Randomized Byzantine Gathering in Rings*

John Augustine
Indian Institute of Technology Madras, India

Arnhav Datar
Indian Institute of Technology Madras, India
Carnegie Mellon University, Pittsburgh, PA, USA

Nischith Shadagopan
Indian Institute of Technology Madras, India

Abstract

We study the problem of gathering k anonymous mobile agents on a ring with n nodes. Importantly, f out of the k anonymous agents are Byzantine. The agents operate synchronously and in an autonomous fashion. In each round, each agent can communicate with other agents co-located with it by broadcasting a message. After receiving all the messages, each agent decides to either move to a neighbouring node or stay put. We begin with the k agents placed arbitrarily on the ring, and the task is to gather all the good agents in a single node. The task is made harder by the presence of Byzantine agents, which are controlled by a single Byzantine adversary. Byzantine agents can deviate arbitrarily from the protocol. The Byzantine adversary is computationally unbounded. Additionally, the Byzantine adversary is adaptive in the sense that it can capitalize on information gained over time (including the current round) to choreograph the actions of Byzantine agents. Specifically, the entire state of the system, which includes messages sent by all the agents and any random bits generated by the agents, is known to the Byzantine adversary before all the agents move. Thus the Byzantine adversary can compute the positioning of good agents across the ring and choreograph the movement of Byzantine agents accordingly. Moreover, we consider two settings: standard and *visual tracking* setting. With visual tracking, agents have the ability to track other agents that are moving along with them. In the standard setting, agents do not have such an ability.

In the standard setting we can achieve gathering in $\mathcal{O}(n \log n \log k)$ rounds with high probability[1] and can handle $\mathcal{O}\left(\frac{k}{\log k}\right)$ number of Byzantine agents. With visual tracking, we can achieve gathering faster in $\mathcal{O}(n \log n)$ rounds whp and can handle any constant fraction of the total number of agents being Byzantine.

2012 ACM Subject Classification Computing methodologies → Self-organization; Theory of computation → Self-organization

Keywords and phrases Mobile agents and robots

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2022.13

Funding *John Augustine*: Supported by the Cybersecurity Research Centre under the IIT Madras Institute of Eminence scheme.
Nischith Shadagopan: Supported by the Young Research Fellowship 2020, IIT Madras.

1 Introduction

Swarm robotics envisage swarms of mobile robots or agents as they are sometimes called, self-organizing and collaborating to achieve shared goals. These smaller agents have various advantages such as robustness, scalability, and efficiency [16, 20]. Significant research has been

* Author names are listed alphabetically. All authors contributed equally to this work.
[1] Throughout this paper,"with high probability" or "whp" in short means with probability at least $1-1/n^c$ for some constant $c \geq 1$

26th International Conference on Principles of Distributed Systems (OPODIS 2022).
Editors: Eshcar Hillel, Roberto Palmieri, and Etienne Rivière; Article No. 13; pp. 13:1–13:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

conducted in the areas of gathering [1, 21], flocking [23], scattering [30], and exploring [11, 25]. Some research has also been conducted considering that the agents are deployed in adverse environments [12].

We focus on the problem of gathering in a ring against Byzantine agents. Although a ring is a simple structure, the nodes are all symmetric. It is therefore no surprise that fundamental symmetry-breaking problems like leader election [19] were first studied in the context of rings with insights then influencing the development of more general algorithms. It is quite easy to note that $\Omega(n)$ rounds are required for gathering any number of agents in a ring. Our goal is to design algorithms with running times that are close to linear in n.

Initially, k anonymous agents are positioned arbitrarily across the ring. Notably, f out of the k agents are Byzantine. The task is to gather all the good agents in a single node of the ring. The agents cannot distinguish between the nodes of the graph. Each agent can communicate with other agents in the same node through a local broadcast. The agents operate synchronously and can either stay or move to a neighbouring node in each round. A single adversary controls the Byzantine agents. The adversary has complete knowledge of the graph and the states of all the agents. The adversary can deviate arbitrarily from the protocol to prevent gathering. The adversary is also adaptive in the sense that it can make decisions based on continual learning. Further, we consider the visual tracking setting where agents have the ability to track other agents moving along with them.

1.1 Related works

Gathering of mobile agents with unique IDs in the presence of Byzantine agents was first considered by Dieudonné et al. [13]. They focus on finding the minimum number of good agents required to gather successfully. They prove that at least $k = 2f + 1$ agents are required to achieve gathering and also show that it is not possible to deterministically gather $k = 2f$ agents in a ring of known size. They also provide an algorithm to gather $k \geq 3f + 1$ agents in a network of known size. Bouchard et al. [6] provide a deterministic algorithm for gathering $k = 2f + 1$ agents, which is the minimum number of agents required to gather when the size of the network is known. They also prove that $k = 2f + 2$ agents are required to gather when the size of the network is unknown, a slight increase from before. Regrettably, both the aforementioned algorithms from [13, 6] have exponential running times and are hence not practical. Thereafter, Bouchard et al. [7] improved the running time by providing a polynomial-time algorithm that uses a piece of global knowledge of size $\mathcal{O}(\log \log \log n)$ and achieves gathering when $k \geq 5f^2 + 6f + 2$. Hirose et al. [18] provided polynomial-time algorithms with a similar number of Byzantine agents but consider the case when the Byzantine agents cannot change their ID. Their algorithms run in $\mathcal{O}((f + \Lambda_{all}) \cdot X(n))$ time, where Λ_{all} represents the length of the maximum ID of all agents and $X(n)$ is the number of rounds to explore a network of size n.

Sudo et al. [31] introduce a new communication model where each node contains a *whiteboard* where agents can leave information. Gathering in this model is trivial. Each agent can leave its ID in its starting node, and the agents can move to the node containing the smallest ID to achieve gathering in $\mathcal{O}(m)$ time, where m is the number of edges. Tsuchida et al. [33] extend this by allowing each node to have an *authenticated whiteboard*. Authenticated whiteboards allow agents to store information along with their signatures. Their algorithm deterministically achieves gathering in $\mathcal{O}(fm)$ time. But authenticated whiteboards is quite an advanced feature when considering mobile agents with basic functionality.

The (non-Byzantine) gathering problem has been researched extensively in rings [9, 22, 10, 24]. Klasing et al. [24] provided deterministic algorithms for rings in an asynchronous model by only using global weak-multiplicity detection. Izumi et al. [22] provided an optimal

$\mathcal{O}(n)$ time algorithm for rings for some configurations when the agents only had local weak-multiplicity detection. Subsequently, D'Angelo et al. [9, 10] designed gathering algorithms for rings using global or local-weak multiplicity detection, provided the agents can take a snapshot of the ring at all occupied nodes.

Research on randomized algorithms for gathering and other search problems has been ongoing for a few years now and is summarised by Alpern and Gal [3]. Alpern et al. [2] provided an algorithm for rendezvous of two agents in a ring that runs in expected $\mathcal{O}(n)$ rounds. Ooshita et al. [28] consider the problem of gathering agents in an anonymous unidirectional ring under the constraint the agents are unaware of the number of agents and the number of nodes. Furthermore, their communication model employs whiteboards. They prove that there cannot exist randomized algorithms for this problem with termination detection. In our model, the agents can move in both directions and are aware of the number of nodes. Therefore their results do not apply in our context. A relevant result in our context is given by Cooper et al. [8] who introduce coalescing random walks. Here, particles perform independent random walks on the graph. Whenever two particles meet at a node, they combine and continue the random walk. The paper explores the time required to combine all the particles in the graph with such an algorithm. This algorithm can be trivially adapted to our model. The combining of particles is enabled in our context by co-located agents generating common random bits. This allows co-located agents to continue the random walk together. Such an algorithm takes $\tilde{\mathcal{O}}(n^2)$ time in expectation to gather all the agents in a ring, which is quite slow. Notably, this algorithm can handle any number of Byzantine agents. Eguchi et al. [14] provide some results on fast randomized gathering but consider the case when there are only two agents which are placed on adjacent nodes initially.

With this paper, we aim to kick off research on fast randomized algorithms with anonymous agents despite the presence of Byzantine agents. Both our results in this paper are log factors away from being linear in n. Our algorithms rely on the vital ability of agents to produce uniform and independent random bits. This integrated with synchronous operation of agents and local broadcast-based model of communication gives co-located agents the ability to generate common random bits. This can be done in the following way

1. Each agent generates a uniform and independently random string of s bits
2. Each agent broadcasts this string to other co-located agents, including itself
3. Each agent computes the XOR of all the strings received.

This procedure allows co-located agents to generate a common random string of arbitrary length as long as there is at least one good agent. We believe that this is a reasonable abstraction for the following two reasons. Firstly, this is an important abstraction which allows us to provide more scalable, faster and more secure (resilience to Byzantine agents) algorithms. Secondly, such an abstraction is easy to realise using existing technological capabilities. At a physical level, there are several results that allow agents to arrange themselves in the form of a circle [34, 17] or other patterns [32]. Once the agents arrange themselves in a fitting pattern, they can be forced to simultaneously broadcast their random bit through physical means like lights. Algorithmically also, there are several results which enable common coin tossing [5, 29, 26, 4, 15]. Regardless of the technology that is employed, we believe that common random bits will be beneficial for a range of robot coordination problems.

1.2 Our Model

We consider the problem of gathering k agents placed arbitrarily on a ring $G = (V, E)$. We denote the number of nodes in the graph G by $n = |V|$. The nodes of the graph do not have any facility to compute, store or communicate any information. The nodes of the graph can be viewed as just a container for agents. Each node $v \in V$ in G has degree 2 and has two labelled ports corresponding to each incident edge. This port labelling is common to all agents. An edge $e = (u, v) \in E$ indicates that an agent can move bidirectionally between node u and node v in one step. The agents cannot distinguish between the nodes of the ring. The agents also do not have a common clockwise/counterclockwise orientation. An agent can fix a particular orientation and remember the number of steps taken to keep track of its position relative to some node.

The agents operate synchronously. Each agent has the ability to compute and store information. Each agent can also communicate with all other co-located agents using local broadcast. In the visual tracking setting, the agents also have the ability to identify other agents that are co-located for at least two consecutive rounds. The agents are aware of the number of nodes n. In the standard setting, the agents are also aware of an upper-bound on the number of Byzantine agents f. The agents have a fixed orientation of direction (clockwise/counterclockwise) but this orientation is not common across all the agents.

Computation is defined by a finite state machine coupled with the ability to generate random bits and communicate messages. At each round $r > 0$, each good agent:

1. broadcasts (to all other co-located agents) a message that is a function of its state and includes public random bits,
2. receives messages broadcasted by other agents in its current node,
3. transitions to a new state as a function of the current state, messages received, and random bits,
4. chooses to either stay in the current location or move through one of the ports (based on the chosen state),
5. and finally updates the state with the direction (clockwise/counterclockwise) in which it moved.

All agents must broadcast a message at the start of each round. Note that this is not a limitation because agents can send an empty $\perp$ message if they have nothing to send.

f of the k anonymous agents are Byzantine, and all these agents are controlled by a single adversary. The remaining $k - f$ agents are called *good agents*. The adversary is computationally unbounded and can deviate arbitrarily from the protocol. The adversary can distinguish between the nodes of G and is also aware of the starting positions of all the good agents. Additionally, the adversary is strongly adaptive in the sense that it can capitalize on information gained over time (including the current round) to choreograph the actions of Byzantine agents. Specifically, the entire state of the system, which includes messages sent by all the agents and any random bits generated by the agents, is known to the adversary at the beginning of the subsequent step in the round. Thus the adversary can compute the positioning of good agents across the ring and choreograph the movement of Byzantine agents accordingly.

1.3 Our Techniques and Contributions

Being sparse and symmetric, rings present a significant challenge. We present algorithms for two settings:

(a) the standard setting and

(b) the setting where agents have visual tracking capability.

In the standard setting we can achieve gathering in $\mathcal{O}(n \log n \log k)$ rounds and can handle $\mathcal{O}\left(\frac{k}{\log k}\right)$ number of Byzantine agents. The algorithm splits the groups into leaders and followers. The leader groups just go around the ring and the follower group merges with the

Table 1 Known results on randomized gathering protocols including those inferred from the literature on coalescing random walks. The $\tilde{\mathcal{O}}$ notation is used to hide polylog(n) factors.

Setting	Time Complexity	Max. Byz. Agents
Coalescing random walks [8]	$\tilde{\mathcal{O}}(n^2)$	$f < k$
With No Byzantine Agents	$\mathcal{O}(n)$ (on exp.)	$f = 0$
Standard setting	$\mathcal{O}(n \log n \log k)$	$f(35 + 34\log_2(f+1)) + 2 < k$
With visual tracking	$\mathcal{O}(n \log n)$	$f = \alpha k$, $0 \leq \alpha < 1$
Lower bound	$\Omega(n)$	$f = 0$

first leader group that it meets. With visual tracking capability, i.e., the ability to identify which co-located agents were also co-located with them in the previous round, we can achieve gathering faster in $\mathcal{O}(n \log n)$ rounds as long as the number of Byzantine agents is some constant fraction of the total number of agents. The difference here is that the follower groups make note of all the leader groups that it meets and chooses one of them to merge with probability proportional to the leader group's size. We can see that both these algorithms follow a common paradigm of splitting into leaders and followers. In the former algorithm, the follower groups merge with the first leader group that they meet but in the latter they take their time to evaluate all the leader groups and then decide to merge with one of them.

The summary of our results can be found in Table 1.

2 Algorithms

We present three algorithms. The first is a warm up that works only when all agents are good and requires an expected $\mathcal{O}(n)$ rounds for all agents to gather. The second algorithm is for the *standard gathering* of anonymous agents setting wherein agents are indistinguishable from round to round. This algorithm takes $\mathcal{O}(n \log n \log k)$ rounds and is resilient against up to a suitable $\mathcal{O}(k/\log k)$ Byzantine agents. Subsequently, we study the variant where agents can *visually track* others that are moving along with them and provide an algorithm that can handle an $f < k$ Byzantine failures. For simplicity we first present our algorithms assuming that all the agents have a common orientation of the ring and discuss briefly how to adapt them to the situation where there is no common orientation.

2.1 Warmup: No Byzantine Agents

We now present a simple warmup gathering algorithm inspired by Alpern et al. [2]. Their work is limited to gathering two agents in a ring whereas ours is more general. Our algorithm runs in expected $\mathcal{O}(1)$ stages. We use the term committee to refer to a set of agents; members of the committee know they are in the committee and those not in the committee know that they are not. A committee is good if it is neither empty nor the full set of agents, i.e., there must be at least one agent in the committee and one that isn't. At each stage, the agents perform the following steps.

Step 1: Committee Election. The agents attempt a committee election that will result in a good committee members with probability at least a constant. We achieve this simply by allowing each agent to self-sample itself with probability $1/k$.

▶ **Lemma 1.** *Assuming $k \geq 2$, the elected committee will be good with probability at least a constant.*

Proof. The two bad events are either that the committee is empty (which can occur with probability $p_1 = (1 - 1/k)^k$) or that all agents are in the committee (which can occur with probability at most $p_2 = (1/k)^k$. Since $1 - p_1 - p_2$ is bounded from below by a constant, the lemma holds. ◀

The elected committee members furthermore choose a random ID – a bit string of a suitable length that is at least $\Omega(\log k)$. This will ensure that the IDs are unique with constant probability.

Step 2: Leader Election. The committee members go around the ring twice in a direction of their choosing. As they go around, they broadcast their ID in all nodes along the way. The agents on the other hand broadcast the smallest ID they have learned about thus far (with a special symbol say $\perp$ dedicated for the case where they have not yet learned an ID). Clearly, once the committee members go around twice, all agents will know the smallest ID (i.e., the leader) assuming the committee is good and committee members have distinct IDs (both occurring with constant probability).

Step 3: Gathering. In this step, the leader stays put for n rounds (but keeps broadcasting its ID). All other agents go around the ring (in a direction of their choosing) and stop when they reach the leader.

Step 4: Verification. If all k agents gather, the procedure stops. Otherwise, we repeat from step 1 onward.

The following theorem follows immediately.

▶ **Theorem 2.** *As long as $k \geq 2$ and $n \geq 1$, all k agents will gather in expected $\mathcal{O}(n)$ rounds.*

We conclude this warmup with a few observations. Firstly, we observe that the algorithm does not require common coin tosses amongst agents within a node. Individual random bits generated by agents suffice. Secondly, we note that it is a Las Vegas algorithm that will terminate only when all agents have gathered (and are aware that gathering has terminated). However, in situations where we prefer that the algorithm runs for a predetermined number of rounds, we can run $\Theta(\log 1/\delta)$ stages and obtain the guarantee that the algorithm gathers all agents with probability at least $1 - \delta$. Finally, if $k \in \Omega(\log n)$ is sufficiently large, we can elect a good committee with high probability in one shot. Each agent can self-sample itself into the committee with probability $p = \Theta(\log n/k)$. Such a self-sampled committee can be shown to be good whp by applying a standard Chernoff bound (for example, Corollary 4.6 from [27]). Thus, the whole algorithm can complete in one stage whp.

2.2 Standard Gathering

The algorithm runs in $\mathcal{O}(\log k)$ stages with stages numbered from $j = 0$ to $\lceil \log_2(f + 1) \rceil - 1$. Each stage has $\lceil 2 \log_{8/7} n \rceil$ phases. At the starting of each stage, all groups below a threshold of 2^j become inactive for this stage. Inactive agents don't do anything for that stage. In each phase, each group decides to be either a leader or a follower (with probability 1/2 each). The role of the leader group is quite simple, it just moves clockwise for $\lfloor \frac{n}{2} \rfloor$ rounds (out of which one round is set aside for randomizing the parity of its position). The follower group moves counterclockwise till it meets a leader group having number of agents greater than a threshold of 2^j for the first time (or up to $\lfloor \frac{n}{2} \rfloor$ rounds, whichever is sooner). After meeting such a leader group, the follower group starts moving in the clockwise direction and behaves exactly like a leader group till the phase ends. The last stage is similar to the earlier stages except all agents are active. Algorithm 1 presents the pseudocode.

Algorithm 1 Gathering in standard setting.

function LEADER
  random ← 1 with probability $\frac{1}{2}$ ▷ This decision is made as a group
  Move one step clockwise if random = 1 ▷ Randomize the parity of the group's position
  Move one step clockwise and broadcast "I am a leader" message for $\lfloor \frac{n}{2} \rfloor$ -1 rounds
end function
function FOLLOWER(j)
  flag ← 1
  for $\lfloor \frac{n}{2} \rfloor$ rounds **do**
    If flag = 1, go one step counterclockwise **else** go one step clockwise and broadcast "I am a leader" message
    Count the number d of leader agents in the node based on the number of "I am a leader" messages received
    if d $\geq 2^j$ and flag = 1 **then** ▷ 2^j is the threshold
      flag ← 0 ▷ At this point, the follower group has essentially coalesced with the leader group
    end if
  end for
end function
for $j = 0$ to $\lceil \log_2(f+1) \rceil - 1$ stages **do** ▷ If f is not known, an upper bound can be used
  Each good agent forms a group with other agents co-located with it.
  if the number of agents in the group $\geq 2^j$ **then** ▷ These are active agents
    for $\lceil 2 \log_{8/7} n \rceil$ phases **do**
      Each good agent forms a group with other agents co-located with it.
      Decide to be a leader group or follower group with equal probability ▷ this decision is made as a group
      Leaders call LEADER(), followers call FOLLOWER(j)
    end for
  end if
end for
for $\lceil 2 \log_{8/7} n \rceil$ phases **do** ▷ last stage
  Each good agent forms a group with other agents co-located with it.
  Decide to be a leader group or follower group with equal probability ▷ this decision is made as a group
  Leaders call LEADER(), followers call FOLLOWER($\lceil \log_2(\mathsf{f}+1) \rceil$)
end for

We define a *good group* as a group having at least one good agent. Define all the groups that have number of agents ≥ the threshold at the start of a stage as *active groups*. Groups below the threshold are said to be *inactive*. We can classify active groups into two types : *true* and *fake*. True active groups have number of good agents greater than or equal to the threshold whereas fake active groups have number of good agents less than the threshold. An agent is said to be *active* if it is part of an active group and otherwise called an *inactive agent*. Two groups are said to be of the same *parity* if in that phase the number of edges between the nodes in which the two groups are present, is even. In this setup, it is important to note that a leader group meets a follower group only if they are of same parity. This is because if the number of edges between them is odd, then they will cross each other while going across an edge and not realise this. Note that in this algorithm once two good agents become members of the same group, they will never separate. So the number of good groups never increases.

There is not much use for the Byzantine agents to behave as a follower group because the leader groups are quite simple and are not affected by follower groups. But the Byzantine agents can form a leader group and then present itself to a follower group. Then the follower group thinks that it has successfully merged with another group.

Suppose a set of Byzantine agents get together and form a leader group b_1. Suppose they meet an active follower group g_1. Group g_1 will think that it has merged with another group and will start behaving like a leader group for the rest of the stage. Group b_1 is now free to do whatever it wants. Group b_1 cannot catch up to any follower group by moving in the counterclockwise direction, it has to move in the clockwise direction to do so. But since g_1 starts moving clockwise from the next round after meeting b_1, any other follower group will always meet g_1 no later than when it meets any agent in b_1. Therefore any other follower group will merge with the active g_1 and the Byzantine adversary cannot do anything about this. Only thing agents in b_1 can do is form leader groups and merge with a group of parity different from g_1 as that group will never meet g_1 due to parity difference. Therefore a Byzantine agent can be part of two groups of Byzantine agents that merge with at most 2 active groups of different parity in any phase.

▶ **Lemma 3.** *In the first $\lceil \log_2(f+1) \rceil$ stages of the algorithm, the number of good agents that are not active at the beginning of the j^{th} stage is at most $34fj$, $0 \leq j \leq \lceil \log_2(f+1) \rceil$ whp. Also at the beginning of the j^{th} stage there is at least one true active group whp.*

Proof. Our proof is by the Principle of Mathematical Induction on j. When $j = 0$, the threshold for this stage is $2^j = 1$. All good groups have size ≥ 1, therefore the number of good agents that are not active at the beginning of the 0^{th} stage is 0. Also since there is at least one good agent, there is at least one true active group, thereby establishing the base case.

Induction Hypothesis: The number of good agents that are not active at the beginning of the t^{th} stage is at most $34ft$ whp. Also at the beginning of the t^{th} stage there is at least one true active group whp.

Inductive Step: Let us now focus on the start of the $j = (t+1)$th stage (or equivalently, the end of the t^{th} stage). We will now analyze what happens during the t^{th} stage. At the beginning of the t^{th} stage we know that the number of good agents that are not active is at most $34ft$ from induction hypothesis. Consider the groups that are active at the beginning of the t^{th} stage, we need to see how many of these good agents become inactive at the start of the $(t+1)^{th}$ stage, say x. Then we can bound the total number of good agents not active at the beginning of $(t+1)^{th}$ stage as $34ft + x$. This is because for the sake of upper bound we are assuming that agents that were inactive in the beginning of the t^{th} stage are inactive at the beginning of the $t+1^{th}$ stage and we only need to count the number of agents that were active at the beginning of the t^{th} stage but are inactive at the beginning of the $t+1^{th}$ stage (call it x).

By the Induction Hypothesis, there is at least one true active group at the beginning of t^{th} stage. Consider any such group g. Then for any other active group g_1, if g is a leader and g_1 is a follower and the two groups are of same parity then these two groups will merge (ignoring Byzantine influence for now). Each of these are independent events with probability $\frac{1}{2}$. Therefore ignoring the Byzantine influence, there is a constant probability of at least $\frac{1}{8}$ with which a group merges in every phase.

The adversary needs to form a group with at least 2^t agents to cross the threshold. We know that such a group can merge with at most two good active groups. Therefore in one phase the Byzantine adversary can cause at most $\frac{f}{2^{t-1}}$ false merges with good active groups. We call it a false merge as in such a merge the number of active groups doesn't decrease. The other thing the adversary can do is combine with a few good agents to form a group with at least 2^t agents. But to do this, the good agents have to be active and therefore in such a merge the number of active groups decreases. Therefore it is not beneficial for the adversary to do this.

Let $G_{j,i}$ denote the number of active groups in the i^{th} phase of the j^{th} stage. For any group, let the probability of choosing a good leader group (also active) to merge with be P_i. We have seen that $P_i \geq \frac{1}{8}$. Let C_i be the number of merges between good groups and let B_i be the number of merges with Byzantine groups.

We have

$$G_{t,i+1} = G_{t,i} - C_i \tag{1}$$

$$\mathbb{E}[G_{t,i+1}|G_{t,i}] = G_{t,i} - \mathbb{E}[C_i|G_{t,i}] \tag{2}$$

$$\mathbb{E}[G_{t,i+1}|G_{t,i}] = G_{t,i} - \left(\frac{G_{t,i}-1}{8} - B_i\right) \tag{3}$$

$$\mathbb{E}[G_{t,i+1}|G_{t,i}] \leq G_{t,i} - \left(\frac{G_{t,i}-1}{8} - \frac{f}{2^{t-1}}\right) \tag{4}$$

$$\mathbb{E}[G_{t,i+1}] \leq \frac{7}{8}\,\mathbb{E}[G_{t,i}] + \left(\frac{1}{8} + \frac{f}{2^{t-1}}\right) \tag{5}$$

Solving this recurrence relation we get

$$\mathbb{E}[G_{t,i}] \leq \left(G_{t,0} - 1 - \frac{8f}{2^{t-1}}\right)\left(\frac{7}{8}\right)^i + \left(\frac{8f}{2^{t-1}} + 1\right)$$

Since there are $2\log_{8/7} n$ phases in each stage, setting i to $\log_{8/7} n^2 = i^*$

$$\mathbb{E}[G_{t,i^*}] \leq \frac{G_{t,0} - 1 - \frac{8f}{2^{t-1}}}{n^2} + \left(\frac{8f}{2^{t-1}} + 1\right)$$

using Markov's inequality

$$P\left(G_{t,i^*} - \left(\frac{8f}{2^{t-1}} + 1\right) \geq 1\right) \leq \mathbb{E}\left[G_{t,i^*} - \left(\frac{8f}{2^{t-1}} + 1\right)\right]$$

$$P\left(G_{t,i^*} - \left(\frac{8f}{2^{t-1}} + 1\right) \geq 1\right) \leq \mathbb{E}[G_{t,i^*}] - \left(\frac{8f}{2^{t-1}} + 1\right)$$

$$P\left(G_{t,i^*} - \left(\frac{8f}{2^{t-1}} + 1\right) \geq 1\right) \leq \frac{G_{t,0} - 1 - \frac{8f}{2^{t-1}}}{n^2} \leq \frac{G_{t,0}}{n^2} \leq \frac{1}{n}$$

Therefore at the end of t^{th} stage there are $\frac{8f}{2^{t-1}} + 1$ active groups whp. Suppose all these active groups become inactive at the beginning of $t+1^{th}$ stage, then $x \leq \left(\frac{8f}{2^{t-1}} + 1\right) \cdot (2^{t+1} - 1)$ as if they are inactive at the beginning of $t+1^{th}$ stage then the number of agents in that group is less than 2^{t+1}.

$$x \leq \left(\frac{8f}{2^{t-1}} + 1\right) \times (2^{t+1} - 1) = 32f + 2^{t+1} - 1 - \frac{8f}{2^{t-1}}$$

since we are in the first part of the algorithm $t+1 \leq \lceil \log_2(f+1) \rceil$

$$x \leq 32f + 2^{t+1} - 1 - \frac{8f}{2^{t-1}} \leq 32f + 2f + 2 - 1 - 16$$
$$x \leq 34f$$

Therefore the number of good agents that are not active at the beginning of the $t+1^{th}$ stage is at most $34ft + 34f = 34(t+1)f$.
Let us count the number of good active agents at the beginning of t^{th} stage. The algorithm starts with $k-f$ good agents. At the beginning of the t^{th} stage there are atmost $34ft$ good agents that are inactive. Therefore there are at least $k-f-34ft$ good agents that are active at the beginning of the t^{th} stage. At the end of the t^{th} stage these good active agents are distributed among $8f/2^{t-1}+1$ groups. Therefore by Pigeon Hole Principle, at least one group has $\frac{k-f-34ft}{\frac{8f}{2^{t-1}}+1} = 2^{t-1} \times \frac{k-f-34ft}{8f+2^{t-1}}$ good active agents. Since $t \leq \lceil \log_2(f+1) \rceil - 1$ and $k-f > 34f(1+\log_2(f+1))+2$, we get that $\frac{k-f-34ft}{8f+2^{t-1}} \geq 4$. Therefore at the end of the t^{th} stage there is at least one group having 2^{t+1} good agents. Therefore at the beginning of the $(t+1)^{th}$ stage there is at least one true active group, thereby completing the proof. ◀

Note here that since there are only $\mathcal{O}(\log k)$ stages, the whp assumption holds in induction.

Consider the $j^* = \lceil \log_2(f+1) \rceil^{th}$ stage. The threshold in this stage is $2^{\lceil \log_2(f+1) \rceil} \geq f+1$. From Lemma 3 there is at least one true active group at the beginning of this stage. Consider any such group g. Then for any other group g_1, if g is a leader and g_1 is a follower and the two groups are of same parity then these two groups will merge. Each of these are independent events with probability $\frac{1}{2}$. Therefore there is a constant probability of at least $\frac{1}{8}$ with which a group merges in every phase.

In this stage, the threshold is greater than f, therefore the Byzantine agents cannot cause any false merges.

Let G_i denote the number of groups with at least one good agent in the i^{th} phase of this stage. Note that G_i denotes the number of good groups, not necessarily active. For any group, let the probability of choosing a good leader group to merge with be $P_i \geq \frac{1}{8}$. Let C_i denote the number of merges between good groups. Then $G_{i+1} = G_i - C_i$ and we have

$$\mathbb{E}[G_{i+1}|G_i] = G_i - \mathbb{E}[C_i|G_i] = G_i - \left(\frac{G_i - 1}{8}\right)$$
$$\mathbb{E}[G_{i+1}] \leq \frac{7}{8}\,\mathbb{E}[G_i] + \frac{1}{8}$$
$$\mathbb{E}[G_i] \leq (G_0 - 1)\left(\frac{7}{8}\right)^i + 1 \qquad \text{(Solving the above recurrence relation)}$$
$$\mathbb{E}[G_{i^*}] \leq \frac{G_0 - 1}{n^2} + 1 \qquad \text{(Setting } i \text{ to } \log_{8/7} n^2 = i^* \text{ which is the number of phases)}$$

Using Markov's inequality, we get $P(G_{i^*} - 1 \geq 1) \leq \frac{G_0 - 1}{n^2}$. Therefore at the end of Algorithm 1, there is one group whp. Thus,

▶ **Theorem 4.** *Given k agents placed arbitrarily on a graph G of n nodes, there exists a randomized gathering protocol that is resilient to a strongly-adaptive Byzantine adversary that can whp gather all good agents in $\mathcal{O}(n \log n \log k)$ rounds as long as k is greater than $f(35 + 34\log_2(f+1)) + 2$ where f is the number of Byzantine agents.*

Extension: No common orientation

Since there are only two possible orientations, by Pigeon Hole Principle, at least $\frac{k-f}{2}$ agents will have the same orientation. Therefore running Algorithm 1 and ensuring that only agents with same orientation interact, we can ensure gathering of at least $\frac{k-f}{2}$ agents. Due to the common port numbering of the graph, the agents can communicate their orientation and ensure they interact only with agents having same orientation as themselves. The restriction on the number of Byzantine agents will change slightly as essentially we now have half the number of good agents than we used to. The new constraint will be $\frac{k-f}{2} > 34f(1+\log_2(f+1))+2$. After gathering $\frac{k-f}{2} > f$ agents by running Algorithm 1, we can achieve gathering in another $\mathcal{O}(n \log n)$ phases by running the following algorithm:

- All groups choose one of the two orientations with probability $\frac{1}{2}$. The groups also choose whether to be leader or follower with equal probability.
- The follower groups go clockwise and make note of the largest leader group that they meet (ties are broken arbitrarily)
- The follower groups merge with the largest leader group that it met. This is possible as the location of leader groups is predictable

We know there is a group with $\frac{k-f}{2} > f$ number of good agents. So the largest group will be of size greater than f. Therefore in each phase there is a constant probability with which a group merges with the group with size at least $\frac{k-f}{2}$ and therefore gathering is achieved in $\mathcal{O}(n \log n)$ rounds whp.

2.3 With visual tracking

One significant drawback in the standard setting is that it limits $f \in \mathcal{O}(k/\log k)$, so a natural question we ask is whether we can relax the model in some reasonable manner to avoid this restriction on f. In this regard, we consider endowing agents with the ability to visually track other agents that are moving along with them. Consider two agents a and b that have been co-located from round $r-\delta$ to r (for some $\delta \geq 0$) but not in round $r-\delta-1$. With visual tracking, we assume that a and b know at round r that they have been co-located for δ rounds, but apart from that, don't have any memory of prior encounters. This leverages the common ability of mobile agents to be able to visually see the other objects that are moving along with them.

The algorithm runs in $\lceil 4 \log_{\frac{4(1+\alpha)}{3+4\alpha}} n \rceil$ phases. In each phase, each group decides to be either a leader group or a follower group. The leader group goes clockwise for some specified number of rounds. The follower group goes counterclockwise (after randomizing the parity of its position) and decides to merge with any one leader group that it meets. Again here a follower group meets a leader group only if they are of same parity. The probability with which a follower group chooses a leader group is in proportion to the number of agents in the leader group, which is achieved via reservoir sampling. Algorithm 2 presents the pseudocode.

Similar to before, there is no value for the Byzantine agents to behave as a follower group, but they can form leader groups and present themselves to follower groups. Note that if Byzantine agents tag along with a follower group and repeatedly behaves like a leader group in consecutive rounds, then the follower group can detect this due to visual tracking. Also note that if the Byzantine agent stops tagging along, then it can never catch up with that follower group again in that phase. Therefore a Byzantine agent can act as a leader agent to a good agent only twice in a phase (once for each orientation).

Algorithm 2 Gathering with visual tracking enabled.

function SELECTLEADER(counter)
  Count the number d of leader agents in the node based on the number of "I am a leader" messages received
  if d > 0 **then**
    temp $\leftarrow$ counter ; counter $\leftarrow$ counter + d
    Choose this leader group with probability $\left(1 - \frac{\text{temp}}{\text{counter}}\right)$ ▷ reservoir sampling
  end if
end function
function FOLLOWER
  counter $\leftarrow 0$
  go one step counterclockwise with probability $\frac{1}{2}$ ▷ group decision to randomize parity of position
  Call SELECTLEADER(counter)
  go one step counterclockwise and call SELECTLEADER(counter) for $\lfloor \frac{n}{2} \rfloor$ -1 rounds
  Move to the chosen leader group (its position is predictable)
end function
for $\lceil 4 \log_{\frac{4(1+\alpha)}{3+4\alpha}} n \rceil$ phases **do**
  Each good agent forms a group with other agents co-located with it.
  Decide to be a leader group or follower group with equal probability ▷ this decision is made as a group
  Leaders go clockwise and broadcast "I am a leader" message for $\lfloor \frac{n}{2} \rfloor$ rounds, followers call FOLLOWER()
end for

Time complexity analysis

Suppose in phase i, there are G_i number of good groups and H_i number of leader groups. We let $f = \alpha k$ for some α. The number of good agents is $k - f$ and out of them let L_i be leader agents. Let C_i be the number of merges between good groups. For any follower group, let the probability of choosing a good leader group to merge with be P_i. Since a Byzantine agent can appear as a leader group only once to a good follower group, we can lower bound P_i as, $P_i \geq \frac{L_i}{L_i+f}$. First lets prove a useful lemma

▶ **Lemma 5.** $\mathbb{E}\left[\frac{L_i}{L_i+f} \cdot (G_i - H_i) | G_i\right] \geq \frac{(1-\alpha)(G_i-1)}{4}$ *(with expectation taken over different combinations of leader groups)*

Proof. At the start of phase i let the sizes of the G_i groups be $s_1, s_2, s_3, ..., s_{G_i}$ excluding the Byzantine agents. Out of these let the sizes of the groups which choose to be leader groups be $s_{t_1}, s_{t_2}, s_{t_3}, ...s_{t_{H_i}}$.

Define the set I_m as $\{(i_1, i_2, ...i_m) \mid i_1 < i_2 < ... < i_m \text{ and } 1 \leq i_l \leq G_i \ \forall\ 1 \leq l \leq m\}$ for all $1 \leq j \leq G_i$. The set I_m denotes the set of possible combinations of choosing m leader groups from the G_i groups. Let $I = I_1 \cup I_2 \cup \cdots \cup I_{G_i}$. The probability of exactly those groups becoming leader groups, for any element in I is $\frac{1}{2^{G_i}}$.

$$\mathbb{E}\left[\frac{L_i}{L_i+f}\times(G_i-H_i)|G_i\right]=\sum_{e\in I}\frac{1}{2^{G_i}}\times(G_i-|e|)\times\sum_{i_j\in e}\left(\frac{\sum s_{i_j}}{\sum s_{i_j}+f}\right)$$

$$\mathbb{E}\left[\frac{L_i}{L_i+f}\times(G_i-H_i)|G_i\right]=\sum_{m=1}^{G_i}\sum_{e\in I_m}\frac{1}{2^{G_i}}\times(G_i-m)\times\sum_{i_j\in e}\left(\frac{\sum s_{i_j}}{\sum s_{i_j}+f}\right)$$

Now

$$\begin{aligned}\sum_{e\in I_m}\frac{1}{2^{G_i}}\times(G_i-m)\times\sum_{i_j\in e}\left(\frac{\sum s_{i_j}}{\sum s_{i_j}+f}\right)&\geq\sum_{e\in I_m}\frac{1}{2^{G_i}}\times(G_i-m)\times\sum_{i_j\in e}\left(\frac{\sum s_{i_j}}{k}\right)\\&=\frac{1}{2^{G_i}}\times(G_i-m)\times\frac{k-f}{k}\times\binom{G_i-1}{m-1}\end{aligned}$$

Therefore

$$\begin{aligned}\mathbb{E}\left[\frac{L_i}{L_i+f}\times(G_i-H_i)|G_i\right]&\geq\sum_{m=1}^{G_i}\frac{1}{2^{G_i}}\times(G_i-m)\times\frac{k-f}{k}\times\binom{G_i-1}{m-1}\\&\geq\frac{1}{2^{G_i}}\times\frac{k-f}{k}\times\sum_{m=1}^{G_i}(G_i-m)\times\binom{G_i-1}{m-1}\end{aligned}$$

Using a standard result from binomial mathematics, we know $\sum_{m=1}^{G_i}(G_i-m)\times\binom{G_i-1}{m-1}=2^{G_i-2}(G_i-1)$. Therefore

$$\mathbb{E}\left[\frac{L_i}{L_i+f}\cdot(G_i-H_i)\mid G_i\right]\geq\frac{(k-f)(G_i-1)}{4k}$$

Also since $f=\alpha k$

$$\mathbb{E}\left[\frac{L_i}{L_i+f}\cdot(G_i-H_i)|G_i\right]\geq\frac{(1-\alpha)(G_i-1)}{4}$$ ◀

Now lets analyze how the number of good groups varies

$$\begin{aligned}\mathbb{E}[G_{i+1}|G_i]&=G_i-\mathbb{E}[C_i|G_i]=G_i-\mathbb{E}[P_i\times(G_i-H_i)|G_i]\\\mathbb{E}[G_{i+1}|G_i]&\leq G_i-\mathbb{E}\left[\frac{L_i}{L_i+f}\times(G_i-H_i)|G_i\right]\\\mathbb{E}[G_{i+1}|G_i]&\leq G_i-\frac{(1-\alpha)(G_i-1)}{4}\leq G_i\times\left(\frac{3+\alpha}{4}\right)+\frac{1-\alpha}{4} && \text{(Using Lemma 5)}\\\mathbb{E}[G_i]&\leq G_0\left(\frac{3+\alpha}{4}\right)^i+1 && \text{(Solving the recurrence relation)}\\\mathbb{E}[G_{i^*}]&\leq\frac{1}{n^2}+1 && (\text{Setting } i \text{ to } \log_{\frac{4}{3+\alpha}}G_0n^2=i^*)\end{aligned}$$

Using Markov's inequality, we get $P(G_{i^*}-1\geq 1)\leq\frac{1}{n^2}$. Therefore after $\log_{\frac{4}{3+\alpha}}G_0n^2$ phases, the number of groups is 1 whp i.e. the algorithm terminates in $\log_{\frac{4}{3+\alpha}}G_0n^2$ phases whp. Thus,

▶ **Theorem 6.** *Given k agents capable of visual tracking placed arbitrarily on a graph G of n nodes, there exists a randomized gathering protocol that is resilient to a strongly-adaptive Byzantine adversary that can whp gather all good agents in $\mathcal{O}(\frac{n\log n}{\log(4/(3+\alpha))})$ rounds as long as k is greater than f where f is the number of Byzantine agents.*

Extension: No common orientation

Let $g = k - f$ denote the number of good agents. For simplicity we assume g is $\Omega(\log n)$. At the beginning, each agent chooses one of the two possible orientations with equal probability. Let C denote the number of good agents having one of the orientations. Then from Chernoff bound we get

$$P\left(|C - \mathbb{E}[C]| \geq \frac{\mathbb{E}[C]}{\delta}\right) \leq 2\exp\left(-\frac{\mathbb{E}[C]\delta^2}{3}\right)$$
$$P\left(|C - \frac{g}{2}| \geq \frac{g}{4}\right) \leq 2\exp\left(-\frac{g}{24}\right) \qquad (\mathbb{E}[C] = \tfrac{g}{2} \text{ and setting } \delta = \tfrac{1}{2})$$

Since g is $\Omega(\log n)$, with high probability there are at least $\frac{g}{4}$ good agents with each orientation. After randomizing the orientation in the above mentioned way, the agents execute Algorithm 2 with agents interacting only with other agents of same orientation. Lets compute the time complexity for gathering the agents with the orientation having lesser number of good agents. We need to compute the α' value for this orientation

$$\begin{aligned}\alpha' &\leq \frac{f}{g/4 + f}\\ &= \frac{f}{(k-f)/4 + f}\\ &= \frac{\alpha}{(1-\alpha)/4 + \alpha}\\ &= \frac{4\alpha}{(1+3\alpha)}\end{aligned}$$

Therefore the agents of this orientation gather in $\mathcal{O}\left(\frac{n\log n}{\log(4/(3+\alpha'))}\right) = \mathcal{O}\left(\frac{n\log n}{\log(4+12\alpha/(3+13\alpha))}\right)$ rounds. In these many rounds, good agents of each orientation would have gathered. Therefore we now have two groups of good agents. These groups are now gathered by running Algorithm 2 except each group chooses one of the two possible orientations with equal probability in each phase. Then in each phase, the probability that the two groups have the same orientation and same parity is $1/4$ and the probability that one of them is a leader and the other is a follower is $1/2$. Given that these events happen, the probability that the two groups combine is at least $\frac{g/4}{g/4+f} = \frac{1-\alpha}{1+3\alpha}$. Therefore in each phase the two groups combine with probability at least $\frac{1-\alpha}{8(1+3\alpha)}$. Hence the two groups combine in $\mathcal{O}(\frac{n\log n}{\log(8+24\alpha/7+25\alpha)})$ rounds with high probability. Therefore the overall algorithm terminates in $\mathcal{O}(\frac{n\log n}{\log(8+24\alpha/7+25\alpha)})$ rounds whp.

3 Conclusion

We studied how to exploit randomization to achieve gathering quickly in the presence of strong Byzantine agents and when agents are anonymous. Our main focus was on rings similar to intial research in other fundamental global symmetry breaking problems [19]. Our algorithms and analysis show that these problems are non-trivial and showcase the need to develop this theory further. Our work raises many follow-up questions. For example, how fast can we gather when the number of good agents is asymptotically smaller than the number of Byzantine agents? Consider the case when $k = n$ and $f = n - \mathcal{O}(1)$, can we do better than achieving gathering through coalescing random walks which takes $\tilde{\mathcal{O}}(n^2)$ time? Or is $\tilde{\Omega}(n^2)$ a lower bound for such algorithms?

Earlier works like [13] have focused on deterministic algorithms with labelled agents. While the use of randomization is clear in the anonymous setting, a natural question is whether randomization can improve the time efficiency in the case of labelled agents.

References

1 Noa Agmon and David Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM J. Comput.*, 36(1):56–82, July 2006. doi:10.1137/050645221.

2 Steve Alpern, V. J. Baston, and Skander Essegaier. Rendezvous search on a graph. *Journal of Applied Probability*, 36(1):223–231, 1999. URL: http://www.jstor.org/stable/3215416.

3 Steve Alpern and Shmuel Gal. *The theory of search games and rendezvous*, volume 55. Springer Science & Business Media, 2006.

4 James Aspnes. Lower bounds for distributed coin-flipping and randomized consensus. *Journal of the ACM (JACM)*, 45(3):415–450, 1998.

5 Michael Ben-Or and Nathan Linial. Collective coin flipping, robust voting schemes and minima of banzhaf values. In *26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 408–416. IEEE, 1985.

6 Sébastien Bouchard, Yoann Dieudonné, and Bertrand Ducourthial. Byzantine gathering in networks. *Distrib. Comput.*, 29(6):435–457, November 2016. doi:10.1007/s00446-016-0276-9.

7 Sébastien Bouchard, Yoann Dieudonné, and Anissa Lamani. Byzantine Gathering in Polynomial Time. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 147:1–147:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2018.147.

8 Colin Cooper, Robert Elsässer, Hirotaka Ono, and Tomasz Radzik. Coalescing random walks and voting on graphs. In *Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing*, PODC '12, pages 47–56, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2332432.2332440.

9 Gianlorenzo D'angelo, Gabriele Di Stefano, and Alfredo Navarra. Gathering on rings under the look—compute—move model. *Distrib. Comput.*, 27(4):255–285, August 2014. doi:10.1007/s00446-014-0212-9.

10 Gianlorenzo D'angelo, Alfredo Navarra, and Nicolas Nisse. A unified approach for gathering and exclusive searching on rings under weak assumptions. *Distrib. Comput.*, 30(1):17–48, February 2017. doi:10.1007/s00446-016-0274-y.

11 Shantanu Das et al. Mobile agents in distributed computing: Network exploration. *Bulletin of EATCS*, 1(109), 2013.

12 Xavier Défago, Maria Potop-Butucaru, and Philippe Raipin Parvédy. Self-stabilizing gathering of mobile robots under crash or byzantine faults. *Distributed Comput.*, 33(5):393–421, 2020. doi:10.1007/s00446-019-00359-x.

13 Yoann Dieudonné, Andrzej Pelc, and David Peleg. Gathering despite mischief. *ACM Trans. Algorithms*, 11(1), August 2014. doi:10.1145/2629656.

14 R. Eguchi, N. Kitamura, and T. Izumi. Fast neighborhood rendezvous. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 168–178, Los Alamitos, CA, USA, December 2020. IEEE Computer Society. doi:10.1109/ICDCS47774.2020.00030.

15 Uriel Feige. Noncryptographic selection protocols. In *40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 142–152. IEEE, 1999.

16 Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*. Springer, 2019. doi:10.1007/978-3-030-11072-7.

17 Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Giovanni Viglietta. Distributed computing by mobile robots: uniform circle formation. *Distributed Comput.*, 30(6):413–457, 2017. doi:10.1007/s00446-016-0291-x.

18 Jion Hirose, Junya Nakamura, Fukuhito Ooshita, and Michiko Inoue. Gathering with a strong team in weakly byzantine environments. In *International Conference on Distributed Computing and Networking 2021*, ICDCN '21, pages 26–35, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3427796.3427799.

19 Daniel S. Hirschberg and James B Sinclair. Decentralized extrema-finding in circular configurations of processors. *Communications of the ACM*, 23(11):627–628, 1980.

20 Zool Hilmi Ismail and Nohaidda Sariff. A survey and analysis of cooperative multi-agent robot systems: Challenges and directions. In Efren Gorrostieta Hurtado, editor, *Applications of Mobile Robots*, chapter 1. IntechOpen, Rijeka, 2019. doi:10.5772/intechopen.79337.

21 Taisuke Izumi, Tomoko Izumi, Sayaka Kamei, and Fukuhito Ooshita. Feasibility of polynomial-time randomized gathering for oblivious mobile robots. *IEEE Transactions on Parallel and Distributed Systems*, 24(4):716–723, 2013. doi:10.1109/TPDS.2012.212.

22 Tomoko Izumi, Taisuke Izumi, Sayaka Kamei, and Fukuhito Ooshita. Mobile robots gathering algorithm with local weak multiplicity in rings. In *Proceedings of the 17th International Conference on Structural Information and Communication Complexity*, SIROCCO'10, pages 101–113, Berlin, Heidelberg, 2010. Springer-Verlag. doi:10.1007/978-3-642-13284-1_9.

23 Gangshan Jing, Yuanshi Zheng, and Long Wang. Consensus of multiagent systems with distance-dependent communication networks. *IEEE Transactions on Neural Networks and Learning Systems*, PP, August 2016. doi:10.1109/TNNLS.2016.2598355.

24 Ralf Klasing, Adrian Kosowski, and Alfredo Navarra. Taking advantage of symmetries: Gathering of many asynchronous oblivious robots on a ring. *Theoretical Computer Science*, 411(34):3235–3246, 2010. doi:10.1016/j.tcs.2010.05.020.

25 Evangelos Kranakis and Danny Krizanc. *Mobile Agents and Exploration*, pages 1338–1341. Springer New York, New York, NY, 2016. doi:10.1007/978-1-4939-2864-4_242.

26 Silvio Micali and Tal Rabin. Collective coin tossing without assumptions nor broadcasting. In *Conference on the Theory and Application of Cryptography*, pages 253–266. Springer, 1990.

27 Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, USA, 2nd edition, 2017.

28 Fukuhito Ooshita, Shinji Kawai, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Randomized gathering of mobile agents in anonymous unidirectional ring networks. *IEEE Trans. Parallel Distributed Syst.*, 25(5):1289–1296, 2014. doi:10.1109/TPDS.2013.259.

29 Michael Saks. A robust noncryptographic protocol for collective coin flipping. *SIAM Journal on Discrete Mathematics*, 2(2):240–244, 1989.

30 Masahiro Shibata, Toshiya Mega, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Uniform deployment of mobile agents in asynchronous rings. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, pages 415–424, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2933057.2933093.

31 Yuichi Sudo, Daisuke Baba, Junya Nakamura, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. An agent exploration in unknown undirected graphs with whiteboards. In *Proceedings of the Third International Workshop on Reliability, Availability, and Security*, WRAS '10, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1953563.1953570.

32 Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, 1999. doi:10.1137/S009753979628292X.

33 Masashi Tsuchida, Fukuhito Ooshita, and Michiko Inoue. Byzantine gathering in networks with authenticated whiteboards. In Sheung-Hung Poon, Md. Saidur Rahman, and Hsu-Chun Yen, editors, *WALCOM: Algorithms and Computation*, pages 106–118, Cham, 2017. Springer International Publishing.

34 Xiaoping Yun, Gokhan Alptekin, and Okay Albayrak. Line and circle formation of distributed physical mobile robots. *J. Field Robotics*, 14(2):63–76, 1997.