E')$ by connecting $G'_1$ and $G'_2$ with an additional edge $(v'_α, v'_{n+β})$. Over $G'$, we consider an execution $Ξ'$ such that, agents in $G'_1$ and $G'_2$ behave similarly to $Ξ$ until $C_t$ occurs in $G'_1$ and $G'_2$, and then make interactions so that $Ξ'$ satisfies global fairness. Since $Ξ$ is globally fair, we can show the following facts after $G'_1$ and $G'_2$ reach $C_t$ in $Ξ'$.

- $v'_α$ has state $s_α$ as long as $v'_{n+β}$ has a state in $S_β$.
- $v'_{n+β}$ has a state in $S_β$ as long as $v'_α$ has state $s_α$.

From these facts, in $Ξ'$, $v'_α$ continues to have state $s_α$ and $v'_{n+β}$ continues to have a state in $S_β$. Hence, in $Ξ'$, each agent in $V'_1$ cannot notice the existence of agents in $V'_2$, and vice versa. This implies that, in stable configurations, $\#red(V) = \#red(V'_1) = \#red(V'_2)$ and $\#blue(V) = \#blue(V'_1) = \#blue(V'_2)$ hold. Since the number of agents in $G$ is odd, $\#red(V) − \#blue(V) = 1$ or $\#blue(V) − \#red(V) = 1$ holds in stable configurations of $Ξ$. Thus, in stable configurations of $Ξ'$, $|\#red(V') − \#blue(V')| = 2$ holds. Since $Ξ'$ is globally fair, this is a contradiction.

From now on, we show the details of the proof.

Let $V = \{v_0, v_1, v_2, v_3, \ldots, v_{n-1}\}$. Assume, for the purpose of contradiction, that there exists $v_α$ that does not change its state after some stable configuration $C_h$ in globally-fair execution $Ξ$. Let $s_α$ be a state that $v_α$ has after $C_h$. Since the number of states is finite, in $Ξ$, there exists a stable configuration $C_t$ that appears infinitely often after $C_h$. Without loss of generality, $\#red(V) − \#blue(V) = 1$ holds after $C_h$. Let $v_β$ be an agent that is adjacent to $v_α$.

Next, consider a communication graph $G' = (V', E')$ that satisfies the following.

- $V' = V'_1 ∪ V'_2$, where $V'_1 = \{v'_0, v'_1, v'_2, \ldots, v'_{n-1}\}$ and $V'_2 = \{v'_{n}, v'_{n+1}, v'_{n+2}, \ldots, v'_{2n-1}\}$.
- $E' = \{(v'_x, v'_y), (v'_{x+n}, v'_{y+n}) ∈ V' × V' \mid (v_x, v_y) ∈ E\} ∪ \{(v'_α, v'_{n+β})\}$.

An example ($n = 5$) of $G$ and $G'$ is shown in Figure 1.

Consider a globally-fair execution $Ξ' = C'_0, C'_1, C'_2, C'_3, \ldots$ over $G'$ as follows:

- For $i ≤ t$, when $v_x$ interacts with $v_y$ at $C_i \rightarrow C_{i+1}$, $v'_x$ interacts with $v'_y$ at $C'_2i \rightarrow C'_{2i+1}$, and $v'_{x+n}$ interacts with $v'_{y+n}$ at $C'_{2i+1} \rightarrow C'_{2i+2}$.

Figure 1: An example of communication graphs $G$ and $G' (n = 5)$
• After $C'_{2t}$, agents make interactions so that $\Xi'$ satisfies global fairness.

In the following, for configuration $C'$ in $\Xi'$ and configuration $C$ in $\Xi$, we say $C'$ of $V'_1$ (resp., $V'_2$) is equivalent to $C$ if $s(v'_i, C') = s(v_x, C)$ (resp., $s(v'_{x+n}, C') = s(v_x, C)$) holds for any $v_x \in V$. Observe that, by the definition of $\Xi'$, $C'_{2t}$ of $V'_1$ and $V'_2$ is equivalent to $C_t$.

From now on, by induction on the index of configuration, we prove the proposition that, for any configuration $C'_m$ that occurs after $C'_{2t}$, there is a configuration $C_a$ (resp., $C_b$) such that 1) $C'_m$ of $V'_1$ (resp., $V'_2$) is equivalent to $C_a$ (resp., $C_b$) and 2) $C_a$ (resp., $C_b$) appears infinitely often in $\Xi$.

The base case is $C'_m = C'_{2t}$. Since $C_t$ appears infinitely often in $\Xi$ and $C'_{2t}$ of $V'_1$ and $V'_2$ is equivalent to $C_t$, the base case holds.

For the induction step, assume that, for $C'_m (m \geq 2t)$, there is a configuration $C_a$ (resp., $C_b$) that satisfies the conditions. We consider two cases for an interaction at $C'_m \rightarrow C'_{m+1}$. The first case considers an interaction of $v'_a$ and $v'_{n+\beta}$, and the second case considers other interactions.

First, we consider the case that $v'_a$ and $v'_{n+\beta}$ interact at $C'_m \rightarrow C'_{m+1}$. By the assumption, we have $s(v'_a, C'_m) = s(v_a, C_a) = s_a$, $s(v'_{n+a}, C'_m) = s(v_a, C_b) = s_{a}$, and $s(v'_{n+\beta}, C'_m) = s(v_{\beta}, C_b)$. Hence, since $v_a$ does not change its state even when it interacts with $v_{\beta}$ at $C_b$, a transition rule $(s(v_{\beta}, C_b), s_a) \rightarrow (s_a, s_a)$ exists, where $s$ is some state. This implies that, when $v'_a$ and $v'_{\beta+n}$ interact at $C'_m$, $v'_a$ keeps the state $s_a$. Thus, $C'_{m+1}$ of $V'_1$ is still equivalent to $C_a$. Additionally, since $s(v'_a, C'_m) = s(v'_{n+a}, C'_m) = s(v_a, C_b) = s_a$ and $s(v'_{n+\beta}, C'_m) = s(v_{\beta}, C_b)$ hold, $v'_{n+\beta}$ changes its state similarly to the case that $v_{\beta}$ interacts with $v_a$ in $C_b$. That is, letting $C_{b'}$ be a configuration immediately after $v_a$ and $v_{\beta}$ interact at $C_b$, $C'_{m+1}$ of $V'_2$ is equivalent to $C_{b'}$. Since $C_b$ occurs infinitely often in $\Xi$ and $\Xi$ is globally fair, $C_{b'}$ occurs infinitely often in $\Xi$. Thus, the proposition holds for $C'_{m+1}$.

Next, we consider the case that at least one agent other than $v'_a$ and $v'_{n+\beta}$ joins an interaction at $C'_m \rightarrow C'_{m+1}$. By the definition, no edge other than $(v'_a, v'_{n+\beta})$ connects $V'_1$ and $V'_2$. Hence, if $v'_i$ and $v'_j$ interact at $C'_m \rightarrow C'_{m+1}$, either $v'_i \in V'_1 \land v'_j \in V'_1$ or $v'_i \in V'_2 \land v'_j \in V'_2$ holds. In the former case, letting $C_{a'}$ be the configuration immediately after $v_i$ and $v_j$ interact at $C_a$, $C'_{m+1}$ of $V'_1$ is equivalent to $C_{a'}$ and $C'_{m+1}$ of $V'_2$ is still equivalent to $C_b$. In the latter case, letting $C_{b'}$ be the configuration immediately after $v_{i-n}$ and $v_{j-n}$ interact at $C_b$, $C'_{m+1}$ of $V'_1$ is still equivalent to $C_a$ and $C'_{m+1}$ of $V'_2$ is equivalent to $C_{b'}$. Since $C_a$ and $C_b$ occur infinitely often in $\Xi$ and $\Xi$ is globally fair, such $C_{a'}$ and $C_{b'}$ occurs infinitely often in $\Xi$ and thus the proposition holds in the case.

Since $\# red(V) - \# blue(V) = 1$ holds after $C_h$ in $\Xi$, $\# red(V'_1) - \# blue(V'_1) = 1$ and $\# red(V'_2) - \# blue(V'_2) = 1$ holds after $C'_{2t}$ in $\Xi'$. Thus, $\# red(V') - \# blue(V') = 2$ holds after $C'_{2t}$ in $\Xi'$. Since $\Xi'$ is globally fair, this is a contradiction.

From now, by using Lemma 16, we show the theorem.

**Theorem 17.** There exists no uniform bipartition protocol with three states and designated initial states over arbitrary communication graphs assuming global fairness.

**Proof.** For the purpose of contradiction, we assume that such a protocol $Alg$ exists.

Let $S = \{s_1, s_2, s_3\}$ be a state set of agents. Without loss of generality, $f(s_1) = f(s_2) = red$ and $f(s_3) = blue$ hold. Consider a globally-fair execution $\Xi$ of $Alg$ over graph $G$ such that the number of agents is odd and no more than $P/2$. By Lemma 16, after some stable configuration $C_t$ in $\Xi$, each agent changes its state infinitely often. This implies that each agent with $s_3$ transitions to $s_1$ or $s_2$ after $C_t$. That is, each blue agent transitions to red state after $C_t$. Since $C_t$ is stable, this is a contradiction. 

\[ \square \]
4.4 Lower Bound for Symmetric Protocols under Global Fairness

In this section, we show that, with arbitrary communication graphs, designated initial states, and no base station assuming global fairness, there exists no symmetric protocol with four states. Recall that, with designated initial states and no base station, clearly any symmetric protocol never solves the problem if the number of agents \( n \) is two. Thus, we assume that \( 3 \leq n \leq P \) holds, where \( P \) is a known upper bound of the number of agents. Note that the symmetric protocol proposed in subsection 4.2 solves the problem for \( 3 \leq n \leq P \).

In this subsection, we newly define \( q \xrightleftharpoons{sym} q' \) as follow:

- For states \( q \) and \( q' \), we say \( q \xrightleftharpoons{sym} q' \) if there exists a sequence of states \( q = q_0, q_1, \ldots, q_k = q' \) such that, for any \( i (0 \leq i < k) \), transition rule \( (q_i, q_i) \rightarrow (q_{i+1}, q_{i+1}) \) exists.

Moreover, we say two agents are homonyms if they have the same state. Intuitively, \( q \xrightleftharpoons{sym} q' \) means that an agent in state \( q \) can transition to \( q' \) by only interactions with homonyms.

**Theorem 18.** There exists no symmetric protocol for the uniform bipartition with four states and designated initial states over arbitrary graph assuming global fairness when \( P \) is twelve or more.

For the purpose of contradiction, suppose that there exists such a protocol \( Alg \). Let \( R \) (resp., \( B \)) be a state set such that, for any \( s \in R \) (resp., \( s' \in B \)), \( f(s) = \text{red} \) (resp., \( f(s') = \text{blue} \)).

First, we show that the following lemma holds from Lemma 16.

**Lemma 19.** \(|R| = |B| \) holds (i.e., \(|R| = 2 \) and \(|B| = 2 \) hold).

**Proof.** Assume, for the purpose of contradiction, that \(|R| \neq |B| \) holds. Without loss of generality, assume that \(|R| = 1 \) and \(|B| = 3 \) hold (clearly \( R \neq \emptyset \) holds and thus only this combination is valid). Moreover, let \( r \) (resp., \( b_1, b_2, \) and \( b_3 \)) be a state belonging to \( R \) (resp., \( B \)).

Consider a globally-fair execution \( \Xi = C_0, C_1, C_2, \ldots \) of \( Alg \) with a communication graph \( G \) such that the number of agents \( n \) is three. From Lemma 16 after some stable configuration \( C_t \) in \( \Xi \), each agent transitions its own state to another state infinitely often. Hence, after \( C_t \), some agent with \( r \) transitions to some \( \text{blue} \) state. Since \( C_t \) is stable, this is a contradiction.

Let \( ini_r \) and \( r \) (resp., \( ini_b \) and \( b \)) be states belonging to \( R \) (resp., \( B \)). In addition, without loss of generality, assume that \( ini_r \) is the initial state of agents.

From the property of symmetry, the following lemma holds.

**Lemma 20.** Consider a symmetric transition sequence \( (ini_r, ini_r) \rightarrow (p_1, p_1), (p_1, p_1) \rightarrow (p_2, p_2), (p_2, p_2) \rightarrow (p_3, p_3), \ldots \) starting from \( ini_r \). For any \( i \), \( p_i \neq p_{i+1} \) holds.

**Proof.** For the purpose of contradiction, suppose that \( p_i = p_{i+1} \) holds for some \( i \). Consider a complete communication graph \( G = (V, E) \) such that the number of agents \( n \) is four, where \( V = \{v_1, v_2, v_3, v_4\} \).

Consider a globally-fair execution \( \Xi \) as follows:

- \( v_1 \) (resp., \( v_3 \)) interacts with \( v_2 \) (resp., \( v_4 \)) \( i \) times.

- After that, make interactions so that \( \Xi \) satisfies global fairness.

Since the initial state of agents is \( ini_r \), all agents have \( p_i \) after the \( i \) interactions. By the assumption, since \( (p_i, p_i) \rightarrow (p_i, p_i) \) holds, every agent keeps state \( p_i \) after that. Hence, \( \Xi \) cannot reach a stable configuration. Since \( \Xi \) is globally fair, this is a contradiction. 


Additionally, we can extend Lemma 20 as follow:

Lemma 21. There exists some $s_0 \in B$ such that $(ini_r, ini_r) \rightarrow (s_0, s_0)$ and $(s_0, s_0) \rightarrow (ini_r, ini_r)$ hold.

Proof. Consider a globally-fair execution $\Xi = C_0, C_1, C_2, \ldots$ of Alg with a complete communication graph $G$ such that the number of agents $n$ is six. First, consider a state $q$ such that there are two or more agents with $q$ in a stable configuration $C_t$ of $\Xi$. Observe that, for any state $q'$ such that $q \sim q'$ holds, $f(q) = f(q')$ holds. This is because, if such equation does not hold, it contradicts the definition of the stable configuration (i.e., it contradicts the fact that each agent cannot change its own color after a stable configuration). Using this fact, we show that the lemma holds.

Since the number of states is four and Lemma 20 holds, there are three possible symmetric transition sequences starting from $ini_r$ as follows:

1. For distinct states $ini_r, p_1, p_2,$ and $p_3$, there exists a transition sequence $(ini_r, ini_r) \rightarrow (p_1, p_1), (p_1, p_1) \rightarrow (p_2, p_2), (p_2, p_2) \rightarrow (p_3, p_3), (p_3, p_3) \rightarrow (x, x), \ldots,$ where $x \in \{ini_r, p_1, p_2\}$.

2. For distinct states $ini_r, p_1,$ and $p_2$, there exists a transition sequence $(ini_r, ini_r) \rightarrow (p_1, p_1), (p_1, p_1) \rightarrow (p_2, p_2), (p_2, p_2) \rightarrow (y, y), \ldots,$ where $y \in \{ini_r, p_1\}$.

3. For distinct states $ini_r$ and $p_1$, there exists a transition sequence $(ini_r, ini_r) \rightarrow (p_1, p_1), (p_1, p_1) \rightarrow (ini_r, ini_r), (ini_r, ini_r) \rightarrow (p_1, p_1), \ldots$.

Case 1: Assume, for the purpose of contradiction, that the transition sequence 1 holds. Let $p \in \{p_1, p_2, p_3\}$ be a state such that $f(p) = red$ holds. By the assumption, in a stable configuration $C_t$ of $\Xi$, #red is three. This implies that there exist two agents with $ini_r$ or $p$. If there exist two agents with $ini_r$ in $C_t$, they can transition to $p' \in \{p_1, p_2, p_3\}$ such that $f(p') = blue \neq f(ini_r)$ holds by only interactions with homonyms. Hence, by the definition of stable configurations, there exists at most one agent with $ini_r$ in $C_t$ and thus there exist two or more agents with $p$.

We consider cases of $p = p_3$, $p = p_2$, and $p = p_1$. In the case of $p = p_3$, $f(p_1) = f(p_2) = blue$ holds. By the transition sequence 1, $p$ can transition to $p_2$ by only interactions with homonyms. By the definition of stable configurations, since $f(p) \neq f(p_2)$ holds, this case does not hold. In the case of $p = p_2$, $f(p_1) = f(p_3) = blue$ holds. By the transition sequence 1, $p$ can transition to $p_3$ by only interactions with homonyms. By the definition of stable configurations, since $f(p) \neq f(p_3)$ holds, this case does not hold. In the case of $p = p_1$, $f(p_2) = f(p_3) = blue$ holds. By the transition sequence 1, $p$ can transition to $p_2$ by only interactions with homonyms. By the definition of stable configurations, since $f(p) \neq f(p_2)$ holds, this case does not hold. Thus, the transition sequence 1 does not hold.

Case 2: Assume, for the purpose of contradiction, that the transition sequence 2 holds. We show that 1) $f(p_1) = f(p_2) = blue$ holds and 2) $y = p_1$ holds. After that, from these facts, we show that 3) the transition sequence 2 does not hold.

First we show 1) $f(p_1) = f(p_2) = blue$ holds. Assume, for the purpose of contradiction, that either $p_1$ or $p_2$ is a red state and the other is a blue state (since $ini_r$ is red state, either $p_1$ or $p_2$ is blue state). In this case, since $ini_r$ can transition to blue state by only interactions with homonyms, there exist at most one agent with $ini_r$ in $C_t$ of $\Xi$. Hence, in $C_t$, there exist two red agents with $p_1$ or $p_2$. By the transition sequence 2, $p_1$(resp., $p_2$) can transition to $p_2$(resp., $p_1$) by only interactions with homonyms. Since $f(p_1) \neq f(p_2)$ holds, this contradicts the definition of stable configuration.

Next, we show 2) $y = p_1$ holds. For the purpose of contradiction, assume that $y = ini_r$ holds. In this case, $p_1$ and $p_2$ can transition to $ini_r$ by only interactions with homonyms. In addition, $f(ini_r) = red$ holds. From 1) $f(p_1) = f(p_2) = blue$, since #blue is three in a stable configuration.
Corollary 24. Consider a state set \( \text{Ini} = \{i_{in}, i_{in}b\} \). When \( s_1 \notin \text{Ini} \) or \( s_2 \notin \text{Ini} \) holds, if transition rule \( (s_1, s_2) \to (s'_1, s'_2) \) exists then \( f(s_1) = f(s'_1) \) and \( f(s_2) = f(s'_2) \) hold.

Consider a globally-fair execution \( \Xi = C_0, C_1, C_2, \ldots \) of \( \text{Alg} \) with a ring communication graph \( G = (V,E) \) such that the number of agents is three, where \( V = \{v_0, v_1, v_2\} \). In a stable configuration of \( \Xi \), either \( \#\text{blue}(V) - \#\text{red}(V) = 1 \) or \( \#\text{red}(V) - \#\text{blue}(V) = 1 \) holds.

First, consider the case of \( \#\text{blue}(V) - \#\text{red}(V) = 1 \).

By Lemma 16 red agents keep exchanging \( r \) for \( i_{in}r \) in \( \Xi \). Moreover, by Lemma 23 there exists a stable configuration in \( \Xi \) such that \( \#i_{in} \leq 1 \) holds. From these facts, there exists a stable configuration \( C_t \) of \( \Xi \) such that there exists exactly one agent that has \( i_{in}r \). Without loss of generality, we assume that the agent is \( v_0 \).

Consider the communication graph \( G' = (V', E') \) that includes four copies of \( G \). The details of \( G' \) are as follows:

Finally, we show 3) the transition sequence 2 does not hold.

Consider a complete communication graph \( G' = (V', E') \) such that the number of agents \( n \) is six, where \( V' = \{v_0, v_1, v_2, \ldots, v_5\} \). Moreover, consider a globally-fair execution \( \Xi \) as follows:

- \( \hat{v}_0, \hat{v}_1, \) and \( \hat{v}_2 \) interact with \( \hat{v}_3, \hat{v}_4, \) and \( \hat{v}_5 \) once, respectively.
- After that, make interactions so that \( \hat{\Xi} \) satisfies global fairness.

After the first item, all agents have \( p_1 \). Hence, from 1) \( f(p_1) = f(p_2) = \text{blue} \) and 2) \( y = p_1 \) (i.e., \( (p_1, p_1) \to (p_2, p_2) \) and \( (p_2, p_2) \to (p_1, p_1) \) hold), there exists transition rule \( (p_1, p_1) \to (s_{r1}, s_{r2}) \) such that \( f(s_{r1}) = f(s_{r2}) = \text{blue} \) does not hold. Since transition rules \( (p_1, p_1) \to (p_2, p_2) \) and \( (p_2, p_2) \to (p_1, p_1) \) exist and only \( p_1 \) and \( p_2 \) are \text{blue} states, a stable configuration \( C_t \) of \( \Xi \) can reach \( \hat{C}_t \) such that both \( p_1 \) and \( p_2 \) exist in \( \hat{C}_t \). Additionally, since \( (p_1, p_2) \to (s_{r1}, s_{r2}) \) exists, some agent can change its color after \( \hat{C}_t \). Since \( \hat{C}_t \) is stable, this is a contradiction. Thus, the transition sequence 2 does not hold.

Case 3: Assume, for the purpose of contradiction, that \( f(p_1) = \text{red} \) holds. Consider a globally-fair execution \( \Xi' \) of \( \text{Alg} \) with a complete communication graph \( G' = (V', E') \) such that the number of agents is six. Since transition rules \( (i_{in}, i_{in}) \to (p_1, p_1) \) and \( (p_1, p_1) \to (i_{in}, i_{in}) \) exist and \( f(i_{in}) = f(p_1) = \text{red} \) holds, there exists transition rule \( (i_{in}, p_1) \to (s_{b1}, s_{b2}) \) such that \( f(s_{b1}) = f(s_{b2}) = \text{red} \) does not hold. Since transition rules \( (i_{in}, i_{in}) \to (p_1, p_1) \) and \( (p_1, p_1) \to (i_{in}, i_{in}) \) exist and only \( i_{in} \) and \( p_1 \) are \text{red} states (and the number of \text{red} agents is three in any stable configuration), a stable configuration \( C'_t \) of \( \Xi' \) can reach \( C'_t \) such that both \( p_1 \) and \( i_{in} \) exist in \( C'_t \). In addition, since \( (i_{in}, p_1) \to (s_{b1}, s_{b2}) \) exists, some agent can change its color from \( C'_t \). Since \( C'_t \) is stable, this is a contradiction. Therefore, \( f(p_1) = \text{blue} \) holds and thus the lemma holds.

Without loss of generality, assume that \( (i_{in}, i_{in}) \to (i_{in}b, i_{in}b) \) and \( (i_{in}b, i_{in}b) \to (i_{in}, i_{in}) \) exist. For any population \( V \), we denote the number of agents with \( i_{in}r \) (resp., \( i_{in}b \)) belonging to \( V \) as \( \#i_{in}r(V) \) (resp., \( \#i_{in}b(V) \)). Moreover, let \( \#i_{in}(V) \) be the sum of \( \#i_{in}r(V) \) and \( \#i_{in}b(V) \). When \( V \) is clear from the context, we simply denote them as \( \#i_{in}r \), \( \#i_{in}b \), and \( \#i_{in} \), respectively.

From now on, we show that Theorem 18 holds if the following lemmas and corollary hold. We show the detail of these intermediate proofs later.

**Lemma 22.** There does not exist a transition rule such that \( \#i_{in} \) increases after the transition.

**Lemma 23.** Consider a globally-fair execution \( \Xi \) of \( \text{Alg} \) with some complete communication graph \( G \). After some configuration in \( \Xi \), \( \#i_{in} \leq 1 \) holds.

**Corollary 24.** Consider a state set \( \text{Ini} = \{i_{in}, i_{in}b\} \). When \( s_1 \notin \text{Ini} \) or \( s_2 \notin \text{Ini} \) holds, if transition rule \( (s_1, s_2) \to (s'_1, s'_2) \) exists then \( f(s_1) = f(s'_1) \) and \( f(s_2) = f(s'_2) \) hold.
implies that, in the same way as the case of prove the following lemmas. From now on, we prove Lemmas 22 and 23, and Corollary 24. First, from Lemmas 16 and 21, we
Proofs of Lemmas 22 and 23, and Corollary 24

An image of $G$ and $G'$ is shown in Figure 2. Consider the following execution $\Xi' = C_0', C_1', C_2', \ldots$ of $Alg$ with $G' = (V', E')$.

- For $i \leq t$, when $v_x$ interacts with $v_y$ at $C_i \rightarrow C_{i+1}$, $v'_x$ interacts with $v'_y$ at $C'_{4i} \rightarrow C'_{4i+1}$, $v'_{x+3}$ interacts with $v'_{y+3}$ at $C'_{4i+1} \rightarrow C'_{4i+2}$, $v'_{x+6}$ interacts with $v'_{y+6}$ at $C'_{4i+2} \rightarrow C'_{4i+3}$, and $v'_{x+9}$ interacts with $v'_{y+9}$ at $C'_{4i+3} \rightarrow C'_{4i+4}$.
- After $C'_{4i}$, make interactions between agents in $V'_{red}$ until agents in $V'_{red}$ converge and $\#ini(V'_{red}) \leq 1$ holds. We call the configuration $C'_t$.
- After $C'_t$, make interactions so that $\Xi'$ satisfies global fairness.

Until $C'_{4t}$, agents in $V'_1$, $V'_2$, $V'_3$, and $V'_4$ behave similarly to agents in $V$ from $C_0$ to $C_t$. This implies that, in $C'_{4t}$, every agent in $V'_{red}$ has state $ini_r$. From Lemma 23 since $ini_r$ is the initial state of agents, it is possible to make interactions between agents in $V'_{red}$ until agents in $V'_{red}$ converge and $\#ini(V'_{red}) \leq 1$ holds. Moreover, since $v_0$ is the only agent that has $ini_r$ in $C_t$, no agent in $V'_{red} \setminus V'_{red}(1 \leq i \leq 4)$ has state $ini_r$ or $ini_b$ in $C'_{4i}$. Hence, $\#ini \leq 1$ holds in $C'_{4t}$. By Corollary 24 if $\#ini \geq 2$ does not hold, no agent can change its color. Thus, since $\#ini \leq 1$ holds after $C'_{4t}$, by Lemma 22 no agent can change its color after $C'_{4t}$. Since $v_1$ and $v_2$ are blue in $C_t$, $v'_1$, $v'_2$, $v'_4$, $v'_5$, $v'_7$, $v'_8$, $v'_{10}$, and $v'_{11}$ are blue in $C'_t$. In addition, $\#blue(V'_{red}) = \#red(V'_{red})$ holds. Hence, $\#blue(V') - \#red(V') = 8$ holds. Since no agent can change its color after $C'_t$, and $\Xi'$ is globally fair, this is a contradiction.

Next, consider the case of $\#red(V) - \#blue(V) = 1$. In this case, we can prove in the same way as the case of $\#blue(V) - \#red(V) = 1$. However, in the case, we focus on $ini_b$ instead of $ini_r$. That is, we assume that agents in $V'_{red}$ (i.e., $v'_0$, $v'_3$, $v'_6$, and $v'_9$) have $ini_b$ in $C'_{4t}$. From $C'_{4t}$, we make $v'_0$ (resp., $v'_6$) interact with $v'_3$ (resp., $v'_9$) once. Then, by Lemma 21 all of them transition to $ini_r$. After that, since all agents in $V'_{red}$ have $ini_r$, we can construct an execution such that only agents in $V'_{red}$ interact and eventually $\#ini(V'_{red}) \leq 1$ holds. As a result, we can lead to contradiction in the same way as the case of $\#blue(V) - \#red(V) = 1$.

Proofs of Lemmas 22 and 23, and Corollary 24

From now on, we prove Lemmas 22 and 23 and Corollary 24. First, from Lemmas 16 and 21 we prove the following lemmas.
Lemma 25. Consider a globally-fair execution $\Xi$ of Alg with some complete communication graph $G = (V, E)$. In any stable configuration of $\Xi$, there is at most one agent with $ini_r$ and at most one agent with $ini_b$.

Proof. For the purpose of contradiction, assume that there exists a stable configuration $C_t$ of $\Xi$ such that there are more than one agent with $ini_r$ (or more than one agent with $ini_b$). Since $G = (V, E)$ is a complete communication graph, two agents with $ini_r$ (or two agents with $ini_b$) can interact. By Lemma 21, $ini_r$ (resp., $ini_b$) transitions to $ini_b$ (resp., $ini_r$). Since $f(ini_r) \neq f(ini_b)$ holds and $C_t$ is stable, this is a contradiction. □

Lemma 26. There exist transition rules $(r, r) \rightarrow (r, r)$, $(b, b) \rightarrow (b, b)$, and $(r, b) \rightarrow (r, b)$.

Proof. Consider a globally-fair execution $\Xi$ of Alg with a complete communication graph $G = (V, E)$ such that the number of agents $n$ is seven. Consider a stable configuration $C_{t_1}$ (resp., $C_{t_2}$) of $\Xi$ such that there exists exactly one agent with $ini_r$ (resp., $ini_b$) and $C_{t_1}$ (resp., $C_{t_2}$) occurs infinitely often in $\Xi$. From Lemmas 25 and 16, $C_{t_1}$ and $C_{t_2}$ exist. Note that, since the number of agents must be $P/2$ or less to use Lemma 16, $P$ must be fourteen or more. Since the number of agents $n$ is seven, there exist at least two agents with $r$ and at least two agents with $b$ in $C_{t_1}$ and $C_{t_2}$. Hence, since $G$ is a complete graph and $C_{t_1}$ and $C_{t_2}$ are stable, if $(r, r) \rightarrow (r', r')$, $(b, b) \rightarrow (b', b')$, and $(r, b) \rightarrow (r'', b'')$ exist, $f(r) = f(r') = f(r'')$ and $f(b) = f(b') = f(b'')$ hold. In addition, by Lemma 25, $r'$ and $r''$ are not $ini_r$, and $b'$ and $b''$ are not $ini_b$. Therefore, the lemma holds. □

From these lemmas, we can prove Lemma 22.

Lemma 22. There does not exist a transition rule such that $\#ini$ increases after the transition.

Proof. For the purpose of contradiction, assume that such a transition rule exists. By Lemma 26, transition rules $(r, r) \rightarrow (r, r)$, $(b, b) \rightarrow (b, b)$, and $(r, b) \rightarrow (r, b)$ exist and these transition rules do not increase $\#ini$. Hence, there exists $(rb, ini) \rightarrow (ini_1, ini_2)$ such that $rb \in \{r, b\}$, $ini \in \{ini_r, ini_b\}$, $ini_1 \in \{ini_r, ini_b\}$, and $ini_2 \in \{ini_r, ini_b\}$ hold.

Consider a globally-fair execution $\Xi$ of Alg with a complete graph $G = (V, E)$ such that the number of agents $n$ is five. Moreover, consider a stable configuration $C_{t_1}$ (resp., $C_{t_2}$) such that there exists exactly one agent with $ini_r$ (resp., $ini_b$) and $C_{t_1}$ (resp., $C_{t_2}$) occurs infinitely often in $\Xi$. From Lemmas 25 and 16, $C_{t_1}$ and $C_{t_2}$ exist. Since the number of agents $n$ is five, there exists at least one agent with $r$ and at least one agent with $b$ in $C_{t_1}$ and $C_{t_2}$. This implies that, an agent with $ini_r$ (resp., $ini_b$) can interact with an agent with $r$ in $C_{t_1}$ (resp., $C_{t_2}$). Similarly, an agent with $ini_r$ (resp., $ini_b$) can interact with an agent with $b$ in $C_{t_1}$ (resp., $C_{t_2}$). Moreover, since $G$ is a complete graph and $\Xi$ is globally fair, those interactions happen infinitely often.

First, we consider the case that $ini = ini_r$ and $rb = r$ hold. Consider an interaction between an agent with $ini_r$ and an agent with $r$ in $C_{t_1}$. Since any agent cannot change its color in a stable configuration, both agents transition to $ini_r$ by the interaction. However, by Lemma 25, two agents cannot have $ini_r$ in any stable configuration. This is a contradiction. Thus, $ini = ini_r$ and $rb = r$ do not hold. In a similar way, $ini = ini_b$ and $rb = b$ do not hold.

Next, we consider the case that $ini = ini_b$ and $rb = b$ hold. Consider an interaction between an agent with $ini_b$ and an agent with $b$ in $C_{t_1}$. Let $C_{t_1}'$ be configuration that can be obtained from $C_{t_1}$ by the interaction. Since any agent cannot change its color after a stable configuration, one $ini_b$ and one $ini_b$ occur by the interaction. Moreover, by Lemma 25, there exist exactly one $ini_r$ and exactly one $ini_b$ in $C_{t_1}'$. This implies that, since $C_{t_1}'$ is stable and $n$ is five, there exists at least one agent with $b$ and thus an agent with $ini_r$ can interact with an agent with $b$ in $C_{t_1}'$. Hence, we can obtain $C_{t_1}''$ from $C_{t_1}'$ by making interaction between an agent with $ini_r$ and an agent with $b$. □
Clearly, in $C_t$, there exist two agents with $ini_b$. This contradicts Lemma 25 and thus $ini = ini_r$ and $rb = b$ do not hold. In a similar way, we can prove that $ini = ini_b$ and $rb = r$ do not hold. Hence, for any $rb \in \{r, b\}$ and $ini \in \{ini_r, ini_b\}$, transition rule $(rb, ini) \rightarrow (ini_1, ini_2)$ with $ini_1$, $ini_2 \in \{ini_r, ini_b\}$ does not exist. Consequently the lemma holds.

By the existence of $(ini_r, ini_r) \rightarrow (ini_b, ini_b)$ and $(ini_b, ini_b) \rightarrow (ini_r, ini_r)$, we can prove the following lemma.

**Lemma 27.** There exists a transition rule $(ini_r, ini_b) \rightarrow (x, y)$ such that $x \in \{r, b\}$ or $y \in \{r, b\}$ holds.

**Proof.** For the purpose of contradiction, assume that, if there is $(ini_r, ini_b) \rightarrow (x, y), x \in \{ini_r, ini_b\}$ and $y \in \{ini_r, ini_b\}$ hold.

Consider a globally-fair execution $\Xi$ of Alg with a complete communication graph $G = (V, E)$ such that the number of agents is three. By Lemma 25, there exists at most one agent with $ini_r$ and at most one agent with $ini_b$ in a stable configuration of $\Xi$. However, by the assumption and the existence of $(ini_r, ini_r) \rightarrow (ini_b, ini_b)$ and $(ini_b, ini_b) \rightarrow (ini_r, ini_r)$, all agents have $ini_r$ or $ini_b$ permanently in $\Xi$. This is a contradiction.

By Lemmas 22, 25 and 27, we show the proof of Lemma 23.

**Lemma 23.** Consider a globally-fair execution $\Xi$ of Alg with some complete communication graph $G$. After some configuration in $\Xi$, $\#ini \leq 1$ holds.

**Proof.** Consider $C_t$ such that $\#ini \leq 2$ holds in $C_t$ and $C_t$ occurs infinitely often in $\Xi$. By Lemma 25, such a configuration exists. Moreover, by Lemma 22, $\#ini \leq 2$ holds even after $C_t$. First, consider the case that $\#ini \leq 1$ holds in $C_t$. By Lemma 22, $\#ini \leq 1$ holds after $C_t$ and thus the lemma holds immediately in this case. Next, consider the case that $\#ini = 2$ holds in $C_t$. By Lemma 25 in $C_t$, there exists an agent $v_1$ (resp., $v_2$) with $ini_r$ (resp., $ini_b$). By Lemma 27, when $v_1$ interacts with $v_2$ at $C_t \rightarrow C_{t+1}$, $\#ini \leq 1$ holds in $C_{t+1}$. By global fairness, $C_{t+1}$ occurs infinitely often in $\Xi$. By Lemma 22, $\#ini \leq 1$ holds after $C_{t+1}$ and thus the lemma holds.

From Lemmas 23 and 16, we can obtain the following lemma.

**Lemma 28.** Let $ini$ and $rb$ be states such that $ini \in \{ini_r, ini_b\}$ and $rb \in \{r, b\}$ hold. If $(ini, rb) \rightarrow (x, y)$ exists, $f(ini) = f(x)$ and $f(rb) = f(y)$ hold.

**Proof.** Assume, for the purpose of contradiction, that there exists $(ini, rb) \rightarrow (x, y)$ such that $f(ini) \neq f(x)$ or $f(rb) \neq f(y)$ holds.

Consider a globally-fair execution $\Xi$ of Alg with a complete communication graph $G = (V, E)$ such that the number of agents $n$ is five. By Lemmas 23 and 16, there exists a stable configuration $C_{t_1}$ (resp., $C_{t_2}$) of $\Xi$ such that there is $v_r$ (resp., $v_b$) which is the only agent with $ini_r$ (resp., $ini_b$) in $C_{t_1}$ (resp., $C_{t_2}$) and there is $v_{rb}$ with $rb$. By the assumption, when $v_r$ (resp., $v_b$) interacts with $v_{rb}$ at $C_t \rightarrow C_{t+1}$ (resp., $C_{t_2} \rightarrow C_{t_2+1}$), $v_r$ (resp., $v_b$) or $v_{rb}$ changes its color. Since $C_{t_1}$ and $C_{t_2}$ are stable, this is a contradiction.

From Lemmas 26 and 28, we can obtain the following corollary.

**Corollary 24.** Consider a state set $Ini = \{ini_r, ini_b\}$. When $s_1 \notin Ini$ or $s_2 \notin Ini$ holds, if transition rule $(s_1, s_2) \rightarrow (s_1', s_2')$ exists then $f(s_1) = f(s_1')$ and $f(s_2) = f(s_2')$ hold.
4.5 Impossibility under Weak Fairness

In this subsection, assuming arbitrary communication graphs and designated initial states and no base station, we show that there is no protocol that solves the problem under weak fairness. Fischer and Jiang [21] proved the impossibility of leader election for a ring communication graph. We borrow their proof technique and apply it to the impossibility proof of a uniform bipartition problem.

**Theorem 29.** There exists no protocol that solves the uniform bipartition problem with designated initial states and no base station under weak fairness assuming arbitrary communication graphs.

**Proof.** For the purpose of contradiction, let us assume that there exists such a protocol $Alg$.

First, consider a ring $R_1$ with three agents $v_0$, $v_1$, and $v_2$. Let $(v_0, v_1), (v_1, v_2)$, and $(v_2, v_0)$ be the edges of $R_1$. Furthermore, let $Ξ = C_0, C_1, C_2, \ldots, C_t, \ldots$ be an execution of $Alg$, where $C_t$ is a stable configuration. Without loss of generality, we assume that $#red = 1$ and $#blue = 2$ hold in $C_t$.

Next, consider a ring $R_2$ with six agents such that two copies of $R_1$ are combined to form $R_2$. Let $v'_0, v'_1, v'_2, v'_3, v'_4,$ and $v'_5$ be agents of $R_2$, and let $(v'_0, v'_1), (v'_1, v'_3), (v'_3, v'_4), (v'_4, v'_5),$ and $(v'_2, v'_0)$ be edges of $R_2$ (see Figure 3).

Now, let us construct the following execution $Ξ' = D_0, D'_0, D_1, D'_1, \ldots$

- For $x$ and $y$ such that either $x = 0$ or $y = 0$ holds, when $v_x$ interact with $v_y$ at $C_i \rightarrow C_{i+1}$, $v'_x$ interacts with $v'_y$ at $D_i \rightarrow D'_i$, and $v'_{x+3}$ interacts with $v'_{y+3}$ at $D'_i \rightarrow D_{i+1}$.

- When $v_1$ interacts with $v_2$ at $C_i \rightarrow C_{i+1}$, $v'_1$ interacts with $v'_5$ at $D_i \rightarrow D'_i$, and $v'_4$ interacts with $v'_2$ at $D'_i \rightarrow D_{i+1}$.

If configurations $C$ of $R_1$ and $D$ of $R_2$ satisfy the following condition, we say that those configurations are equivalent.

- For $i$ ($0 \leq i \leq 2$), $s(v_i, C) = s(v'_i, D) = s(v'_{i+3}, D)$ holds.

From now on, by induction on the index of configuration, we show that $C_r$ and $D_r$ are equivalent for any $r \geq 0$. Clearly $C_0$ and $D_0$ are equivalent, so the base case holds immediately. For the induction step, we assume that $C_t$ and $D_t$ are equivalent, and then consider two cases of interaction at $C_{t+1}$.

First we consider the case that, for $x$ and $y$ such that either $x = 0$ or $y = 0$ holds, agents $v_x$ and $v_y$ interact at $C_t \rightarrow C_{t+1}$. In this case, at $D_t \rightarrow D'_t$, $v'_x$ interacts with $v'_y$ and, at $D'_t \rightarrow D_{t+1}$, $v'_{x+3}$ interacts with $v'_{y+3}$. By the induction assumption, $s(v_x, C_t) = s(v'_x, D_t) = s(v'_{x+3}, D_t)$ and $s(v_y, C_t) = s(v'_y, D_t) = s(v'_{y+3}, D_t)$ hold. Thus, agents $v'_x$ and $v'_{x+3}$ change their state similarly to $v_x$, and agents $v'_y$ and $v'_{y+3}$ change their state similarly to $v_y$. Hence, $C_{t+1}$ and $D_{t+1}$ are equivalent in this case.

Next, we consider the case that $v_1$ and $v_2$ interact at $C_t \rightarrow C_{t+1}$. In this case, at $D_t \rightarrow D'_t$, $v'_1$ interacts with $v'_5$ and, at $D'_t \rightarrow D_{t+1}$, $v'_4$ interacts with $v'_2$. By the induction assumption,
s(v_1, C_l) = s(v_1', D_l) = s(v_2, C_l) = s(v_2', D_l) hold. Thus, agents v_1' and v_4' change their state similarly to v_1, and agents v_2' and v_5' change their state similarly to v_2. Hence, C_{l+1} and D_{l+1} are equivalent in this case.

Thus, C_r and D_r are equivalent for any r ≥ 0.

This implies that, after D_l, #red = 2 and #blue = 4 hold. Moreover, since Ξ is weakly fair, clearly Ξ' is weakly fair. This is a contradiction. □

5 Concluding Remarks

In this paper, we consider the uniform bipartition problem with designated initial states assuming arbitrary communication graphs. We clarified the problem solvability, and even provided tight bounds (with respect to the number of states per agent) in the case of global fairness. Concretely, with no base station under global fairness, we proved that four is the minimum number of states per agent to enable asymmetric protocols, and five is the minimum number of states per agent to enable symmetric protocols. With no base station under weak fairness, we proved the impossibility to obtain an asymmetric protocol. On the other hand, with a base station, we propose a symmetric protocol with 3P + 1 states under weak fairness, and a symmetric protocol with three states under global fairness.

Our work raises interesting open problems:

- Is there a relation between the uniform bipartition problem and other classical problems such as counting, leader election, and majority? We pointed out the reuse of some proof arguments, but the existence of a more systematic approach is intriguing.

- What is the time complexity of the uniform bipartition problem?

- Is the uniform bipartition problem in arbitrary communication graphs with arbitrary initial states feasible? In case the answer is yes, what is the space complexity?

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] 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, Michael J Fischer, and Hong Jiang. Self-stabilizing population protocols. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(4):13, 2008.

[8] 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.

[9] Joffroy Beauquier, Peva Blanchard, and Janna Burman. Self-stabilizing leader election in population protocols over arbitrary communication graphs. In *International Conference on Principles of Distributed Systems*, pages 38–52. Springer, 2013.

[10] 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.

[11] 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.

[12] Stav Ben-Nun, Tsvi Kopelowitz, Matan Kraus, and Ely Porat. An $o(\log^{3/2} n)$ parallel time population protocol for majority with $o(\log n)$ states. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 191–199, 2020.

[13] Petra Berenbrink, Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Peter Kling, and Tomasz Radzik. A population protocol for exact majority with $o(\log 5/3 n)$ stabilization time and $\Theta(\log n)$ states. In *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, volume 121 of LIPIcs, pages 10:1–10:18, 2018.

[14] Petra Berenbrink, Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Peter Kling, and Tomasz Radzik. Time-space trade-offs in population protocols for the majority problem. *Distributed Computing*, pages 1–21, 2020.

[15] Petra Berenbrink, George Giakkoupis, and Peter Kling. Optimal time and space leader election in population protocols. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 119–129, 2020.

[16] Petra Berenbrink, Dominik Kaaser, and Tomasz Radzik. On counting the population size. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 43–52, 2019.

[17] Olivier Bournez, Jérémie Chalopin, Johanne Cohen, Xavier Koegler, and Mikael Rabie. Population protocols that correspond to symmetric games. *International Journal of Unconventional Computing*, 9, 2013.

[18] Davide Canepa and Maria Gradinariu Potop-Butucaru. Self-stabilizing tiny interaction protocols. In *Proceedings of the Third International Workshop on Reliability, Availability, and Security*, page 10. ACM, 2010.
[19] 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.

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

[21] Michael Fischer and Hong Jiang. Self-stabilizing leader election in networks of finite-state anonymous agents. In *International Conference On Principles Of Distributed Systems*, pages 395–409. Springer, 2006.

[22] Leszek Gąsieniec, 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.

[23] Leszek Gąsieniec, Grzegorz Stachowiak, and Przemyslaw Uznanski. Almost logarithmic-time space optimal leader election in population protocols. In *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures*, pages 93–102. ACM, 2019.

[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] Adrian Kosowski and Przemyslaw Uznanski. Brief announcement: Population protocols are fast. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 475–477, 2018.

[26] 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.

[27] George B Mertzios, Sotiris E Nikoletseas, Christoforos L Raptopoulos, and Paul G Spirakis. Determining majority in networks with local interactions and very small local memory. In *International Colloquium on Automata, Languages, and Programming*, pages 871–882. Springer, 2014.

[28] 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.

[29] Yuichi Sudo, Fukuhito Ooshita, Taisuke Izumi, Hirotugu Kakugawa, and Toshimitsu Masuzawa. Time-optimal leader election in population protocols. *IEEE Transactions on Parallel and Distributed Systems*, 2020.

[30] Yuichi Sudo, Fukuhito Ooshita, Hirotugu Kakugawa, and Toshimitsu Masuzawa. Loosely-stabilizing leader election on arbitrary graphs in population protocols. In *International Conference on Principles of Distributed Systems*, pages 339–354. Springer, 2014.

[31] Tomoki Umino, Naoki Kitamura, and Taisuke Izumi. Differentiation in population protocols. *6th workshop on biological distributed algorithms(BDA)*, 2018.
[32] 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.

[33] Hiroto Yasumi, Fukuhito Ooshita, and Michiko Inoue. Uniform partition in population protocol model under weak fairness. *the 23rd International Conference on Principles of Distributed Systems*, 2019.

[34] 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.

[35] 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.