Communication and Location Discovery in Geometric Ring Networks*

Leszek Gąsieniec[a], Tomasz Jurdzinski[b], Russell Martin[a,*], Grzegorz Stachowiak[b]

[a] *Dept of Computer Science, University of Liverpool, United Kingdom*
[b] *Institute of Computer Science, University of Wrocław, Poland*

Abstract

We study a distributed coordination mechanism for uniform agents located on a circle. The agents perform their actions in synchronised rounds. At the beginning of each round an agent chooses the direction of its movement from clockwise, anticlockwise, or idle, and moves at unit speed during this round. Agents are not allowed to overpass, i.e., when an agent collides with another it instantly starts moving with the same speed in the opposite direction (without exchanging any information with the other agent). However, at the end of each round each agent has access to limited information regarding its trajectory of movement during this round. We assume that n mobile agents are initially located on a circle unit circumference at arbitrary but distinct positions unknown to other agents. The agents are equipped with unique identifiers from a fixed range. The *location discovery* task to be performed by each agent is to determine the initial position of every other agent.

Our main result states that, if the only available information about movement in a round is limited to distance between the initial and the final position, then there is a superlinear lower bound on time needed to solve the location discovery problem. Interestingly, this result corresponds to a combinatorial symmetry breaking problem, which might be of independent interest. If, on the other hand, an agent has access to the distance to its first collision with another agent in a round, we design an asymptotically efficient and close to optimal solution for the location discovery problem. Assuming that agents are anonymous (there are no IDs distinguishing them), our solution applied to randomly chosen IDs from appropriately chosen range gives an (almost) optimal algorithm, improving upon the complexity of previous randomized results.

Keywords: mobile robots, distributed algorithms, location discovery, boundary patrolling, combinatorial structures

---

*A preliminary version of this paper appeared in the 35th IEEE International Conference on Distributed Computing Systems (ICDCS) 2015, pp. 517–526.

*Corresponding author.
Email address: russell.martin@liverpool.ac.uk (Russell Martin)

1. Introduction

One of the most studied network topologies in the context of distributed computation, as well as coordination mechanisms for mobile agents, is the ring network [1, 2, 3]. Recently, studies of geometric ring networks were initiated in the context of terrain exploration by agents/robots with limited communication and navigation capabilities [4, 5]. This refers to the concept of swarms, i.e., large groups of limited but cost-effective entities (robots, agents) that can be deployed to perform an exploration in a hard-to-access hostile environment. The usual swarm robot properties include anonymity, negligible dimensions, no explicit communication, and no common coordinate system (cf. [6]). Some of these models assume limited visibility of the surrounding environment and asynchronous operation. In most situations involving such weak robots, the fundamental research question concerns the feasibility of solving a given task (cf. [7, 8]). The cost of the algorithm is usually measured in terms of length of a robot's walk or the time needed to complete the task. There are several algorithmic solutions providing efficient distributed coordination mechanisms in a variety of models, e.g. [9, 10, 6]. The dynamics of "beads on a ring" and billiard systems is also of independent interest, e.g. [11].

One of the fundamental tasks in ad hoc distributed environments is to determine the actual network topology. This topic was studied in networks modeled as graphs [12, 13, 14], as well as networks deployed in a geometric environment [15, 16, 17, 18]. Most of those solutions work under the assumption that neighbors (in a graph) can exchange messages, or that agents have some visibility allowing them to inspect their nearby neighborhood.

In the case of networks containing swarm robots, communication and visibility capabilities are often severely restricted. Lack of these capabilities in some settings can be overcome by the possibility of agents monitoring their own trajectories, sensing collisions with other agents, or inferring some information from the fact that all agents behave in a fixed regular fashion. Another factor simplifying various tasks might be a restriction on the class of environments or the allowed movement trajectories of agents.

Following [5, 19] we consider a model where the agents operate in synchronised rounds, and they lack direct means of communication. The trajectory of an agent in a given round is represented as a continuous curve that connects the start and the end points of the route adopted by the agent. While moving along their trajectories the agents collide with their immediate neighbours, and information on the exact location of those collisions might be recorded and further processed. When agents are located on a circle, each agent may eventually conclude on the relative location of all agents' initial positions, even given only limited information about its trajectory, e.g., at specified time intervals. This, in turn, enables other distributed mechanisms based on full synchronisation, e.g. equidistant distribution along the circumference of the circle and an optimal boundary patrolling scheme. Most of the models adopted in the literature on swarms assume that the agents are either almost or entirely oblivious, i.e., throughout the computation process the agents follow a very simple, rarely amendable, routine of actions. Such a scenario is studied in [19, 20, 21, 22], where agents are entirely oblivious but can register all their collisions. (In [20, 21] agents might have different velocities, and in [21] they might have different masses.) In this paper we adopt the model from [5], where even the possibility of an agent tracking its own trajectory is severely limited. (The model we study can also be seen as a variation of that studied in [23].) In order to overcome this weakness, more adaptivity of behavior is allowed. So, the ultimate goal of this line of research is to determine how much information about their trajectories agents need to solve some communication or exploration problems, and how efficiently these problems can be solved.

Our main focus is on deterministic solutions for these communication and exploration problems for agents having unique IDs, which is necessary for symmetry breaking. However, our results can be applied to randomly chosen IDs from an appropriately chosen range to improve upon the complexity of previous randomized results. See Section 8 for more details in the randomized setting.

1.1. Model

A network A is deployed on a circle with circumference one, along which n agents (i.e., the elements of A) move and interact in synchronised rounds, where each round lasts one unit of time. The agents do not necessarily share the same sense of direction, i.e., while each agent distinguishes between its own clockwise (C) and anticlockwise (A) directions, agents may not have a coherent view on this. The direction "clockwise" is also called "right", and we also refer to "anticklockwise" as "left". At the beginning of a round, an agent a assigns one of the values from the set $\{\text{idle}, \text{right}, \text{left}\}$ to its local variable dir_a. When the option "idle" is chosen, the agent starts the round without moving in any direction. In the case that $\text{dir}_a = \text{right}$ or $\text{dir}_a = \text{left}$, the agent starts the round moving at unit speed on the circle in the direction dir_a. We assume that agents are not allowed to overpass each other along the circle. When two agents moving in the opposite directions collide with each other, they instantly start moving with the same speed but in the opposite directions. If an agent a moving in the direction $\text{dir} \in \{\text{right}, \text{left}\}$ collides with another agent a' which is currently idle, then a stays idle after the collision and a' immediately starts moving in the direction dir (i.e., in the same "objective" direction in which a was moving before the collision, irrespective of the fact whether a and a' have consistent senses of direction). The agents cannot leave marks on the ring, they have zero visibility, and they cannot exchange messages. Instead, during each round each agent has access to some (specified) information about its trajectory during this round. This information can be processed or stored for further analysis. Since the agents never overpass, we may assume that the agents are arranged in an implicit (i.e. never disclosed to the agents) periodic order from a_1 to a_n.

Each agent has access to its relative position at the end of a round; more precisely, it knows the distance dist() to the right (according to its own sense of direction) between its position at the beginning of the round and the position at the end of the round, measured in the agent's clockwise direction. In other words, there is no "universal" coordinate system on the circle, the distance is measured relative to the starting position of an agent at the start of the round. We distinguish three variants of the model:

- *basic* – an agent is **not** allowed to start a round idle, it has to start moving either in the right or the left direction;
- *lazy* – an agent is allowed to start a round idle, moving right or left;
- *perceptive* (or 1-perceptive) – this is the basic model with the additional feature that an agent gets the value coll() at the end of each round, which is equal to the distance between its position at the beginning of the round and the position of its first collision in that round.

Thus, the basic model is the weakest one. The lazy model extends the basic model by increasing an agent's freedom in choosing various movement options. The perceptive model, on the other hand, extends the basic model by providing more information about an agent's own trajectory to itself.

1.2. Notation and definitions

In this paper we address deterministic algorithms which require (for symmetry breaking) that agents have unique identifiers (IDs). We assume that each ID is a natural number in the set $\{1, \ldots, N\}$ and each agent is aware of the value of N.[1] As the agents know N, we can assume that the ID of each agent is a bit string of length $\log N$. (More precisely, we need $\lceil \log(N+1) \rceil$ bits.)

We also consider randomized algorithms, and in this case the agents are uniform and anonymous. That is, they are indistinguishable from other agents; in particular, no IDs are provided in this case.

The actual number of agents is denoted by n. In general, we assume that the only information available to agents about n is whether n is odd or even. Additionally, we assume that $N \geq n > 4$ or $n = 3$.[2]

For an agent a, ID_a denotes the identifier of a, and $\mathrm{ID}_a[i]$ denotes the ith bit of ID_a. We also assume that at the beginning of each round, each agent a can set a local variable dir_a with value left, right or idle (only in the lazy model), and the value dir_a (in general) determines the way in which a starts moving in

[1]The assumption that IDs belong to the known range is very common in anonymous networks, it is motivated e.g. by the fact that manufacturers assign unique MAC addresses to devices, etc. However, for breaking symmmetry, it is sufficient that nodes have unique IDs, while they do not know the range of IDs. Some of the techniques from this paper might be applicable in such a scenario, by running respective algorithms for consecutive values $N = 2^0, 2^1, 2^2, \ldots$ Nevertheless, in general, efficient solutions in this scenario might require new tools and ideas.

[2]At the end of the paper we address the issue how to determine that $n \leq 4$ and how efficiently the parity of n can be determined by the agents.

the next round. For natural numbers i and j, let $[i, j] = \{k \in \mathbb{N} \,|\, i \leq k \leq j\}$ and let $[i] = [1, i]$.

By *right ring distance* between agents a and a' we mean 1 plus the number of agents on the ring between a and a' going from a to a' in the clockwise direction. The *left ring distance* is defined analogously. If no common sense of direction is established, the right/left distance from the point of view of an agent is measured according to its own sense of direction. Observe that, by the model's restrictions, the relative order of agents on the ring does not change. Thus, the ring distance between agents does not change during executions of algorithms. For an agent a, $N_a(k)$ denotes the set of agents in ring distance at most k from a.

Let $\mathcal{S} = (S_1, \ldots, S_k)$ be a sequence of subsets of $[N]$. We say that agents *execute* $\mathcal{S}$ in a sequence of k rounds if the agent $a \in [N]$ sets $\text{dir}_a = \text{right}$ in the ith round iff $a \in S_i$; otherwise $\text{dir}_a = \text{left}$. Moreover, given a set $A' \subseteq A$ of "marked" agents we say that $\mathcal{S}$ is *executed* on A' if agents from A' set their directions in consecutive rounds according to $\mathcal{S}$, while each $a \in A \setminus A'$ sets dir_a to right in each round.

1.3. A basic tool

Let an (n_C, n_A)-round be any round in which n_C agents start the round clockwise and n_A agents start the round anticlockwise (according to some "objective" sense of direction). A simple but key property of the ring networks was observed in [5].

Lemma 1. *[5] Assume that the positions of agents $a_1, \ldots, a_n$ at the start of an (n_C, n_A)-round are $p_1, \ldots, p_n$. Then, during the round all agents are rotated along the initial positions by a rotation index of $r = (n_C - n_A) \bmod n$, i.e., the position of a_i at the end of the round is $p_{1+(i-1+r) \bmod n}$.*

By the above lemma, each agent experiences the same shift by r places in a round. Therefore, we define the *rotation index* of a round as the number of places by which agents move in that round in the clockwise direction. Thus, the rotation index of an (n_C, n_A)-round is equal to $(n_C - n_A) \bmod n$.

In this paper, SingleRound denotes one round of computation in which each agent a starts moving in the direction dir_a. ReversedRound denotes one round of computation in which each agent a starts moving opposite to the direction dir_a. Note that, after an execution of SingleRound followed by ReversedRound, each agent a gets to the position occupied by a before these two rounds transpired, provided agents do not change their local variables dir_a in between the two rounds.

1.4. Problems considered in the paper and previous results

The main goal of this paper is to evaluate the feasibility and complexity of the *location discovery* (LD) problem in the models we consider. The **location discovery problem** is to determine the initial position (i.e. starting position when all agents simultaneously "wake up" to begin the procedure) of every other agent[3]. That is, at the end of an execution of an algorithm, each agent $a \in A$ should know initial positions of all other agents, with respect to its own initial position.

We consider several problems which turn out to be efficient tools for solving the location discovery problem. Moreover, they are interesting as themselves, since they are useful in designing more complicated communication mechanisms. Below, we define these problems.

Direction agreement. The *direction agreement* is to agree on which direction is *clockwise* and which is *counterclockwise*. That is, at the end of the direction agreement procedure all agents have coherent view on which direction is clockwise, independent of any "objective" sense of direction. All agents must terminate this procedure at the same round.

Leader election. The *leader election* problem is solved when *exactly* one agent is assigned the status "leader" and all other agents have the status "non-leader". (Note that we do not require that non-leaders know the ID of the leader or any other information about it.) As before, the agents must terminate this procedure at the same round.

Nontrivial move problem. We say that a round is a *trivial move* if its rotation index belongs to the set $\{0, n/2\}$ and it is a *nontrivial move* otherwise. The *nontrivial move* problem is to assign to each agent a its direction dir_a such that if in a round each agent a starts in the direction dir_a, then this round is a nontrivial move. As for the other two problems, all agents must terminate this problem at the same round.

For the direction agreement, leader election, and the nontrivial move problem we use the notion of *coordination problems*.

As a tool for solutions of other problems, we also consider the emptiness testing problem.

Emptiness testing. Let $A \subseteq [N]$ denote the set of IDs of agents in the network. *Emptiness testing* is a protocol which given $B \subseteq [N]$, determines whether $B \cap A = \emptyset$. (That is, each agent $a \in A$ knows B as an input and it is aware of the fact whether $A \cap B \neq \emptyset$ at the end of an execution of the protocol.)

The location discovery problem in the basic and perceptive model were studied in [5]. It has been shown that there exists a randomized solution for anonymous networks (i.e. for identical agents without IDs) working in time $O(n \log^2 n)$ with high probability in the perceptive model. If n is odd, this solution works also under the assumptions of the basic model. In [19], oblivious algorithms are studied, in which an agent is not allowed to change its direction at the beginning of a round. However, agents have access to positions of all their collisions during a round. It has been shown that, for some initial configurations, the location

[3] In [5], it is required that eventually each agent stops at its initial position. In this paper this requirement is ignored. A simple way to achieve this is to reverse all rounds of the algorithm (see properties of SingleRound and ReversedRound). However, in our solutions agents collect information which allows them to get back on the initial positions much faster than by reversing all steps of an original algorithm.

discovery problem is infeasible in this model. On the other hand, there is a family of initial configurations for which the location discovery can be solved efficiently in (sub)linear time.

1.5. Our results

In this paper, we examine the complexity of deterministic leader election, nontrivial move, direction agreement, and location discovery problems. We also study the impact on the complexity of these problems of the parity of n, and whether agents initially share the same sense of direction. In all considered settings we obtain results which are optimal or close to optimal (see Tables 1 and 2).

First, we show that the complexity of all coordination problems is asymptotically equal up to an additive $O(\log N)$ factor. This gives an efficient and simple solution for location discovery when n is odd (Section 3).

The key technical contribution of the paper states that lack of the common sense of direction for even n substantially changes the complexity of all considered problems, at least in the basic and lazy model. That is, the complexity of all coordination problems and location discovery is superlinear with respect to n if N is superpolynomial with respect to n (i.e., $N = n^{\omega(1)}$). More precisely, all considered problems require $\Omega(n \log(N/n)/\log n)$ rounds in this setting (see Table 1). The reason for these large lower bounds is that the considered tasks require the solution of a kind of "symmetry-breaking" problem. We define a purely combinatorial notion of a *distinguisher* (see Section 4) to describe this symmetry-breaking problem which we think might be of independent interest). Using the probabilistic method, we also show that this bound is tight.

Table 1: Deterministic solutions in general setting ($n > 4$ or $n = 3$, and known parity). In the table, we use the following abbreviations: bm for the basic model, lm for the lazy model and pm for the perceptive model. The function $f(n, N)$ in the last row is equal to $f(n, N) = (\log N)(\log n) + \sqrt{n \log N}$ (see Lemma 41 and Theorem 47 for details).

	leader election	nontrivial move	direction agreement	location discovery
odd n	$O(\log N)$ (Cor. 19)	$\Theta(\log(N/n))$ (Prop. 20)	$O(1)$ (Prop. 18)	$n + O(\log N)$ (Lem. 6 & Cor.19)
bm, even n	$\Theta(\frac{n \log(N/n)}{\log n})$ (Cor. 30)	$\Theta(\frac{n \log(N/n)}{\log n})$ (Cor. 30)	$\Theta(\frac{n \log(N/n)}{\log n})$ (Cor. 30)	not solvable (Lem. 5)
lm, even n	$\Theta(\frac{n \log(N/n)}{\log n})$ (Cor. 33)	$\Theta(\frac{n \log(N/n)}{\log n})$ (Cor. 33)	$\Theta(\frac{n \log(N/n)}{\log n})$ (Cor. 33)	$n + \Theta(\frac{n \log(N/n)}{\log n})$ (Cor. 34)
pm	$O(f(n, N))$ (Thm. 7 & Lem. 41)	$O(f(n, N))$ (Thm. 7 & Lem. 41)	$O(f(n, N))$ (Thm. 7 & Lem. 41)	$\frac{n}{2} + O(f(n, N))$ (Lem. 6 & Thm. 47)

For the perceptive model, we provide a construction which solves the nontrivial move problem in $O((\log N)(\log n) + \sqrt{n \log N})$ rounds, thus the lower bound $\Omega(n \log(N/n)/\log n)$ does not hold for this case.

Table 2: Deterministic solutions *with* common sense of direction ($n > 4$ or $n = 3$, and known parity). In the table, we use the following abbreviations: bm for the basic model, lm for the lazy model and pm for the perceptive model.

	leader election	nontrivial move	location discovery
odd n	$O(\log N)$ (Lem. 13)	$\Theta(\log(N/n))$ (Prop. 20)	$n + O(\log N)$ (Lem. 17)
bm, even n	$O(\log^2 N)$ (Lem. 13)	$O(\log^2 N)$ (Lem. 15)	not solvable (Lem. 5)
lm, even n	$O(\log N)$ (Lem. 13)	$O(\log N)$ (Lem. 15)	$n + O(\log N))$ (Lem. 17)
pm	$O(\log N)$ (Lem. 13)	$O(\log N)$ (Lem. 15)	$\frac{n}{2} + O((\log N)(\log n) + \sqrt{n})$ (Lem. 6, 44 & 46)

We also show that using solutions of the coordination problems considered in the paper, the location discovery problem can be solved in $n + o(n)$ rounds in the lazy model (or basic model with odd n) and in $n/2 + o(n)$ rounds in the perceptive model, provided $\log N = o(\sqrt{n})$ (see the last columns of Tables 1 and 2 for details). These results are optimal up to additive $o(n)$ factors (using Lemma 6 described later).

A summary of results in Tables 1 and 2 is given for the scenario when the parity of the number of agents n is known initially. If that is not the case, one can determine the parity of n in $O(\log^2 N)$ rounds in the basic model and in $O(\log N)$ rounds in other models (Theorem 53). Then, the upper bounds in the basic model in the general setting (Table 1) for coordination problems become $O(\log^2 N + \frac{n\log(N/n)}{\log n})$. Thus, the lower bound $\Omega(\frac{n\log(N/n)}{\log n})$ remains tight provided $\log N = O(n/\log n)$.

Interestingly, our results can be applied to improve the best randomized solutions in anonymous networks (without IDs of agents) [5] by assigning IDs randomly from the range $[m^2]$, for an appropriately chosen approximation, m, of the size n.

The technical results on distinguishers presented in Section 4 give some new insight on the classic combinatorial search problem called *counterfeit coins problem*. We discuss connections between distuinguishers and other combinatorial search problems in Section 9.

1.6. Structure of the paper

First, in Section 2, we provide some basic facts and tools regarding the considered model which will be used throughout the paper. In Section 3, we establish relationships between asymptotic complexities of coordination problems, summarized in Theorem 7. We also discuss consequences of these reductions when the size n of a network is odd.

In Section 4, the complexity of the nontrivial move problem in the basic model is examined. In particular, a superlinear lower bound on the complexity of nontrivial move is shown, and an (almost) matching upper bound is provided. In Section 5 we show that the lower bound on symmetry breaking from Section 4 applies in the lazy model as well. In Section 6, a construction allowing us to reduce the complexity of location discovery to $n/2 + o(n)$ is described in the perceptive model.

As we assume that $n > 4$ (or $n = 3$) in most of this paper, and often require that the parity of n is known (e.g., to determine whether location discovery is solvable or not), we complement the paper by discussing the problem about determining the parity of n, and solving the considered problems when $n \leq 4$. These issues are presented in Section 7. Then, in Section 8 we discuss how our solutions can be applied to build efficient randomized algorithms. Connections between combinatorial structures introduced in Section 4 (distuinguishers) and the combinatorial search problem are discussed in Section 9.

Finally, conclusions and open problems are presented in Section 10.

2. Basic Properties of the Model

In this section we make a few observations regarding features and limitations of the model studied in the paper. First, we provide tools allowing agents to infer some knowledge from information about its traversed distances and observed positions of collisions. Then, basic lower bounds on the complexity of location discovery are stated.

Lemma 2. *All agents can determine in $O(1)$ rounds whether a rotation index of a given round is 0, $n/2$, larger than $n/2$ or smaller than $n/2$ (according to their own senses of directions).*

Proof. Assume that an algorithm runs two consecutive rounds with the same directions dir_a of all agents. Then, the sum s of distances $\mathrm{dist}()$ on which an agent a is shifted in these rounds is larger than 1 if and only if the rotation index of such a round is larger than $n/2$. Similar relationships hold for other values of s and rotation indexes.

To give the reader more intuition, let us discuss the relationship between the rotation index and the distances traversed by agents in more detail. Consider an agent a_1 and assume that its sense of direction is coherent with "objective" sense of direction. (The other cases can be analyzed analogously.) Moreover, let a_i denote the agent in right ring distance $i-1$ from a_1, and let p_i for $i \in [n]$ be the initial position of a_i, where $0 = p_1 < p_2 < \cdots < p_n < 1$. Let us fix directions of agents arbitrarily and let $r < n$ be the rotation index corresponding to these fixed directions. Now consider two consecutive rounds in which agents start moving with these fixed directions. Then, according to Lemma 1, the position of a_1 at the end of the former round is p_{i_1} for $i_1 = 1 + r \bmod n$. Moreover, the position of a_1 at the end of the latter round is p_{i_2} for $i_2 = 1+(i_1+r-1) \bmod n$. Thus,

- $0 < r < n/2$ iff $0 < p_{i_1} < p_{i_2} < 1$,
- $r = 0$ iff $p_{i_1} = p_{i_2} = 0$,
- $r = n/2$ iff $p_{i_1} \neq 0$ and $p_{i_2} = 0$, and
- $r > n/2$ iff $0 < p_{i_2} < p_{i_1} < 1$.

□

For a fixed set of agents A, we define the *rotation index* $\mathrm{RI}(B)$ of a set B as the rotation index of a round in which all elements of $B \cap A$ start the round moving right (clockwise) and the remaining agents start the round moving left (anticlockwise). (Note that we assume an objective sense of direction when talking about agents which start a round moving clockwise/anticlockwise.)

Lemma 3. *(a) $RI(B) = 2|B|$ mod n.*

(b) $RI(B) = 0$ if and only if $|B| \in \{0, n/2, n\}$.

(c) If $RI(B) \neq 0$, then $0 < |B| < n$.

(d) If $RI(B) \neq 0$, and $B = B_1 \cup B_2$ for disjoint B_1, B_2, then $RI(B_1) \neq 0$ or $RI(B_2) \neq 0$.

Proof. For (a), it is sufficient to observe that $\mathrm{RI}(B) = (|B| - (n - |B|)) \bmod n = 2|B| \bmod n$.

Items (b) and (c) are obvious (see Lemma 1). Assume to the contrary that $\mathrm{RI}(B) \neq 0$ and $\mathrm{RI}(B_1) = \mathrm{RI}(B_2) = 0$ for a partition B_1, B_2 of B. Then, (b) and (c) imply that $|B_1|, |B_2| \in \{0, n/2, n\}$, $0 < |B| < n$, and $|B| \neq n/2$. Moreover, $|B_1| + |B_2| = |B|$. One can easily check that it is impossible to satisfy all these relationships simultaneously. □

Now, we make an observation regarding information which can be inferred by an agent using the distance between its starting position and the first collision in a round (i.e., coll()).

Proposition 4. *Assume that an agent b_0 starts moving in a round in the direction dir_{b_0}, and let consecutive agents in the direction dir_{b_0} from b_0 be denoted $b_1, \ldots, b_{n-1}$. Moreover, let the geometric distance (on the ring) between b_{i-1} and b_i be x_{i-1}. If $b_1, \ldots, b_k$ start the round in the direction dir_{b_0} for $k < n-1$, and b_{k+1} starts in the opposite direction to dir_{b_0}, then the relative position of the first collision of b_0 is equal to $(x_0 + x_1 + \cdots + x_k)/2$.*

Proof. One can easily prove by induction on $j \geq 0$ the following fact: b_{k-j} collides with b_{k-j+1} in distance $(x_{k-j} + \cdots + x_k)/2$ from the initial position of b_{k-j}. □

2.1. Lower bounds on the complexity of location discovery

As observed by Friedetzky et al. [5], location discovery cannot be solved in the basic model when n is even.

Lemma 5. *[5] It is impossible to solve the location discovery problem in the basic model with even n.*

The reason of this impossibility result follows from the fact that, when n is even, the rotation index of any round in the basic model is always even. Therefore, an agent can only visit positions of agents having even ring distance from itself.

Below, we make an observation regarding lower bounds on the complexity of location discovery.

Lemma 6. *For each large enough natural number n, the following statements hold:*

1. *The location discovery problem in the basic and lazy model cannot be solved in less than $n-1$ rounds in the worst case.*

2. *The location discovery problem in the perceptive model cannot be solved in less than $(n-1)/2$ rounds in the worst case.*

Proof. The goal of each agent is to determine real numbers $x_0, \dots, x_{n-1}$, i.e. the distances between consecutive agents. At the beginning of the procedure, it is only known that $\sum_{i=0}^{n-1} x_i = 1$. The only information which an agent gets in a round, is the sum $x_i + x_{(i+1)\text{mod } n} + \cdots + x_{j\text{mod } n}$ for some $0 \le i, j < n$ corresponding to the relative distance between its starting and final positions in a round, i.e. the value of dist() (see Lemma 1). In the perceptive model, an agent gets additionally another linear combination, coll(), of $x_0, \dots, x_{n-1}$ equal to the distance to the first collision in a round. This number is 1/2 times the sum $x_i + x_{(i+1) \text{ mod } n} + \cdots + x_{j \text{ mod } n}$ for some $0 \le i, j < n$, by Proposition 4. Therefore, the basic facts from linear algebra imply that the necessary number of rounds to accomplish the task of location discovery is $n-1$ in the basic and lazy models and $(n-1)/2$ in the perceptive model. Indeed, otherwise an agent would have to determine $x_0, \dots, x_{n-1}$ uniquely from $m < n$ linear equations on variables $x_0, \dots, x_{n-1}$

One may argue that there is the following flaw in the above reasoning. As agents can adapt their behavior on the basis of partial knowledge about $x_0, \dots, x_{n-1}$, the set of linear equations is not fixed in advance. And, the agents can somehow encode information about their knowledge by choosing their initial direction in consecutive rounds. In order to address this nuance, we provide a formal proof using a counting argument.

For a given large enough natural number n, consider the set of binary words of length $n^2(n-1)$. For each such word w of length $n^2(n-1)$, let $w = w_1 w_2 \dots w_{n-1}$, where $|w_i| = n^2$ and let y_i be the natural number whose binary encoding is equal to w_i. (That is, $0 \le y_i < 2^{n^2}$.)

For a given word w as above and induced numbers $y_1, \dots, y_{n-1}$, consider a ring network with n agents such that the distance x_i between the agent a_i and the agent a_{i+1} for $1 \le i < n$ is equal to $x_i = \frac{1}{2^{2n^2}} \cdot (1 + y_i)$. Finally, the distance x_0 between a_n and a_1 is equal to $1 - \sum_{i=1}^{n-1} x_i$. As observed earlier, in a round, a_i learns the value

$$x = x_i + x_{(i+1) \bmod n} + \dots + x_{j \bmod n}$$

for some $0 \le i, j < n$. If x_0 is not a summand in x, the number x learnt by a_1 is equal to a natural number smaller than $n \cdot 2^{n^2}$ multiplied by $\frac{1}{2^{2n^2}}$. Thus, the result can be encoded in at most $\log(n \cdot 2^{n^2}) < n^2 + 2\log n$ bits. Similarly, if x_0 appears as a summand in x, then $1 - x$ is equal to a natural number smaller than $n \cdot 2^{n^2}$ multiplied by $\frac{1}{2^{2n^2}}$. Thus again, x can be encoded in at most $n^2 + 2\log n$ bits. Therefore for a_1, there are at most

$$2^{r(n^2+2\log n)}$$

different results of an execution of r rounds of the location discovery algorithm. On the other hand each execution on an instance corresponding to a binary word w of length $n^2(n-1)$ uniquely determines the distances x_i and therefore uniquely determines the word w. Thus, the number of rounds r has to satisfy the inequality

$$r(n^2 + 2\log n) \ge \log(2^{n^2(n-1)}) = n^2(n-1)$$

which implies that the number of rounds $r \ge n - 1$ for sufficiently large n. Similarly, one can argue that at least $(n-1)/2$ rounds are necessary in the perceptive model. $\square$

3. Reductions between considered problems

In this section we establish reductions between the coordination problems. The results proved in this section are illustrated in Figures 1 and 2, and are summarized in Theorem 7. They work for arbitrary n, provided $n > 4$ or $n = 3$.

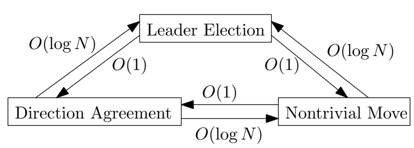

Figure 1: Complexity of reductions among coordination problems if n is odd or the model is either perceptive or lazy.

Theorem 7. *For each model considered in the paper (basic, lazy, perceptive) the asymptotic complexity of all coordination problems (direction agreement, leader election, nontrivial move) are equal up to an additive term $O(\log N)$, provided $n > 4$ or $n = 3$.*

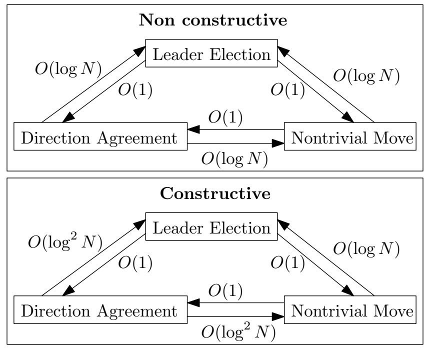

Figure 2: Complexity of reductions among coordination problems in the basic model (even n).

3.1. The setting with the nontrivial move problem solved

In this section, we assume that the nontrivial move problem is solved.

Lemma 8. *If the nontrivial move problem is solved, the direction agreement problem can be solved in $O(1)$ rounds, also in the case that agents do not have assigned IDs.*

Proof. Recall that, given a nontrivial move t, each agent can check whether the rotation index of t is larger than $n/2$ or smaller than $n/2$ according to its sense of direction using Lemma 2 (note that the rotation index of a nontrivial move is not 0 nor $n/2$). Note that agents a_1, a_2 with opposite senses of direction experience rotation indexes r and $n - r$ respectively for some $r < n$. Thus, the agents a_1, a_2 experience the rotation index of t as larger than $n/2$ iff a_1 and a_2 have the same sense of direction. Therefore, if agents experiencing the rotation index of t as larger than $n/2$ change their sense of direction, the direction agreement problem is solved. □

For greater clarity, we provide a pseudocode of the above described solution as Algorithm 1.

Algorithm 1 DirAgr(a)

Assign dir_a as in a nontrivial move ▷ Assumption: nontrivial move solved
SINGLEROUND
$d_1 \leftarrow \mathrm{dist}()$
SINGLEROUND
$d_2 \leftarrow \mathrm{dist}()$
if $d_1 + d_2 > 1$ **then**
    change sense of direction

Lemma 9. *Assume that the nontrivial move problem is solved. Then, it is possible to solve the leader election problem in $O(\log N)$ rounds.*

Proof. The idea of our solution is as follows. Given a nontrivial move, we solve the direction agreement problem in time $O(1)$ (Lemma 8). Then, let X be the set of agents which start moving right in a known nontrivial move (according to agreed common sense of direction). Thus, $0 < |X| < n$ and we choose X as the initial set of candidates for the leader. Then we iterate over consecutive bits of IDs and gradually decrease the set of candidates by fixing consecutive bits of candidates. More precisely, in the ith step, we check whether the rotation index of $X_0 = \{b \,|\, b \in X, \mathrm{ID}_b[i] = 0\}$ or $X_1 = \{b \,|\, b \in X, \mathrm{ID}_b[i] = 1\}$ is non-zero. By Lemma 3(d), at least one of them has nonzero rotation index and we restrict X to the corresponding subset (X_0 or X_1). In this way, we eventually get a nonempty set of agents with all bits of IDs fixed, i.e., the set with one element (the leader). □

For greater clarity, we provide a pseudocode as Algorithm 2.

Algorithm 2 LeaderWithNMove(a)

Solve the direction agreement problem ▷ assumption: nontrivial move solved; see Lemma 8
$X \leftarrow$ all agents starting right in a nontrivial move
for $i = 1, 2, \ldots, \lceil \log(N+1) \rceil$ **do**
    $X_0 \leftarrow \{b \,|\, b \in X, \mathrm{ID}_b[i] = 0\}$ ▷ i.e., set $a \in X_0$ iff $a \in X$ and $\mathrm{ID}_a[i] = 0$
    if $\mathrm{RI}(X_0) \neq 0$ **then**
        $X \leftarrow X_0$ ▷ i.e., set $a \in X$ iff $a \in X_0$
    else
        $X \leftarrow X \setminus X_0$ ▷ i.e., set $a \in X$ iff $a \notin X_0$
Set the status of a as *leader* iff $a \in X$.

3.2. The setting with the chosen leader

In this section, we assume that (exactly) one agent in a network has the status "leader".

Lemma 10. *If the leader is chosen, one can solve the nontrivial move problem in $O(1)$ rounds.*

Proof. Assume that the leader a is chosen. Consider two assignments of directions: (1) $\mathrm{dir}_b = \mathrm{right}$ for each $b \in A$ and (2) $\mathrm{dir}_b = \mathrm{right}$ for each $b \neq a$ and $\mathrm{dir}_a = \mathrm{left}$. The rotation indexes r_1, r_2 of such two rounds differ by 2 modulo n (Lemma 1). As $n > 4$ or $n = 3$, at least one of two numbers which differ by 2 modulo n does not belong to $\{0, n/2\}$. Thus, the nontrivial move problem is solved. □

Corollary 11. *If the leader is chosen, one can solve the direction agreement problem in $O(1)$ rounds.*

Proof. Given the leader, we obtain a nontrivial move in $O(1)$ rounds (Lemma 10). Next, we apply the solution from Lemma 8 to obtain a common sense of direction in $O(1)$ rounds. □

3.3. The setting with the common sense of direction

In this section we consider the setting that agents have the common sense of direction. We show simple efficient solutions for leader election and nontrivial move in the basic model which rely on a subroutine for emptiness testing. Recall this problem was defined in Section 1.4.

Lemma 12. *Assuming all agents share a common sense of direction, the emptiness testing problem can be solved in $\lceil \log(N+1) \rceil$ rounds in the basic model, and in one round in the lazy and perceptive model. Moreover, if n is odd, the emptiness testing is solvable in one round in the basic model as well.*

Proof. First, assume that the considered model is lazy/perceptive or n is odd. In order to test whether $B' = B \cap A$ is empty:

- every $a \in B$ moves right,
- other agents move left in the basic model and perceptive model (they cannot choose "idle" in these models); and they start as idle agents in the lazy model.

If the agents' positions at the end of the round with these directions are different from their starting positions, then $B' \neq \emptyset$. Otherwise,

- in the basic/perceptive model: either $|B'| = 0$, or $|B'| = n/2$ or $|B'| = n$.
- in the lazy model: either $|B'| = 0$ or $|B'| = n$.

As for the distinction between the cases $|B'| = 0$ and $|B'| = n$, notice that each agent $a \in A$ knows whether $\mathrm{ID}_a \in B'$. Thus, it can also distinguish whether $|B'| = 0$: if $\mathrm{ID}_a \in B'$ then $|B'| = n$ and $|B'| = 0$ otherwise. This settles the problem in the lazy model, where we knew that either $|B'| = 0$ and $|B'| = n$ (when the position of an agent at the beginning of the round and at the end of the round are equal).

In the basic and perceptive model, it might be the case that $|B'| = n/2$ and thus the elements from $A \setminus B'$ are still not able to distinguish the cases $|B'| = 0$ and $|B'| = n/2$: However,

- Perceptive model:

  If $|B'| = n/2$, then each agent has at least one collision during a round, while there are no collisions if $|B'| = 0$ or $|B'| = n$ (as all agents start the round with the same direction). Thus, in the perceptive model, an agent can distinguish the case $|B'| = n/2$ from $|B| \in \{0, n\}$ by observing, whether it has noticed a collision at all during a round.

- Basic model:

  Here, we assume that n is odd and therefore the case $|B'| = n/2$ does not happen. Thus, each agent a knows whether $|B'| > 0$ if and only if $\mathrm{ID}_a \in B$.

Regarding even n and the basic model, we apply Algorithm 3. It relies on the fact that, if $|B'| = n/2$, there is $i \in \lceil \log(N+1) \rceil$ and $j \in \{0, 1\}$ such that $0 < |\{a \mid a \in B', \mathrm{ID}_a[i] = j\}| < n/2$. As before, let $B' = B \cap A$. Note that

Algorithm 3 Emptiness(a, B) $\triangleright$ assumption: even n, basic model, comm. direction

If $a \in B$ then $\mathrm{dir}_a \leftarrow$ right else $\mathrm{dir}_a \leftarrow$ left
SingleRound
If dist() $\neq 0$: return $B \cap A$ is not empty
for $i = 1, 2, \ldots, \lceil \log(N+1) \rceil$ **do**
    If $\mathrm{ID}_a[i] = 1$ and $a \in B$ then $\mathrm{dir}_a \leftarrow$ right else $\mathrm{dir}_a \leftarrow$ left
    SingleRound
    If dist() $\neq 0$: return $B \cap A$ is not empty
If $a \in B$ return $B \cap A$ is not empty else return $B \cap A$ is empty.

if dist() $\neq 0$ after the first round, then B' is certainly not empty. Otherwise, $|B'| \in \{0, n, n/2\}$. Each agent $a \in B$ knows that $|B'| > 0$, so we only need to allow the agents outside B to distinguish between $|B'| = 0$ and $|B'| = n/2$ (note that no agents outside B attend the protocol if $|B'| = n$). If $|B'| = n/2$, then there exists a bit $i \in [\lceil \log(N+1) \rceil]$ such that $0 < |\{a \mid a \in B', \mathrm{ID}_a[i] = 1\}| < n/2$, and therefore the ith round in the for-loop gives a nonzero rotation index, i.e., dist() $\neq 0$ for each agent after this round. On the other hand, if $|B'| = 0$, each round of the for-loop will give the rotation index 0. (Note that, as before, from the point of view of an external observer, the cases $|B'| = 0$ and $|B'| = n$ might be indistinguishable, since it is possible that each round gives rotation index 0 when $|B'| = n$. However, as agents know that $|B'| \neq n/2$ and $|B'| \in \{0, n\}$, it is enough that they check if their own ID is in B.) $\square$

With help of the emptiness testing protocol, we devise a solution to the leader election problem. The idea of our solution is based on a binary search approach similar to that from Lemma 9. The main obstacle here is that, without a nontrivial move, the initial set of candidates for the leader is just $X = A$ and it has size n, thus its rotation index is 0. And, the case that it is split in two subsets X_1, X_2 of size $n/2$ is indistinguishable from the case that it is split in $X_1 = X$ and $X_2 = \emptyset$ (or vice versa), at least on the basis of rotation indexes of appropriate sets. Therefore, we use the more sophisticated emptiness testing from Lemma 12.

Lemma 13. *Assuming all agents share common sense of direction, the leader election problem can be solved in $O(\log^2 N)$ rounds in the basic model (with even n) and in $\lceil \log(N+1) \rceil$ rounds in other settings.*

Proof. We implement a binary search approach with help of the emptiness testing, where each round fixes one bit in the binary representation of a (nonempty) set of candidates for the leader. For completeness we describe an algorithm which employs this idea as Algorithm 4. The proof relies on the fact that $X \neq \emptyset$

Algorithm 4 Leader(a) $\triangleright$ assumption: common sense of direction

```
X ← all agents                                   ▷ i.e., set a ∈ X
for i = 1, 2, ..., ⌈log(N+1)⌉ do
    Y ← {b | b ∈ X, ID_b[i] = 0}      ▷ i.e., set a ∈ Y iff a ∈ X and ID_a[i] = 0
    if Y is not empty then             ▷ use Emptiness(a,Y) – Lemma 12
        X ← Y                          ▷ i.e., set a ∈ X iff a ∈ Y
Set the status of a as leader iff a ∈ X.
```

after each iteration and it contains element with fixed leftmost i bits of their IDs after the ith iteration of the for-loop. Thus, all bits of IDs of elements of X are fixed after the for-loop. Thus, $|X| = 1$ by uniqueness of IDs. □

An efficient solution for the nontrivial move problem can be easily obtained from Lemma 13 and Lemma 10.

Corollary 14. *If all agents have the same sense of direction, the nontrivial move problem can be solved in $O(\log^2 N)$ rounds in the basic model (with even n) and in $\lceil \log(N+1) \rceil$ rounds in other settings.*

Below we show that the nontrivial move problem can also be solved in $O(\log N)$ rounds in the basic model with even n, thus strengthening the $O(\log^2 N)$ from Corollary 14 for the basic model, and matching the bound from this corollary for other models. However, the result in the following lemma is weaker, as this is based only on a nonconstructive proof using the probabilistic method.

Lemma 15. *If all agents have the same sense of direction, the nontrivial move problem can be solved in $O(\log N)$ rounds.*

Proof. We randomly choose sets $S_1, \ldots, S_k$ such that each $a \in [N]$ is an element of S_i with probability $1/16$, where all choices are independent. Assume that $n \geq 16$. Then, in round i, the agents from S_i move right and the remaining ones move left. The expected number of agents moving right is $n/16$ and, by Chernoff bounds, the actual number of agents moving right in a round is in $[n/32, 3n/32]$ with probability $\geq 1 - 2^{-\Theta(n)}$. Thus, the ith round gives a nontrivial move with probability $1 - 2^{-cn}$ for some constant c. Let $T_{N,k}$ be an event that a sequence $S_1, \ldots, S_k$ does not give a nontrivial move for at least one set $X \subset [N]$ of agents such that $|X| \geq 16$. Then,

$$\text{Prob}(T_{N,k}) < \sum_{j=16}^{N} \binom{N}{j} 2^{-ckj} \leq \sum_{j=16}^{N} 2^{j(\log(Ne/j) - ck)}.$$

The last inequality is using that $\binom{N}{j} \leq \left(\frac{N \cdot e}{j}\right)^j$.

If $k > 2(\log N)/c$, then $\log(Ne/j) - ck \leq -1$ and

$$\mathrm{Prob}(T_{N,k}) < \sum_{j=16}^{N} 2^{-j} < 1/2^{15}.$$

Finally, we inspect the case that $4 < n < 16$. Note that there are only polynomially many subsets of $[N]$ of size at most 15. More precisely, the number of such sets is at most

$$p_{16}(N) \leq \sum_{i=5}^{15} \binom{N}{i}.$$

Moreover, for each set A of size n in the range $[5, 15]$ and each i, the probability that S_i gives a nontrivial move is at least some positive constant c, independent of N. In particular, this probability is not smaller than the probability that exactly one element of A belongs to S_i which is

$$|A| \cdot \frac{1}{16}\left(1 - \frac{1}{16}\right)^{|A|} \geq \frac{5}{16}\left(1 - \frac{1}{16}\right)^{15} = c$$

for some constant c. Thus, one can derive a sufficiently large constant c' such that the probability that there is a set of size in the range $[5, 15]$ without a non-trivial move in $S_1, \ldots, S_{c' \log N}$ is at most

$$p_{16}(N) \cdot (1-c)^{c' \log N} < \frac{1}{2}.$$

The above facts give, by the probabilistic method, the result stated in the lemma. □

Finally observe that Lemma 9 and Lemma 15 give the following reduction from the direction agreement problem to the leader election problem.

Corollary 16. *Assume that the direction agreement is solved. Then, the leader election problem can be solved (non-constructively) in $O(\log N)$ rounds.*

3.4. Application of coordination problems for location discovery

Given the reductions summarized in Figure 1 and Figure 2 (see also Theorem 7), one can simply solve the location discovery problem in the lazy model, irrespective of the parity of n, or in the basic model for odd n.

Lemma 17. *Assume that (at least) one among the following problems is solved: nontrivial move, leader election, direction agreement. Then, location discovery can be solved in $n + O(\log N)$ rounds in the lazy model with arbitrary n and in the basic model with odd n.*

Proof. Given a solution to any of the coordination problems, we can solve the leader election problem and the direction agreement problem in $O(\log N)$ rounds (see Figures 1 and 2).

Then, in the lazy model, the leader starts every round moving right and other agents start as idle. The rotation index is equal to 1 and each agent knows the relative positions of all other agents after n rounds.

In the basic model with odd n, each agent sets its direction to left, except the leader which sets the direction to right. The rotation index of each round is equal to 2 in such a case which implies that if we repeat SINGLEROUND with this setting several times, each agent gets back to its original position after exactly n rounds and it can determine the initial positions of all other agents at this moment, on the basis of distances between its positions at the end of consecutive rounds. Indeed, if we denote the distances by $x_1, x_2, \ldots, x_n$, where x_1 is the distance to the closest agent in the direction right, then the consecutive distances are $x_1 + x_2, x_3 + x_4, \ldots, x_n + x_1$ in the first $\lceil n/2 \rceil$ rounds and $x_2 + x_3, x_4 + x_5, \ldots, x_{n-1} + x_n$ in the next $\lfloor n/2 \rfloor$ rounds. □

Note that the above result for the basic model applies in the stronger perceptive model as well. However, we provide more efficient solutions for this model later.

3.5. Solutions for the case that n is odd

In this section we study the actual complexity of our considered problems in the case that n is odd.

The crucial difference between the cases of odd and even n follows from the following observation: If $n_C \neq 0$ and $n_A \neq 0$ in a round then the round is nontrivial in the case of odd n. On the other hand, this is not necessarily the case for even n, as, e.g., $0 \neq n_C = n_A = n/2$ or $n_C \in \{\frac{3}{4}n, \frac{1}{4}n\}$, $n_A = n - n_C$ do not give a nontrivial move. (Recall from Section 1.4 that a rotation index of $n/2$ is considered a trivial move.)

Proposition 18. *The direction agreement problem can be solved in $O(1)$ time in the basic model, provided n is odd.*

Proof. Observe that, if n is odd, the rotation index of a round is zero only when all agents start the round moving in the same direction (since $n/2 \notin \mathbb{N}$). Thus, one can "test" whether all agents share the same sense of direction in a round in which each agent a starts with direction dir_a=right. Then if the rotation index of such round is zero, all agents have the same sense of direction. Otherwise, this setting of directions gives a nontrivial move (as n is odd). Thus, we can apply Lemma 8 to obtain a common sense of direction. □

Corollary 19. *If the number of agents n is odd, the leader election problem and the nontrivial move problem can be solved in time $O(\log N)$. The location discovery problem can be solved in $n + O(\log N)$ rounds.*

Proof. The complexity of the leader election problem and the nontrivial move problem follows from Proposition 18 and Theorem 7 while the complexity of location discovery follows from Proposition 18 and Lemma 17. □

Below we provide a slightly modified variant of a solution for the nontrivial move problem, reducing the complexity from $O(\log N)$ to $O(\log(N/n))$.

Proposition 20. *The time complexity of the nontrivial move problem in the basic model with odd n is $\Theta(\log(N/n))$.*

Proof. Since n is odd, the rotation index cannot be equal to $n/2$ and thus it is sufficient to obtain any rotation index not equal to 0. As observed in Proposition 18, an assignment dir_a =right for each a gives a nontrivial move if and only if the agents do not share a common sense of direction. Thus, in a round with such assignment of directions, either we obtain a nontrivial move (problem solved) or we know that all agents share the same sense of direction. In the latter case, it is sufficient to split A in two nonempty subsets A_1, A_2 such that the elements of A_1 start a round moving right and the elements of A_2 start a round moving left. We split agents on the basis of consecutive bits of the binary encoding of their IDs (of length $\lceil \log(N+1) \rceil$). We succeed after finding i such that the set of agents with 0 on the ith bit and the set of agents with 1 on the ith bit are nonempty. We use the observation that there are at most $\lceil \log(N/n) \rceil$ bits such that all agents share the same value of IDs on those bits. Thus, assume that the sets $\{a \,|\, \mathrm{ID}_a[i] = 0\}$ and $\{a \,|\, \mathrm{ID}_a[i] = 1\}$ are not empty for some $i \in [\lceil \log(N+1) \rceil]$. If agents share the same sense of direction, and an agent a chooses direction right if and only if $\mathrm{ID}_a[i] = 0$, then we get a nontrivial move after inspecting at most $\lceil \log(N/n) \rceil$ bits. The following algorithm formalizes this idea.

Algorithm 5 NMove(a) ▷ assumption: odd n

$\mathrm{dir}_a \leftarrow$ right
SingleRound
$d_1 \leftarrow \mathrm{dist}()$
if $d_1 \neq 0$ **then**
  return
else
  $X \leftarrow$ all agents ▷ i.e., set $a \in X$
  for $i = 1, 2, \ldots$ **do**
    $Y \leftarrow \{b \,|\, b \in X, \mathrm{ID}_b(i) = 0\}$
    if $a \in Y$ then dir$\leftarrow$ right else dir$\leftarrow$ left
    SingleRound
    If $\mathrm{dist}() \neq 0$ return current value of dir ▷ Nontrivial move

As for the lower bound, assume that all agents share the same sense of direction and we have an algorithm $\mathcal{A}$. Let X_i be the largest set of IDs such that all elements of X_i choose the same direction in the first i rounds of $\mathcal{A}$. Then, $|X_i| \geq N/2^i$. If $i < \log(N/n)$, then $|X_i| > n$ which implies that $\mathcal{A}$ does not give a nontrivial move if the set of IDs is equal to X_i. $\square$

4. Basic model with even n

It is known (Lemma 5) that the location discovery problem cannot be solved in the basic model when n is even. However, we can still try to solve other coordination problems. We show in this section that, if n is even, the complexities of these problems are significantly larger than for the case of odd n.

First, we define a combinatorial notion of a *distinguisher*. Then, a relationship between the size of a distinguisher and the complexity of the corresponding nontrivial move problem is established. Finally, tight bounds on the smallest size of distinghuishers and the complexity of the nontrivial move problem are showed.

Definition 21. *We say that a family $\mathcal{S} = \{S_1, \ldots, S_k\}$ of subsets of $[N]$ is a (N, n)-*distinguisher *of size k if for each $X_1, X_2 \subseteq [N]$ such that $|X_1| = |X_2| = n$ and $X_1 \cap X_2 = \emptyset$, there exists $i \in [k]$ such that $|S_i \cap X_1| \neq |S_i \cap X_2|$.*

Definition 22. *Let $N \in \mathbb{N}$ and let $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a nondecreasing function. A family $\mathcal{S} = S_1, \ldots, S_{f(N,N)}$ of subsets of $[N]$ is a* strong (N, f)-distinguisher *if the prefix $S_1, \ldots, S_{f(N,n)}$ of $\mathcal{S}$ is a (N, n)-distinguisher for each $n \leq N/2$.*

The *weak nontrivial move* problem is to assign to each agent a a direction dir_a such that if a starts a round in the direction $\text{dir}_a \in \{\text{right}, \text{left}\}$, then the rotation index r in the round is not equal to 0. (A round with the rotation index $n/2$ is treated as a weak nontrivial move, which is not the case in the standard definition of a nontrivial move.)

In what follows, we show a reduction between the complexity of the weak nontrivial move problem and the smallest size of a distinguisher.

Proposition 23. *Let $n > 4$ be an even number and $N \geq n$.*

1. *Assume that a protocol $\mathcal{A}$ solves the weak nontrivial move problem in the basic model in at most $f(N, n)$ rounds when the value of n is* known *to the agents. Then, there exists a $(N, n/2)$-distinguisher of size $f(N, n)$.*
2. *Assume that a protocol $\mathcal{A}$ solves the weak nontrivial move problem in the basic model in at most $f(N, n)$ rounds when the actual value of n is* unknown *to the agents. Then, there exists a strong (N, f')-distinguisher for $f'(N, n/2) = f(N, n)$.*

Proof. First, assume that n is known and $\mathcal{A}$ solves the weak nontrivial move problem. Observe that, until the first round of $\mathcal{A}$ with a weak nontrivial move, the only information available to each agent is that its starting position in a round is equal to its position at the end of a round. Thus, its behavior can be defined by a sequence of sets $S_1, S_2, \ldots$, such that the agent a chooses direction right in round i (provided no nontrivial move appeared before) if and only if $a \in S_i$. Let us fix an arbitrary sense of direction as "correct". Then, consider the situation in which the set of agents X_1 with the correct sense of direction and the set of agents X_2 with the incorrect sense of direction satisfy $|X_1| = |X_2| = n/2$.

Let $m_1 = |X_1 \cap S_i|$, $m_2 = |X_2 \cap S_i|$. Then, the rotation index (mod n) in round i is

$$\begin{aligned} & (|X_1 \cap S_i| + |X_2 \setminus S_i|) - (|X_1 \setminus S_i| + |X_2 \cap S_i|) \\ = \; & (m_1 + n/2 - m_2) - (n/2 - m_1 + m_2) \\ = \; & 2(m_1 - m_2). \end{aligned}$$

And therefore the ith round of $\mathcal{A}$ gives a (weak) nontrivial move if and only if $2(m_1 - m_2) \notin \{0, n\}$, which implies $m_1 \neq m_2$. On the other hand, $m_1 \neq m_2$ is equivalent to the fact that S_i distinguishes X_1 and X_2. In conclusion, the sequence $S_1, S_2, \ldots, S_{f(N,n)}$ defining $\mathcal{A}$ is a $(N, n/2)$-distinguisher.

For unknown n, the result follows from the above reasoning and the fact that $\mathcal{A}$ has to tackle arbitrary even $n \leq N/2$ which reflects the difference between a standard (N, n)-distinguisher and its strong counterpart. □

Now, we provide a lower bound on the size of a strong (N, n)-distinguisher with a simple proof based on a counting argument (a similar bound in another context was given e.g. in [24]). Although this result is subsumed by Lemma 25, we provide it to give some intuition before a more complicated, and less intuitive, proof of Lemma 25.

Lemma 24. *If $\mathcal{S}$ is a strong (N, f)-distinguisher for any $N > 4$ and any nondecreasing function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, then $f(N, n) = \Omega\left(\frac{n \log(N/n)}{\log n}\right)$.*

Proof. First, we show that a strong (N, f)-distinguisher $\mathcal{S}$ satisfies the property that for each two different sets $X_1, X_2 \subset [N]$ such that $|X_1| = |X_2| = n$, there exists $i \leq f(N, n)$ such that $|X_1 \cap S_i| \neq |X_2 \cap S_i|$ (note that X_1 and X_2 do not have to be disjoint!). Indeed, assume to the contrary that this is not the case for $\mathcal{S}$, and thus $|X_1 \cap S_i| = |X_2 \cap S_i|$ for some different sets X_1, X_2 of size $n > 1$ and each $i \in [f(N, n)]$. Let $Y_1 = X_1 \setminus X_2$ and $Y_2 = X_2 \setminus X_1$. Then, $Y_1 \cap Y_2 = \emptyset$, $|Y_1| = |Y_2| \leq n$ and $|Y_1 \cap S_i| = |Y_2 \cap S_i|$ for each $i \in [f(N, n)]$. This implies that $\mathcal{S}$ is not a strong (N, f)-distinguisher, which is a contradiction.

Let $\mathcal{S} = (S_1, \ldots, S_k)$ be a strong (N, f)-distinguisher. The above observation implies that, for any $X \neq X'$, $X, X' \subset [N]$ of size n, the sequences $|X \cap S_1|, \ldots, |X \cap S_k|$ and $|X' \cap S_1|, \ldots, |X' \cap S_k|$ are not equal, where $k = f(N, n)$. As each S_i gives at most $n + 1$ possible values of $|X \cap S_i|$ for $X \subset [N]$ of size n (i.e., the size of $X \cap S_i$ is in $[0, n]$), there are $\binom{N}{n}$ subsets of $[N]$ of size n and the sequence $|X \cap S_1|, \ldots, |X \cap S_k|$ has to be unique for each $X \subset [N]$ of size n, the length k of a strong (N, f)-distinguisher is at least

$$k \geq \log_{n+1} \binom{N}{n} = \Omega\left(\frac{\log \binom{N}{n}}{\log(n+1)}\right) = \Omega\left(\frac{n \log(N/n)}{\log n}\right)$$

for $n > 1$. For the last equality we use the relation $\binom{N}{n} \geq \left(\frac{N}{n}\right)^n$. □

It turns out that the result of Lemma 24 can be strengthened, that is, we show that the same asymptotic lower bound holds for a standard distinguisher as well. However, our proof of this fact is much more complicated. It applies techniques from [25], designed for proving lower bounds on size of selective families. We stress here that the lower bound for a strong variant of a distinguisher does not imply an analogous lower bound for a "standard" variant of a distinguisher. As observed in the proof of Lemma 24, the prefix of size $f(N, n)$ of a strong (N, f)-distinguisher gives an opportunity to "distinguish" **each** pair of sets of size n. On the other hand, a standard (N, n)-distinguisher is supposed to give a difference only on **disjoint** sets of size n.

Lemma 25. *If $\mathcal{S}$ is a (standard) (N, n)-distinguisher for $N > 2$ and $n \le N/128$ where n is a natural power of two, then the size of $\mathcal{S}$ is $\Omega\left(\frac{n \log(N/n)}{\log n}\right)$.*

Proof. Let us first stress that the calculations from the previous lemma do not apply here, since a ("standard") distinguisher does not have to "distinguish" small sets, so it does not have to distinguish non-disjoint sets of size n either.

In the proof, we use a notion from [25] and [26]:

Definition 26. *[25, 26] Let $l \le k \le n$. A family $\mathcal{F}$ of k-subsets (i.e. subsets of size k) of $[N]$ is (N, k, l)-intersection free if $|F_1 \cap F_2| \ne l$ for every $F_1, F_2 \in \mathcal{F}$.*

Fact 27. *[25, 26] Let $\mathcal{F}$ be an $(N, k, k/2)$-intersection free family where k is a power of 2 and $k \le N/64$. Then,*

$$\log |\mathcal{F}| \le \frac{11k}{12} \log(N/k).$$

Assume that n is a natural power of two. Let $G(V, E)$ be a graph, whose vertices are all $2n$-subsets of $[N]$, where the edges connect vertices corresponding to sets which have exactly n common elements. That is, $(X_1, X_2) \in E$ for $X_1, X_2 \in V$ if and only if $|X_1 \cap X_2| = n$. Let $\alpha(G)$ and $\chi(G)$ denote the size of the largest independent set of G and the chromatic number of G, respectively. We claim that

$$\begin{aligned} \log \chi(G) &\ge \frac{1}{6} n \log(N/(2n)) \quad \text{and} && (1) \\ \log \chi(G) &\le |\mathcal{S}| \log(2n + 1). && (2) \end{aligned}$$

Proof of (1):
We use the fact that $\chi(G) \ge \frac{|V|}{\alpha(G)}$. Moreover, as each independent set of G is a $(N, 2n, n)$-intersection free family of sets, Fact 27 implies that

$$\log \alpha(G) \le \frac{11}{12} \cdot (2n) \cdot \log(N/(2n)) \le \frac{22}{12} n \log(N/(2n)).$$

Therefore

$$\begin{aligned} \log \chi(G) &\ge \log |V| - \log \alpha(G) \\ &\ge \log \binom{N}{2n} - \tfrac{22}{12} n \log(N/(2n)) \\ &\ge 2n \log(N/(2n)) - \tfrac{22}{12} n \log(N/(2n)) \\ &= \tfrac{1}{6} n \log(N/(2n)), \end{aligned}$$

which gives (1). In the third inequality, we use the relation $\binom{a}{b} \geq \left(\frac{a}{b}\right)^b$.
Proof of (2):
Let $\mathcal{S} = (S_1, \ldots, S_m)$ be a (N, n)-distinguisher. Observe that, for any two sets X_1, X_2 of size $2n$ such that $|X_1 \cap X_2| = n$, there exists S_i such that $|S_i \cap X_1| \neq |S_i \cap X_2|$. Indeed, if there are X_1, X_2 of size $2n$ such that

$$(|S_1 \cap X_1|, \ldots, |S_m \cap X_1|) = (|S_1 \cap X_2|, \ldots, |S_m \cap X_2|),$$

and X_1, X_2 are neighbors in G, then $X_1 \setminus X_2$ and $X_2 \setminus X_1$ have size n and they are indistinguishable by $\mathcal{S}$ (which contradicts the assumption that $\mathcal{S}$ is a (N, n)-distinguisher).

The above property implies that, for any tuple $(p_1, \ldots, p_m), p_i \in [0, 2n]$, the set $\{X \mid \forall_i\ |S_i \cap X| = p_i\}$ is an independent set in $G(V, E)$. Therefore, as there are $(2n+1)^m$ sequences $(p_1, \ldots, p_m)$ such that each $p_i \in [0, 2n]$, the set of nodes of G can be split into $(2n+1)^m$ independent sets. It implies that $\chi(G) \leq (2n+1)^m$. Thus

$$\log \chi(G) \leq m \log(2n+1),$$

which proves (2).

Finally, observe that (1) and (2) imply the statement of the lemma. $\square$

Corollary 28. *Each algorithm solving the (weak) nontrivial move problem requires $\Omega(n \log(N/n)/\log n)$ rounds in the basic model with known value of n.*

Proof. If n is a natural power of two, the result follows directly from Proposition 23 and Lemma 25. As we analyze asymptotic complexity, n equal to natural powers of two gives an infinite set of instances establishing the claimed lower bound. $\square$

Now, using the probabilitstic method, we show that there exists a solution for the nontrivial move problem that nearly matches the lower bound from Corollary 28.

Theorem 29. *In the basic model, there exist solutions of the nontrivial move problem working in $O(n \log(N/n)/\log n)$ rounds for each $n \in [N]$ and $n > 4$, and also when n is unknown.*

Proof. Let us choose a sequence $\mathcal{S}$ of sets $S_1, S_2, \ldots$ probabilistically, such that each $x \in [N]$ belongs to S_i with probability $1/2$, where all choices are independent. Then, our algorithm is defined such that, in round i, the agents with IDs in S_i choose direction right and the other ones choose the direction left. We show that the family $\mathcal{S} = (S_1, \ldots, S_k)$ chosen in this way gives a protocol solving the nontrivial move problem with positive probability, provided the size n of the network is smaller than $N/3$. That is, the following event holds with positive probability: for each $X \subset [N]$ such that $|X| < N/3$, the nontrivial move appears during an execution of the prefix of $\mathcal{S}$ of size $O(n \log(N/n)/\log n)$, where $n = |X|$. Then we build a sequence $\mathcal{C}$ of size $O(N/\log N)$ which gives a nontrivial move on each $X \subset [N]$ of size at least $N/3$. Thus, by interleaving $\mathcal{S}$ and $\mathcal{C}$, the theorem holds thanks to the probabilistic method.

Let us fix a set of IDs $A \subset [N]$ of size n and assign sense of direction to them such that $A = A_c \cup A_i$, where A_c is the set of agents with correct sense of directions, $|A_c| = n_c$ and $|A_i| = n - n_c$. Recall that a round does *not* give a nontrivial move if and only if it is a $(0, n)$-round, $(n, 0)$-round, $(n/2, n/2)$-round, $(3n/4, n/4)$-round, or a $(n/4, 3n/4)$-round. (Also recall the definition of an "(a, b)-round" found at the start of Section 1.3.) The set of choices which give e.g. an $(n/2, n/2)$-round satisfies the property that the number of agents which

- have correct sense of direction and belong to S_i, OR
- have incorrect sense of direction and does not belong to S_i

is equal to $n/2$. Thus, the number of such choices of directions is equal to $\sum_{j=0}^{\min(n_c,n/2)} \binom{n_c}{j}\binom{n-n_c}{n/2-j}$. Other cases can be calculated analogously. Thus, for a round defined by S_i as above, we have:

$$\begin{aligned}
\mathrm{Prob}((n/2,n/2)\text{-round}) &= \tfrac{1}{2^n}\textstyle\sum_{j=0}^{\min(n_c,n/2)} \binom{n_c}{j}\binom{n-n_c}{n/2-j} \\
&= \tfrac{1}{2^n}\binom{n}{n/2} \le \tfrac{c_0}{n^{1/2}}, \\
\mathrm{Prob}((0,n)\text{-round}) &= \tfrac{1}{2^n}\binom{n_c}{0}\binom{n-n_c}{n-n_c} = \tfrac{1}{2^n}, \\
\mathrm{Prob}((n,0)\text{-round}) &= \tfrac{1}{2^n}\binom{n_c}{n_c}\binom{n-n_c}{0} = \tfrac{1}{2^n}, \\
\mathrm{Prob}((n/4,3n/4)\text{-round}) &= \tfrac{1}{2^n}\textstyle\sum_{j=0}^{\min(n_c,3n/4)} \binom{n_c}{j}\binom{n-n_c}{3n/4-j} \\
&= \tfrac{1}{2^n}\binom{n}{n/4} = 1/2^{\Theta(n)}, \text{ and} \\
\mathrm{Prob}((3n/4,n/4)\text{-round}) &= \mathrm{Prob}((n/4,3n/4)\text{-round}) \\
&= 1/2^{\Theta(n)}.
\end{aligned}$$

In the above calculations, we use the relationship that $\sum_{i=0}^{\min(a,c)} \binom{a}{i}\binom{b-a}{c-i} = \binom{b}{c}$ and Stirling's formula which determines the constant c_0 in the second line. The above estimations imply that

$$\mathrm{Prob}(\text{a round defined by } S_i \text{ is a trivial move for} |A| = n) \le c_1/\sqrt{n} \qquad (3)$$

for some constant c_1, provided n is large enough. Let us consider all sets of IDs A such that $|A| \in [2^{i-1}, 2^i)$, for i such that $2^i \le N/3$. Let

$$k = c\frac{2\log\binom{N}{2^i}}{i-1} \qquad (4)$$

for a large enough constant c whose value will be determined later. By E_i we denote the event that a sequence of sets $S_1, \ldots, S_k$ does *not* give a nontrivial move for all sets A whose size is in $[2^{i-1}, 2^i)$. Then,

$$\begin{aligned}
\mathrm{Prob}(E_i) &\le \textstyle\sum_{d=2^{i-1}}^{2^i} (\mathrm{Prob}(\text{triv. move on a set of size } d))^k \cdot \binom{N}{d} 2^d \\
&\le \textstyle\sum_{d=2^{i-1}}^{2^i} \frac{\binom{N}{d}2^d \cdot c_1}{2^{(i-1)k/2}} \le c_1 \sum_{d=2^{i-1}}^{2^i} \frac{\binom{N}{d}^2}{\binom{N}{2^i}^3} \\
&\le c_1 \textstyle\sum_{d=2^{i-1}}^{2^i} \frac{1}{\binom{N}{2^i}} \le c_1 \sum_{d=2^{i-1}}^{2^i} \frac{1}{2^{2^i}} < c_1 \frac{1}{2^i}.
\end{aligned}$$

In the above calculations, we use the following facts:

- $\binom{N}{d}2^d$ is the number of possible choices of sets of size d, and senses of direction of elements of these sets (used in the first inequality);
- Prob(triv. move on a set of size d) $\leq \frac{c_1}{\sqrt{d}} \leq \frac{c_1}{2^{(i-1)/2}}$ (used in the second inequality);
- $2^{(i-1)k/2} \geq \binom{N}{2^i}^c \geq \binom{N}{2^i}^3$ for $c \geq 3$ (which follows from (4); used in the third inequality);
- $2^d \leq \binom{N}{d}$ for $d \leq N/3$ (used in the third inequality);
- $\binom{N}{d} \leq \binom{N}{2^i}$ for $d \leq N/3$ (used in the fourth inequality);
- $\binom{N}{2^i} \geq (N/2^i)^{2^i} \geq 2^{2^i}$ if $2^i < N/2$ (used in the fifth inequality).

Let $i_0 = \lceil \log 4c_1 \rceil + 1$ and $i_1 = \lfloor \log(N/3) \rfloor$. The above calculations show that, the union of events $E_{i_0}, E_{i_0+1}, \ldots, E_{i_1}$ holds with probability at most $\sum_{i_0}^{i_1} c_1/2^i < 1/2$. Therefore, by the probabilistic method, there exists a sequence $\mathcal{S}$ that gives a nontrivial move for each set of IDs of size in $[2^{i_0}, 2^{i_1}] \subset [4c_1, N/c]$. It remains to tackle the cases that $n < 2^{i_0}$ and $n > 2^{i_1}$.

For the case that $n < 2^{i_0}$, let $c' = 2^{i_0}$. As c' is a constant independent of n, the number of sets of size $< c' = 2^{i_0}$ is polynomial with respect to N. The probability that a round gives a nontrivial move for a given set of size lec' is larger than some positive constant $p_{c'}$, independent of N. Indeed, the probability that there exists a set of size $< c' = 2^{i_0}$ without a nontrivial move in the prefix of $\mathcal{S}$ of length m is at most

$$\left(\sum_{i=1}^{c'} \binom{N}{i}\right) \cdot (1 - p_{c'})^m < \frac{1}{N}$$

for large enough $m = O(\log N)$. Therefore, on a sufficiently long prefix of $\mathcal{S}$ of length $O(\log N) = O(n \log(N/n)/\log n)$, the nontrivial move appears for each set of size $< 2^{i_0}$ with probability $1 - 1/N$.

Now, we consider the case that the size $n > 2^{i_1} > N/3$. The number of such sets is upper bounded by 2^N. And, for each such set, each round gives a nontrivial move with probability at least $c'/\sqrt{N}$ for a constant c'. By a simple calculation, one can show that the nontrivial move appears for each such set on a long enough prefix of $\mathcal{S}$ of size $O(N/\log N)$ with probability $1 - 1/N$. More precisely, on a prefix of $c''N/\log N$, the probability that there is a set without a nontrivial move is smaller than

$$2^N (c'/\sqrt{N})^{c''N/\log N} < 1/N$$

for a large enough constant c''.

□

Finally, the above bounds (Cor. 28, Th. 29) and the equivalence of complexities of coordination problem (Th. 7) lead to the following corollary.

Corollary 30. *The time complexity of the nontrivial move problem, the leader election problem, and the direction agreement problem in the basic model (with even n) is $\Theta(n \log(N/n)/\log n)$.*

Given the relationship between distinghuishers and the nontrivial move problem (Prop. 23), the lower bound from Lemma 25 and Cor. 30, we get the following bound.

Corollary 31. *The size of the smallest (N, n)-distinguisher for $N \geq n$ is $\Theta(n \log(N/n)/\log n)$.*

For each $N \in \mathbb{N}$, there exists a strong (N, f)-distinguisher for some $f(N, n) \in O(n \log(N/n)/\log n)$. Moreover, if $\mathcal{S}$ is a strong (N, f)-distinguisher, then $f(N, n) = \Omega(n \log(N/n)/\log n)$.

5. Lazy model with even n

In this section we show that all the results from the previous section apply for the lazy model as well. Thus, the additional possibility of choosing to stay idle at the beginning of a round does not change the asymptotic complexity of the coordination problems we consider.

Proposition 32. *If the nontrivial-move problem can be solved in $f(N, n)$ rounds in the lazy model then the weak nontrivial move problem can be solved in $2f(N, n)$ rounds in the basic model.*

Proof. Let $\mathcal{A}$ be a protocol solving the the nontrivial move problem in the lazy model. It can be defined by the sequence of pairs of sets (R_i, L_i) such that the agents from $R_i \cap A$ ($L_i \cap A$, resp.) move right (left, resp.) in the round i while the remaining ones stay idle at the beginning of the round, where A is the set of IDs of agents. Let $A = A_C \cup A_I$ be an arbitrary set of n agents, where A_C (resp. A_I) are the agents with the correct (resp. incorrect) sense of direction. Assume that $\mathcal{A}$ gives a nontrivial move for this configuration in round i. Then, the rotation index in round i is not equal to 0 nor to $n/2$. Let r_i be the actual difference between the number of agents which start that round moving in the clockwise direction and the number of agents which start that round moving in the anticlockwise direction. Then $r_i \bmod n \notin \{0, n/2\}$ and

$$r_i \notin \{0, n, -n, n/2, -n/2\}. \tag{5}$$

Now, we show how to build a protocol $\mathcal{A}'$ which simulates $\mathcal{A}$ in the basic model. (That is, we have to deal with the obstacle that agents cannot start rounds immobilized.) The algorithm $\mathcal{A}'$ tries to "simulate" the ith round of $\mathcal{A}$ in two rounds. In both rounds, the agents from $R_i \cap A/L_i \cap A$ move right/left, respectively. The elements of $X = A \backslash (R_i \cup L_i)$ start the round in right direction in the former round and in left direction in the latter round. Let $p = |X \cap A_C| - |X \cap A_I|$

be the difference between the number of agents with the correct sense of direction and those with the incorrect sense of direction in X. Then, the rotation index of $\mathcal{A}'$ is equal to $(r_i - p) \bmod n$ in the former round and $(r_i + p) \bmod n$ in the latter round. We claim that either $r_i + p \notin \{0, n, -n\}$ or $r_i - p \notin \{0, n, -n\}$ which in turn implies that one of the rounds gives a weak nontrivial move in the basic model. In order to check that $r_i + p \notin \{0, n, -n\}$ or $r_i - p \notin \{0, n, -n\}$, we assume that $r_i + p \in \{0, n, -n\}$, i.e., $p \in \{-r_i, n - r_i, -n - r_i\}$). Then, the possible values of $r_i - p$, are $2r_i, 2r_i - n, 2r_i + n$. If each of them belongs to the set $\{0, n, -n\}$, then $r_i \in \{0, n/2, -n/2\}$. This in turn contradicts (5). □

The above proposition (Proposition 32), earlier established results regarding the complexity of problems in the basic model (Corollary 30), and reductions among them (Theorem 7) give the following corollary.

Corollary 33. *The time complexity of the nontrivial move problem, the direction agreement problem, and the leader election problem in the lazy model (with even n) is $\Theta(n \log(N/n)/ \log n)$.*

Since an agent cannot infer any information about the initial positions of other agents without a nontrivial move (unless we consider the perceptive model), the bounds for coordination problems impose analogous bounds on the location discovery.

Corollary 34. *The time complexity of the location discovery problem in the lazy model is $n + \Theta(n \log(N/n)/ \log n)$.*

Proof. The lower bound follows from the lower bound $\Omega(n \log(N/n)/ \log n)$ on the nontrivial move problem (Corollary 33) and the lower bound on location discovery from Lemma 6. The upper bound is obtained by solving the leader election and direction agreement (Corollary 33), and then using the leader and the common sense of direction in order to solve the location discovery problem (Lemma 17). □

6. Perceptive model without common sense of direction

Since the basic model is too weak for the task of location discovery (when n is even), we considered the lazy model. Although one can solve location discovery in this model, the overhead cost for this problem is $\Omega(n \log(N/n)/ \log n)$. In [5], it is shown that location discovery can be solved in the perceptive model (i.e., when the position of the first collision in a round can be detected while each agent has to start the round moving to the right or left). In this section, we inspect efficiency of coordination problems as well as location discovery in this model. First, we show that the perceptive model gives an opportunity to exchange information between neighbors on a ring (Section 6.1). Then, we use this feature to build algorithms for the nontrivial move problem which break the lower bounds working in the basic model and the lazy model (Section 6.2). Finally, using these solutions as tools, we provide a solution for the positions discovery problem in time $n/2 + o(n)$ provided $\log N = o(\sqrt{n})$ which is optimal up to the $o(n)$ term (Section 6.3).

6.1. Communication on a ring

First, we discuss the following *neighbors discovery* task in which each agent a should:

- learn (relative) location of its left neighbor $\mathrm{Left}(a)$ and its right neighbor $\mathrm{Right}(a)$;
- determine whether $\mathrm{Left}(a)$ and $\mathrm{Right}(a)$ have the same sense of direction as a has.

Algorithm 6 solves this problem based on the fact that each two IDs differ on at least one bit. (Some calculations performed by agents are not explicitly described in the algorithm, they are discussed later.) In Algorithm 6, each execution of SingleRound is followed by ReversedRound in which each agent starts a round with the direction opposite to its local direction dir. This gives a guarantee that each agent starts each application of SingleRound at exactly the same position as its position before the execution of the algorithm (so, its distances to neighbours are the same as well).

Algorithm 6 NeighborDiscovery(a)

$\mathrm{dir}_a \leftarrow \mathrm{right}$ ▷ All agents choose direction right
SingleRound; $\mathrm{start}_{\mathrm{right}} \leftarrow \mathrm{coll}()$; ReversedRound
$\mathrm{dir}_a \leftarrow \mathrm{left}$
SingleRound; $\mathrm{start}_{\mathrm{left}} \leftarrow \{\mathrm{coll}()\}$; ReversedRound
$D_{\mathrm{left}} \leftarrow \{\mathrm{start}_{\mathrm{left}}\}$; $D_{\mathrm{right}} \leftarrow \{\mathrm{start}_{\mathrm{right}}\}$ ▷ distances to collisions
for $i = 1, 2, \ldots, \lceil \log(N+1) \rceil$ **do**
    for $k \in \{\mathrm{left}, \mathrm{right}\}$ **do**
        if $\mathrm{ID}_a[i] = 0$ **then** $\mathrm{dir} \leftarrow k$
        else $\mathrm{dir} \leftarrow$ direction opposite to k
        SingleRound; $D_k \leftarrow D_k \cup \{\mathrm{coll}()\}$; ReversedRound
$\mathrm{dist}_{\mathrm{right}} \leftarrow 2\min(D_{\mathrm{right}})$ ▷ $\mathrm{dist}_{\mathrm{right}}$ is the distance to $\mathrm{Right}(a)$
$\mathrm{dist}_{\mathrm{left}} \leftarrow 2\min(D_{\mathrm{left}})$ ▷ $\mathrm{dist}_{\mathrm{left}}$ is the distance to $\mathrm{Left}(a)$
if $2\mathrm{start}_{\mathrm{left}} = \mathrm{dist}_{\mathrm{left}}$ **then**
    $\mathrm{agree}_{\mathrm{left}} \leftarrow \mathrm{false}$
else ▷ $\mathrm{agree}_{\mathrm{left}} = \mathrm{true}$ iff a and $\mathrm{Left}(a)$ share sense of dir.
    $\mathrm{agree}_{\mathrm{left}} \leftarrow \mathrm{true}$
if $2\mathrm{start}_{\mathrm{right}} = \mathrm{dist}_{\mathrm{right}}$ **then**
    $\mathrm{agree}_{\mathrm{right}} \leftarrow \mathrm{false}$
else ▷ $\mathrm{agree}_{\mathrm{right}} = \mathrm{true}$ iff a and $\mathrm{Right}(a)$ share sense of dir.
    $\mathrm{agree}_{\mathrm{left}} \leftarrow \mathrm{true}$

Proposition 35. *Algorithm 6 gives solution to neighbors discovery in $O(\log N)$ rounds.*

Proof. Consider a and $a' = \mathrm{Right}(a)$. First, we show that, in some round, they start moving towards each other (which gives the distance to collision equal to half of their distance, the smallest possible). If they have *opposite* sense of direction, then this happens in line 2 of the algorithm. Otherwise (i.e., they have the same sense of direction), they start moving towards each other for such i, k that $\mathrm{ID}_a[i] \neq \mathrm{ID}_{a'}[i]$ and either:

- $\mathrm{ID}_a[i] = 0$, $\mathrm{ID}_{a'}[i] = 1$ and $k = \text{right}$, or
- $\mathrm{ID}_a[i] = 1$, $\mathrm{ID}_{a'}[i] = 0$ and $k = \text{left}$.

Finally observe that a and $\mathrm{Right}(a)$ have opposite senses of directions if and only if the smallest (right) distance to collision for a appears when all agents start a round moving to the right (according to their senses of direction). An analogous observation holds for relative senses of direction of a and $\mathrm{Left}(a)$. $\square$

Proposition 36. *If each agent knows:*

- *locations of its neighbors (relative to its initial location); AND*
- *sense of direction of its neighbors (with respect to its own sense of direction);*

then each agent can transmit one bit of information to its neighbors in time $O(1)$.

Proof. Assume that each agent a is going to transmit bit z_a. Consider a *phase* which consists of four rounds:

- Round $1, 2$: if $z_a = 1$ then $\mathrm{dir}_a \leftarrow \text{right}$ else $\mathrm{dir}_a \leftarrow \text{left}$;
  SingleRound, ReversedRound
- Round $3, 4$: if $z_a = 0$ then $\mathrm{dir}_a \leftarrow \text{right}$ else $\mathrm{dir}_a \leftarrow \text{left}$;
  SingleRound, ReversedRound

Consider an agent a and its right neighbor $b = \mathrm{right}(a)$. We show that a is able to determine z_b (the bit transmitted by b) based on results of Rounds 1 and 3 (i.e., on moments of first collisions). We say that a round is *clear* if a and b collide in the middle point between their original locations, which means that they start a round moving towards each other. We have two cases:
Case 1: the sense of direction of a and b agree:

(a) $z_a = 1$: if Round 1 is clear, then $z_b = 0$, otherwise $z_b = 1$.

(b) $z_a = 0$: if Round 3 is clear, then $z_b = 1$, otherwise $z_b = 0$.

Case 2: the sense of direction of a and b do *not* agree:

(a) $z_a = 1$: if Round 1 is clear, then $z_b = 1$, otherwise $z_b = 0$.

(b) $z_a = 0$: if Round 3 is clear, then $z_b = 0$, otherwise $z_b = 1$.

Communication with $\mathrm{left}(a)$ can be performed in an analogous way.

$\square$

Since agents can learn location of their neighbors and their sense(s) of direction in $O(\log N)$ rounds (see Proposition 35), Proposition 36 leads to the following corollary.

Corollary 37. *There exists a possibility to exchange $p \geq 1$ bits of information between each two neighbors in the perceptive model in time $O(p)$, after a $O(\log N)$ preprocessing.*

The above corollary gives opportunity to simulate any distributed algorithm on a ring in message passing model (i.e., when each pair of neighbors can exchange a message in one round of computation). However, the time efficiency of such simulations is limited by the fact that only one bit of information is exchanged between neighbors in $O(1)$ rounds.

Let *information dissemination task* with parameters d and p be to disseminate a message m_a with p bits by each agent a to all agents in ring distance $\leq d$ from a.

Corollary 38. *Information dissemination task in which agents are supposed to transmit messages of length p on the ring distance d can be accomplished in time $O(p \cdot d)$, after a $O(\log N)$ preprocessing.*

Proof. At the beginning, in $O(\log N)$ rounds, we perform preprocessing which collects information sufficient for exchange of messages between neighbors (see Corollary 37). In particular, agents learn whether their neighbors have the same or opposite (to them) sense of direction (see Proposition 35). Then the algorithm works in d phases. In the first phase, each agent sends its own message to both its neighbors in $O(p)$ rounds (Corollary 37)). In the ith phase for $1 < i \leq d$, each agent a sends two p-bit messages:

- Firstly, the p-bit message received from the left neighbor (according to the sense of direction of a) Left(a) in phase $i-1$ is sent by a to the right neighbor Right(a).
- Then, the p-bit message received from the right neighbor (according to the sense of direction of a) Right(a) in phase $i-1$ is sent by a to the left neighbor Left(a).

Overall, all p-bit messages are transmitted on the left and right ring distance d from their sources in $O(pd)$ rounds. □

Assume that $A' \subset A$ is a set of *marked* agents such that each agent knows whether it is marked or not and the ring distance between any different $a, a' \in A'$ is at least d. Moreover, each $a \in A'$ has a message M_a of size $\leq p$. The *sparsed information dissemination task* with parameters A', d and p is to deliver the message of each $a \in A'$ to all agents in the ring distance $\leq d$ from a. For an agent in A', we denote this task by Diss(M_a, d). Using the procedure exchanging a bit of information between each pair of neighbors in time $O(1)$, we obtain the following result.

Corollary 39. *Sparsed information dissemination task in which agents in distances $\geq d$ are supposed to transmit messages of length p on the ring distance d can be accomplished in time $O(p+d)$, after a $O(\log N)$ preprocessing.*

Proof. At the beginning, in $O(\log N)$ rounds, we perform preprocessing which collects information sufficient for exchange of messages between neighbors (see Corollary 37). In particular, agents learn whether their neighbors have the same or opposite (to them) sense of direction (see Proposition 35).

Then, in order to accomplish the task in $O(p+d)$ rounds, we use pipelining. Each agent a keeps two queues: the queue Q_L^a containing the sequence of bits which should be transmitted to the left neighbor Left(a) and the queue Q_R^a containing the sequence of bits which should be transmitted to the right neighbor Right(a). At the beginning, both queues of marked agents contain the sequence of bits of their original message, while the queues of other agents are empty. The algorithm works in $p+d$ phases. In each phase, each agent a sends:

- the first bit from Q_L^a to its left neighbor Left(a),
- the first bit from Q_R^a to its right neighbor Right(a).

Moreover, each sent bit is removed from the appropriate queue and each bit received by a from the left or right neighbor is added to Q_R^a or Q_L^a, respectively. Given the constraints of sparsed information dissemination task, one can verify that each original message will be transmitted on the ring distance d in $p+d$ phases (since each agent can receive at most one message of length p from its left neighbor and at most one message of length p from its right neighbor).

However, in the above described solution, we have to tackle the fact that an agent has no direct way to convey a message of the type "I have nothing to transmit" (or "My queue is empty"). One can solve this issue by a simple encoding, e.g., 00/11 encodes 0/1, while 01 encodes "no bit to transmit". □

6.2. Nontrivial Move

As we know, the nontrivial move problem is intuitively to break balance between the number of agents moving clockwise and anticlockwise. In our solution we use (N,k)-selective families from [25].

Definition 40. *Let $n < N$. A family $\mathcal{F}$ of subsets of $[N]$ is (N,n)-selective if, for every non empty subset Z of $[N]$ such that $|Z| \leq n$, there is a set F in $\mathcal{F}$ such that $|Z \cap F| = 1$.*

Clementi et al. [25] showed that for any $N > 2$ and $n \leq N$, there exists an (N,n)-selective family of size $O(n\log(N/n))$.

Let a *local leader* for some fixed number d be an agent a with the largest ID among agents in the ring distance d from a.

One can build a solution of the nontrivial move problem by establishing local leaders for exponentially growing distances $d = 2^k$ and trying to execute $(N, 2^k)$-selective family on those leaders. As the number of local leaders is

$\leq n/2^k$, it becomes smaller than 2^k for $k > \frac{1}{2}\log n$ and gives a nontrivial move after $O(2^{\frac{1}{2}\log n}\log N) = O(\sqrt{n}\log N)$ rounds.

This solution can be further modified such that a nontrivial move is obtained in $O(\log N \cdot (\log\log N) + \sqrt{n\log N})$ rounds. The idea is to postpone the use of the selective families until the cost of using them might be similar to the cost of decreasing the number of local leaders. More precisely, one starts by electing local leaders until the consecutive local leaders are in distances $\geq \log N$ from each other (see the former for-loop in Algorithm 7). Using sparsed information dissemination as before, this part of the algorithm can be accomplished in $O(\log N \cdot (\log\log N))$ rounds. Then, at each phase $k = 1, 2, \ldots$, local leaders are chosen $2^k \log N$ apart, and the $(N, 2^k)$-selective family is applied (see the latter for-loop in Algorithm 7). Thus phase k costs $2^k \log N$ rounds. The nontrivial move problem is solved at the latest when $n/(2^k \log N) < 2^k$, i.e, for some $k \leq \frac{1}{2}\log\left(\frac{n}{\log N}\right) + O(1)$. This in turn gives the nontrivial move in

$$O(\log N \cdot (\log\log N) + 2^{\frac{1}{2}\log\left(\frac{n}{\log N}\right)}\log N) = O(\log N \cdot (\log\log N) + \sqrt{n\log N})$$

rounds.

Algorithm 7 NMovePercQuick(a)

$\mathrm{dir}_a \leftarrow$ right; set the status of a as a local leader;
SingleRound
If the current directions give a nontrivial move: return
Establish 1-bit communication ▷ Cor. 37
for $k = 0, 1, \ldots, \lceil \log\log N \rceil$ **do**
  Sparsed dissemination of ID_a of local leaders on distance 2^k ▷ Cor. 39
  if $\mathrm{ID}_a = \max(N_a(2^k))$ **then**
    set the status of a as the *local leader*
  else
    set the status of a as *not leader*
  NMoveWithLeader ▷ Lemma 10
  if a nontrivial move appears during NMoveWithLeader **then**
    return
for $k = 1, 2, \ldots, \lceil \log N \rceil$ **do**
  Sparsed dissemination of ID_a of local leaders on distance $2^k\lceil \log N \rceil$
  if $\mathrm{ID}_a = \max(N_a(2^k))$ **then**
    set the status of a as the *local leader*
  else
    set the status of a as *not leader*
  Execute a $(N, 2^k)$-selective family $\mathcal{F}$ on the set of local leaders
  if a nontrivial move appears during execution of $\mathcal{F}$ **then** return

If $\log\log N > \log n$, the unique leader is elected already in the former for-loop of Algorithm 7. If the leader is elected, the nontrivial move problem can be solved in $O(1)$ rounds by Lemma 10. Let NMoveWithLeader be the algorithm which solves the nontrivial move problem in $O(1)$ rounds. If $\log\log N > \log n$, a nontrivial move appears in an execution of NMoveWithLeader in line 11 of Algorithm 7 for some $k \leq \lceil \log n \rceil$. Therefore, Algorithm 7 gives the nontrivial move in

$$O(\log N \cdot \min(\log\log N, \log n) + \sqrt{n \log N})$$

rounds. The following lemma summarizes the above described ideas in a more formal way.

Lemma 41. *Algorithm 7 solves the nontrivial move problem in*

$$O\left((\log N)(\log n) + \sqrt{n \log N}\right)$$

rounds in the perceptive model.

Proof. First, we prove that the algorithm actually solves the nontrivial move problem. If $\log n \leq \log\log N$, the leader of the whole network is chosen in the former for-loop and a nontrivial move appears in line 11, by Lemma 10. Let us fix some "objective correct" sense of direction. Let A_C, A_I be the subsets of agents with correct and incorrect sense of direction, respectively. If the algorithm does not finish its execution in line 3, the rotation index is in $\{0, n/2\}$ for a round with all dir_a equal to right. If exactly one agent changes its initial direction in a round, the rotation index will increase by 2 or -2. That is, it will not be in $\{0, n/2\}$ since $n > 4$, and a nontrivial move will be obtained. Let L be the set of local leaders in the kth iteration of the latter for-loop. As discussed before, the size of L is at most $n/(2^k \log N)$. If $n/(2^k \log N) \leq 2^k$, the selective family in line 20 will select exactly one element from the set of local leaders in some round. This follows from the fact that the set of local leaders is not larger than 2^k in such case and therefore the $(N, 2^k)$-selective family is sufficient to select exactly one element from L.

Regarding time complexity, the former for-loop requires

$$O(\log N \cdot \min(\log n, \log\log N))$$

rounds. In the latter for-loop, each phase $k = 1, 2, \ldots$ costs $O(2^k \log N)$ rounds. The nontrivial move is obtained at the latest when $n/(2^k \log N) < 2^k$, i.e, for $k \leq \frac{1}{2}\log\left(\frac{n}{\log N}\right) + O(1)$. Thus, the number of rounds of the latter for-loop is $O(2^{\frac{1}{2}\log\left(\frac{n}{\log N}\right)+O(1)} \log N) = O(\sqrt{n \log N})$. Summarizing, the algorithm works in

$$O\left(\log N \cdot \min(\log n, \log\log N) + \sqrt{n \log N}\right)$$

rounds. However, if $\log\log N < \frac{1}{4}\log n$, then

$$\log N \cdot (\log\log N) = O(n^{1/4} \log n) = O(\sqrt{n \log N})$$

and therefore the asymptotic complexity of the algorithm can be expressed as $O((\log N) \cdot (\log n) + \sqrt{n \log N})$. $\square$

6.3. Location Discovery in the perceptive model

In this section we design an efficient solution for the location discovery in the perceptive model. Using results from the previous section and Theorem 7, we can assume that the leader is elected and the common sense of direction is established in $O\left(\sqrt{n \log N} + (\log N)(\log n)\right)$ rounds. Throughout this section, we number the subscripts of agents such that a_1 denotes the leader and a_i is the $(i-1)$th agent on the ring in the clockwise direction from the leader.

We solve the location discovery problem in two stages. First, each agent determines its right ring distance to the leader. In order to achieve this goal in the standard message passing model on a ring, linear time is necessary. In order to perform this task faster, we use arithmetic relationships between distances to collisions (coll()) and distances traversed in consecutive rounds (dist()). For appropriately designed protocol, an agent in ring distance $\leq d^2$ from the leader will be able to learn its ring distance in $O(d \log N)$ rounds. Then, using the knowledge about ring distances of agents to the leader, the location discovery will be finally solved in the following way. Let $x_1, \ldots, x_n$ be the original distances between agents. (That is, x_i is the distance between a_i and a_{i+1} for $i < n$ and x_n is the distance between a_n and a_1.) Here, we plan movements of agents in such a way that, for each agent and each round, the distance to collision in the round and the distance traversed in the round gives a linear equation over $x_1, \ldots, x_n$ which is linearly independent from equations derived before. In this way each round provides two new equations and $n/2$ rounds are sufficient to determine the actual values of $x_1, \ldots, x_n$, since they give a system of n independent linear equations over n variables.

6.3.1. Ring distances

Now, we design the RingDist protocol in which each agent learns its right ring distance from the leader. Throughout this section, ring distance denotes the right ring distance from the leader. Moreover, we assume the subscripts are such that a_1 is the leader and a_i is the $(i-1)$th agent in the clockwise direction from the leader.

Let Shift(l) for $l \in \mathbb{N}$ be a round in which $\text{dir}_{a_i} = \text{right}$ for each $i \in [l]$ and $\text{dir}_{a_i} = \text{left}$ for $i > l$. Moreover, Shift$(-l)$ is a round with directions of agents opposite to their direction in Shift(l). Observe that the rotation index of Shift(l) is equal to $(l - (n - l)) \bmod n \equiv 2l \bmod n$.

RingDist works under assumption that (exactly) one distinguished agent has the status leader (it is denoted a_1). Each agent but the leader starts an execution of a protocol with unspecified ring distance. The idea of Algorithm 8 is that the agents gradually learn their ring distances in the following way:

- The agents in ring distance ≤ 4 learn their distances in step 1 (the same applies to the agents $\geq n - 4$, although they learn only their relative values, without knowing n). This task can be performed by an execution of the procedure Diss, which efficiently implements sparsed information dissemination task (see Corollary 39).

- In the ith iteration of the for-loop, the agents $a_{k+k}, a_{k+2k}, \ldots, a_{k+k^2}$ for $k = 2^i$ learn their ring distances in the following way (see Fig. 3). For each

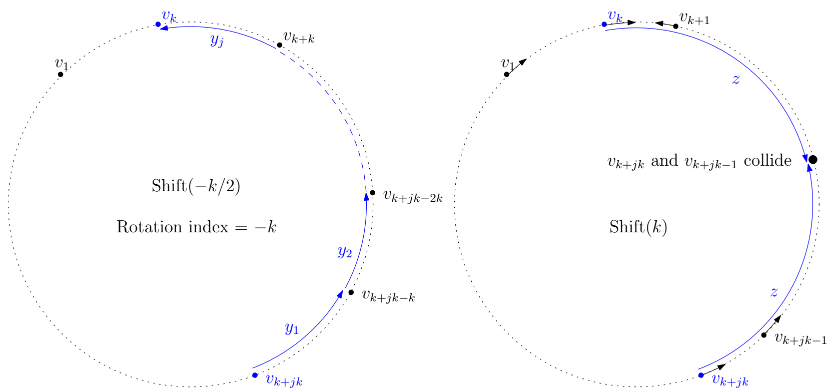

Figure 3: An illustration for Algorithm 8. The agent is in the right ring distance $k + jk$ from the leader for some $1 \le j \le k$ iff $2z$ is equal to the sum $\sum_{i=1}^{j} y_i$.

  $l > k$, the value of coll() in Shift(k) is equal to $z = (x_k + \cdots + x_{l-1})/2$ (see Prop. 4 for $b_0 = a_l$, dir = left and thus $b_i = a_{(l-i) \bmod n}$). On the other hand, if one applies Shift$(-k/2)$ several times, the values of dist() in the jth execution of Shift$(-k/2)$ is equal to $y_j = x_{l-jk} + \cdots + x_{l-(j-1)k+1}$, since the rotation index of Shift$(-k/2)$ is equal to $-k$. Using these relationships, we see that there exists j such that $2z = y_1 + \cdots + y_j$ iff $l = k + k \cdot j$. This observation is exploited in RingDist in order to determine ring distances of $a_1, \ldots a_{k+k^2}$ in the ith iteration of the main for-loop for $k = 2^i$.

- The remaining agents a_j for $j \le k + k^2$ learn their distances during execution of line 10, as each agent knowing its ring distance propagates it in the distance k.

Then, it remains to guarantee that the for-loop is finished when all agents know their ring distances and $2^i = O(\sqrt{n})$. To this aim, we execute CheckCompleteness. Note that the agent a_n knows that it is the last one already at the beginning (without knowing n), as it is the left neighbour of the leader. CheckCompleteness is a round in which all agents different from a_n move left, while a_n moves right iff it already knows its own right ring distance (which in turn implies that every other agent knows its ring distance as well). Thus, the rotation index of this round is not zero iff each agent knows its ring distance. In the following, we show more formally that the above described idea works. First, we make an observation following from the definition of Shift (the rotation index of Shift(l) is $2l$) and Proposition 4.

Proposition 42. *Let $k = 2^i$ for $i \le \log N$. Assume that agents $a_1, \ldots, a_k$ know their right ring distances from the leader before the ith iteration of the for-loop (and other agents know that they do not belong to $\{a_1, \ldots, a_k\}$). Then,*

Algorithm 8 RingDist(a)

1: **if** $a = a_1$: Diss("leader",4) ▷ The leader a_1 broadcasts its message on ring distance 4
2: **for** $i = 1, 2, \ldots, \lceil \log N \rceil$ **do**
3: $\quad k \leftarrow 2^i$
4: $\quad$ For $j = 1, \ldots, k$: Shift$(-k/2)$; $y_j \leftarrow$ dist()
5: $\quad$ Repeat k times: Shift$(k/2)$ ▷ Reverse the result of $k\times$ Shift$(-k/2)$
6: $\quad$ Shift(k); $z \leftarrow$ coll(); Shift$(-k)$
7: $\quad$ **if** $2z = y_1 + \cdots + y_j$ for some j and $a \notin \{a_1, \ldots, a_k\}$:
8: $\quad\quad$ Set the ring distance of a to $k + jk$; mark a ▷ i.e., $a \leftarrow a_{k+jk}$
9: $\quad$ **if** $a = a_{k+jk}$ for $j \leq k$ and a marked **then**
10: $\quad\quad$ Diss$(k + jk, k)$ ▷ Marked agents broadcast their ring dist. on distance k
11: $\quad$ If CheckCompleteness: return ▷ See description of the alg. for details

for $l > k$ the values of z, $y_1, \ldots, y_k$ recorded by the agent a_l satisfy the following conditions in the iteration i of the for-loop:

- $y_j = x_{l-kj} + x_{l-kj+1} + \cdots + x_{l-k(j-1)-1}$;
- $z = (x_k + \cdots + x_{l-1})/2$.

The following corollary is an immediate consequence of Proposition 42.

Corollary 43. *The condition $2z = y_1 + \cdots + y_j$ is satisfied for an agent $a \notin \{a_1, \ldots, a_k\}$ iff a is in the right ring distance $k + jk$ from the leader (i.e., $a = a_{k+jk}$).*

Lemma 44. *Assume that the leader is elected and all agents share common sense of direction. Then, each agent a determines its ring distance during the algorithm RingDist and the algorithm lasts $O(\sqrt{n} + (\log n) \cdot (\log N))$ rounds.*

Proof. Before the for-loop, the agents in ring distance ≤ 4 are aware of their ring distance, while the agent a_n knows that it is "the last one". We show by induction that, after the ith iteration of the main for-loop, the agents $a_1, \ldots, a_{k^2}$ know their right ring distances from the leader for $k = 2^i$. As the base step is obvious, assume inductively that before the ith iteration, the agents $a_1, \ldots, a_{(2^{i-1})^2}$ know their right ring distances from the leader for $i > 1$. Thus, in particular $a_1, \ldots, a_k$ know their right ring distances from the leader for $k = 2^i$, since $2^i \leq (2^{i-1})^2$ for $i > 1$. This assures that agents are able to perform all steps in the i iteration of the main for-loop. Then, Proposition 42 and Corollary 43 imply that each agent a_{k+jk} for $j \in [k]$ and $k = 2^i$ becomes aware of its ring distance before dissemination of distances in line 10. In line 10 agents $a_{k+k}, a_{k+2k}, \ldots, a_{k+k^2}$ broadcast information about their ring distances to agents in their ring distance $\leq k$. Thus, the remaining agents a_l for $l \leq k + k^2 + k$ learn their ring distances from the agents $a_{k+k}, a_{k+2k}, \ldots, a_{k+k^2}$. Finally, for the smallest i such that $2^i + (2^i)^2 + 2^i \geq n$, the for-loop is finished and all agents know their ring distances.

In order to execute the ith iteration, $O(2^i + \log N)$ rounds are sufficient (by Corollary 39, dissemination of distances by marked agents can be done in $O(2^i + \log N)$ rounds). Time complexity of the whole procedure is

$$O\left(\sum_{i=1}^{\frac{1}{2}\log n} (2^i + \log N)\right) = O(\sqrt{n} + (\log n)(\log N)).$$

$\square$

6.3.2. Location discovery

In this section we describe a solution for the location discovery problem based on protocols presented before. Recall that, given the common sense of direction and the leader, one can obtain a round with rotation index 2 by assigning the direction left to all agents but the leader. If n is odd, this gives a solution to the location discovery problem in n rounds. The goal of this section is to get advantage of information provided by positions of the first collision in a round, in order to decrease time from n to $n/2$ and manage the case that n is even.

Here, we assume that the leader a_1 is elected, the agent(s) in the right ring distance i from the leader is a_{i+1} and each agent a_i knows i (see Lemma 44). Moreover, we assume that n is even. However, quite a similar solution can be built for the case that n is odd. Indeed, in such a case it is sufficient to change Alg. 9 a little, in order to assure that information collected by agents during an execution of the algorithm allows for determinining all distances (see the proof of Proposition 45).

Let x_i denote the distance on the ring between the agent a_i and the agent a_{i+1} (or a_1 if $i = n$). (Note that this is the geometric distance on the ring, not the ring distance!)

Let Convolution(j) be a round in which the agents' directions are as follows (see Figure 4):

> $\mathrm{dir}_{a_{2i-1}} = \mathrm{right}$, $\mathrm{dir}_{a_{2i}} = \mathrm{left}$ for each $i \in [n/2]$, with an exception: $\mathrm{dir}_{a_{2j}} = \mathrm{right}$.

Let Pivot(j) be a round in which the agents' directions are as follows: $\mathrm{dir}_{a_{j+1}} = \mathrm{dir}_{a_{j+2}} = \cdots = \mathrm{dir}_{a_{j+n/2}} = \mathrm{left}$ and $\mathrm{dir}_{a_j} = \mathrm{dir}_{a_{j-1}} = \cdots = \mathrm{dir}_{a_{j-n/2+1}} = \mathrm{right}$. (Here, the subscript indices are calculated modulo n and a_0 is identified with a_n.) Observe that the rotation index of Convolution(i) is equal to 2 and the rotation index of Pivot(i) is equal to 0 for each i. In the following,

Algorithm 9 Distances(a)

```
for i = 1, 2, ..., n/2 do
    Convolution(n/2 − i + 1)
Pivot(n); Pivot(n − 1); Pivot(n − 2);
```

we show that information collected during an execution of Algorithm 9 by an agent can determine original positions of all other agents.

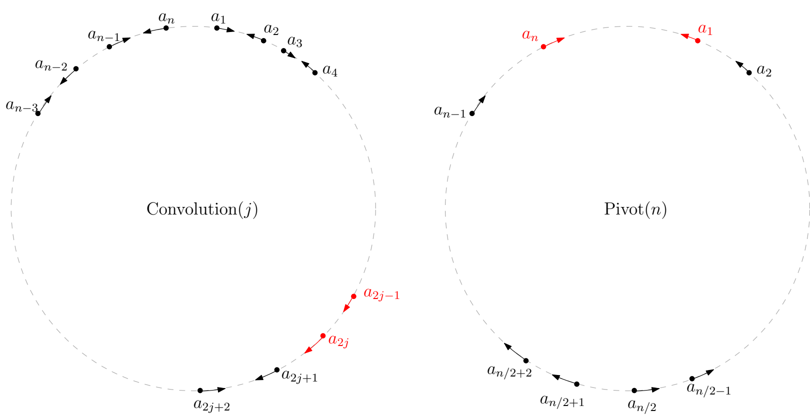

Figure 4: An illustration of Convolution(j) and Pivot(n). In Convolution(j), the red agents (a_{2j} and a_{2j-1}) are the only one which do not collide with a neighbor in the middle of the distance between them. In Pivot(j), the red agents (a_1 and a_n for Pivot(n)) are the only one which collide with a neighbor in the middle of distance between them.

Proposition 45. *After the for-loop of Algorithm 9, the following conditions hold for each $i \in [n/2]$:*

(a) the agent a_{2i-1} can determine the values of $x_1, x_2, \ldots, x_{n-2}$ and $x_{n-1}+x_n$.

(b) the agent a_{2i} can determine the values of $x_1, x_2, \ldots, x_{n-3}$, x_n, and $x_{n-2}+x_{n-1}$

Proof. First, observe that the agent a_k stays at the original position of $a_{k+2(i-1)}$ before the ith execution of line 2, since the rotation index of Convolution is equal to 2. Therefore, the only agent with "swapped" direction stays in each round of the for loop at the original position of a_j for $j = 2\left(\frac{n}{2} - i + 1\right) + 2(i-1) = n$. As a result, the agent a_k learns in the $(i+1)$st execution of line 2 exactly the same values as a_{k+2i} in the first execution of line 2. Consider this first round. As a_{2j-1} and a_{2j} start the round moving towards each other for each $j \in [n/2-1]$ (i.e., $j < n/2$), they collide after traversing the distance $x_{2j-1}/2$ and therefore learn coll$() = x_{2j-1}/2$. On the other hand, the rotation index of Convolution with any parameter is 2 and thus each agent a_k learns dist$() = x_k + x_{k+1}$. All these observations lead to the following conclusion:

- Each agent with odd right ring distance from the leader learns the values of $x_{2j-1}+x_{2j}$ for each $j \in [n/2]$ (registered as dist() in consecutive rounds) and x_{2k-1} (corresponding to coll()) for each $k \in [1, n/2-1]$.
- Each even agent learns the values of $x_{2(j-1)}+x_{2j-1}$ for each $j \in [n/2]$ and x_{2k-1} for each $k \in [1, n/2-1]$.

Given the above, each agent can deduce the values enumerated in the proposition. E.g., each agent with odd right ring distance from the leader solves the system of equations $x_{2j-1} + x_{2j} = d_j$ for $j \leq n/2$ and $x_{2j-1} = c_j$ for $j < n/2$, where d_j and c_j are the values observed as dist() and coll() in various rounds. □

Now, observe that

- The agent a_{2i-1} has the first collision during Pivot(n) in distance $x_n/2 + (x_1 + \cdots + x_{2i-2})/2$ for each $i \in [\lfloor n/4 \rfloor]$ As a_{2i-1} knows $x_1, \ldots, x_{2i-2}$ and $x_{n-1} + x_n$ by Proposition 45(a), it can determine $x_n/2$ from Pivot(n) and therefore also x_{n-1}.
- The agent a_{2i-1} has the first collision during Pivot($n-1$) in distance $x_{n-1}/2 + (x_{2i-1} + \cdots + x_{n-2})/2$ for each $i \in [n/4+1, n/2-1]$. As it knows $x_{2i-1}, \ldots, x_{n-2}$ and $x_{n-1} + x_n$, it can determine x_{n-1} from Pivot($n-1$) and then x_n as well.

  A similar reasoning works for a_{n-1} as well.

By combining the above with Prop. 45(a), one can conclude that each agent a_i with odd right ring distance from the leader i knows original positions of all agents. A similar argument applies for even agents and executions of Pivot($n-1$) and Pivot($n-2$), since Prop. 45(b) can be seen as Prop. 45(a) "shifted" by -1.

Thus, we obtain the following conclusion.

Lemma 46. *The protocol Distances (Alg. 9) solves the location discovery problem in at most $\frac{n}{2} + 3$ rounds, provided the leader (a_1) is elected, agents share common sense of direction and each agent knows its ring distance.*

By combining the subroutines described before (nontrivial move and leader election algorithms – Lemma 41 and Theorem 7; RingDist – Lemma 44 Distances – Lemma 46), we get the following result.

Theorem 47. *The location discovery problem can be solved in the perceptive model in $n/2 + O\left(\sqrt{n \log N} + (\log N)(\log n)\right)$ rounds.*

7. On distinguishing parity of n and small values of n

Most of solutions of considered problems presented in the paper differ significantly for odd and even n. Moreover, they were designed under assumption that $n > 4$ or $n = 3$. In this section we show that it is possible to determine whether $n > 4$ and distinguish odd n from even n efficiently.

7.1. Small values of n

Now, we consider the problem of determining whether $n > 4$ or not. (Recall that most of our solutions rely on the assumption that $n > 4$ or $n = 3$.)

Proposition 48. *Assume that a ring network works under assumptions of the basic model. Then,*

1. *if $n = 3$, the location discovery and coordination problems can be solved in $O(\log N)$ rounds;*

2. *in order to distinguish the cases* $n \in \{1,2\}$ *and* $2 < n = \Theta(m)$ *for* $4 < m \leq N$, $\Theta(m \log(N/m)/\log m)$ *rounds are necessary;*

3. *the cases* $n = 1$ *and* $n = 2$ *are indistinguishable and therefore coordination problems and the location discovery are not solvable for* $n \in \{1,2\}$*;*

4. *the cases* $n \in \{1,2\}$ *and* $n = 4$ *can be distinguished in* $O(\log N)$ *rounds;*

5. *if* $n = 4$*, the location discovery is not solvable.*

Proof. The case $n = 3$ has been considered in previous sections, the standard solutions apply here.

If $n \in \{1,2\}$ then the rotation index of each round is equal to 0 and therefore one cannot distinguish between $n = 1$ and $n = 2$. Note that it is not possible to obtain even a weak nontrivial move in this case (i.e., rotation index $n/2$). Due to the lower bound from Corollary 28, and the upper bound from Theorem 29, $\Theta(m \log(N/m)/\log m)$ rounds are needed to obtain a (weak) nontrivial move if $2 < n = \Theta(m)$. Moreover, without a (weak) nontrivial move, agents do not have any opportunity to learn anything about the network or its size: the lack of nontrivial move in r rounds might be caused either by the fact that the number of agents is in $\{1,2\}$ or by the fact that $r = o(m \log(N/m)/\log m)$, where m is the actual number of agents in the network.

Now, assume that $n \in \{1,2,4\}$. If a round with $\mathrm{dir}_a = \mathrm{right}$ for each agent a gives the rotation index not equal to 0, we know that $n \notin \{1,2\}$ and thus $n = 4$. Otherwise, either all agents have the same sense of direction or halve of them have "correct" sense of direction and the other halve have the opposite sense of direction. For this case, we execute $(N,4)$-selective family of [25] which guarantees that exactly one agent is selected in $O(\log N)$ rounds. By choosing direction left of the selected agents (and keeping right for the remaining ones), either:

- a round with non-zero rotation index appears for $n = 4$ (in a round selecting exactly one agent);

- a round with non-zero rotation index does not appear, which means that $n \in \{1,2\}$.

Finally, the location discovery is not solvable for $n = 4$ as 4 is even. □

Observe that the above proposition does not give an answer to the question how efficiently the cases $n = 4$ and $n = \Theta(m)$ for $4 < m < N$ can be distinguished. As we noticed in the proof of Proposition 48, a weak nontrivial move can be obtained for $n = 4$ in $O(\log N)$ rounds. The question is whether agents can determine that $n = 4$ in another way than by noticing that a protocol for nontrivial move does not give a positive result even if the number of rounds allows for running the nontrivial move protocol for $n = N$, i.e., in $O(N/\log N)$ rounds. (That is, whether weak nontrivial moves can give sufficient information to distinguish $n = 4$ and $n = \Theta(m)$.)

Proposition 49. *Assume that a ring network works under assumption of the lazy model. Then,*

1. *if $n \in \{3, 4\}$, the location discovery and coordination problems can be solved in $O(\log N)$ rounds;*
2. *in order to distinguish the cases $n = 1$ and $n = \Theta(m)$ for $4 < m \leq N$, $\Theta(m \log(N/m)/ \log m)$ rounds are needed;*
3. *the cases $n = 1$ and $n = 2$ can be distinguished in $O(\log N)$ rounds.*

Proof. The case $n = 3$ has been considered in previous sections, the standard solutions apply here.

Similarly to the basic model, the rotation index of each round is equal to 0 for $n = 1$ and therefore $\Theta(m \log(N/m)/ \log m)$ rounds are needed to distinguish $n = 1$ and $n = \Theta(m)$.

Let us execute $(N, 4)$-selective family of [25] in such a way that all selected agents start a round moving right and the remaining agents start the round idle. Then, $(N, 4)$-selective family guarantees a round in which exactly one agent is selected in $O(\log N)$ rounds. The rotation index of such round is 1 (or $-1 \bmod n$) for $n > 1$ which gives a weak nontrivial move for $n = 2$ and a nontrivial move for $n = 4$, excluding the case $n = 1$.

Given a nontrivial move ($n = 4$), we can solve the direction agreement problem and the leader election problem in the same way as for $n > 4$ in $O(\log N)$ rounds. With common sense of direction and the leader, the location discovery is solved in 4 rounds, as described in Lemma 17. □

Analogously to the case $n = 4$ in the basic model, we do not know whether weak nontrivial move allows to distinguish the case $n = 2$ and $n = \Theta(m)$ for $4 < m < N$ faster than $\Theta(m \log(N/m)/ \log m)$.

Proposition 50. *Assume that a ring network works under assumption of the perceptive model. It is possible to determine in $O(\log N)$ rounds whether $n \leq 4$ and solve the location discovery problem in $O(\log N)$ rounds.*

Proof. Here, we can use neighbor discovery protocol (Algorithm 6) and the possibility of exchanging messages between neighbors to solve the task. First, agents learn IDs and order of all agents in their ring distance 4 in $O(\log N)$ rounds. In this way, they determine whether $n \leq 4$. And, if $n \leq 4$, they determine the exact value of n. Consider $n = 4$ The agents can determine geometric distances (on the circle) to their neighbors by distances to collisions in appropriate rounds. Then, after a round with rotation index 2, they can similarly determine two remaining distances. We leave the details and the cases $n = 1, 2, 3$ as an exercise. □

7.2. Parity checking

As complexity (and feasibility) of some problems differ substantially depending on the fact whether n is odd or even, a natural question arises about an efficient method of determining parity of n.

First, one can try to apply a nontrivial move protocol for odd n, working in $O(\log N)$ rounds (Proposition 20). If no nontrivial move is obtained during execution of this protocol, we know that n is even. If nontrivial move is solved, the common sense of direction can be agreed in $O(1)$ rounds (Lemma 8).

So, from now on, we assume that agents share the common sense of direction.

Before we describe our solution for the problem of parity checking, we slightly redefine the notion of a rotation index of a set. That is, the *modified rotation index* of a set B, $\mathrm{MRI}(B)$, satisfies the conditions: $\mathrm{MRI}(B) = \mathrm{RI}(B)$ if $|B| \notin \{n/2, n\}$ and $\mathrm{MRI}(B) = n$ for $|B| \in \{n/2, n\}$. Thus, $\mathrm{MRI}(B) = |B| - (n - |B|) \equiv 2|B| \pmod n$ if $|B| \notin \{n/2, n\}$. Recall that an agent can relatively easily determine in $O(1)$ rounds whether the rotation index of a set is greater than, equal or smaller than $n/2$, even when it does not know the actual value of n or its estimation (Lemma 2). Thus, in order to distinguish whether the modified rotation index of a set B is larger, smaller or equal to $n/2$, we only need to deal with $|B| \in \{0, n/2, n\}$ which appears iff $\mathrm{RI}(B) = 0$. In order to distinguish the cases $B = \emptyset$ from $|B| \in \{n/2, n\}$, the emptiness testing algorithm can be applied (Lemma 12). For completeness, this idea is implemented in Algorithm 10 (TestMaj).

Now, we describe the idea of our algorithm determining parity of n implemented as Algorithm 11 and using as a subroutine the procedure TestMaj described above. The goal of the algorithm is to select a subset $L \subset A$ of agents such that $\mathrm{MRI}(L)$ is as close to $n/2$ as possible. If we find such L that $\mathrm{MRI}(L) = n/2$ then $n/2$ is natural and n is even. In the case that the algorithm does not find such L, it will still determine the set of agents L and an agent $a \notin L$ (i.e., at the end, each agent b knows whether $b \in L$ and whether $b = a$) such that $\mathrm{MRI}(L) < n/2$ and $\mathrm{MRI}(L \cup \{a\}) > n/2$. As we show later, one can easily determine parity of n, given L and a as above. The algorithm implements a kind of binary search procedure. It starts with $L = \emptyset$ and $X = A$. Then, using consecutive bits of IDs, it gradually increases the size of L and decreases the size of X, still preserving the invariant that $\mathrm{MRI}(L) < n/2$ and $\mathrm{MRI}(L \cup X) > n/2$ and $L \cap X = \emptyset$. Finally, after $O(\log N)$ repetitions, the size of X is equal to 1. The details are presented as Algorithm 11.

Algorithm 10 TestMaj(a, B)

---

if Emptiness(a, B)=true **then** return *minority* ▷ test if $|B| = 0$
if $a \in B$ **then** $\mathrm{dir}_a \leftarrow$ right **else** $\mathrm{dir}_a \leftarrow$ left
SingleRound; $p_1 \leftarrow$ dist()
if $p_1 = 0$ **then** return *majority* ▷ emptiness is already excluded, $|B| \in \{n/2, n\}$
SingleRound; $p_2 \leftarrow$ dist()
if $p_1 + p_2 > 1$ **then**
  return *majority*
else
  if $p_1 + p_2 = 1$ **then** return *halve* **else** return *minority*

Proposition 51. *The algorithm TestMaj satisfies the following conditions: if MRI$(B) < n/2$, it returns* minority*; if MRI$(B) = n/2$, it returns* halve*; if MRI$(B) > n/2$, it returns* majority.

Algorithm 11 Balance(a)

NMove(a) ▷ see Proposition 20 and Alg. 5
if no nontrivial move **then** return "n EVEN" **else** establish comm. sense of dir. ▷ Lemma 8
$L \leftarrow \emptyset$
$X \leftarrow A$
for $i = 1, 2, \ldots, \lceil \log(N+1) \rceil$ **do**
    $X_1 \leftarrow X \cap \{a \mid \mathrm{ID}_a[i] = 1, a \in A\}$
    $X_2 \leftarrow X \setminus X_1$
    $r_1 \leftarrow \mathrm{TestMaj}(a, L \cup X_1)$
    $r_2 \leftarrow \mathrm{TestMaj}(a, L \cup X_2)$
    if r_1 =halve or r_2 =halve **then** return "n is EVEN"
    if r_1 =majority **then**
        $X \leftarrow X_1$
    else
        if r_2 =majority **then**
            $X \leftarrow X_2$
        else
            $L \leftarrow L \cup X_1$, $X \leftarrow X_2$
if $a \in L$ **then** $\mathrm{dir}_a \leftarrow$ right **else** $\mathrm{dir}_a \leftarrow$ left
SingleRound, $p_1 \leftarrow \mathrm{dist}()$
if $a \in L \cup X$ **then** $\mathrm{dir}_a \leftarrow$ right **else** $\mathrm{dir}_a \leftarrow$ left
SingleRound, $p_2 \leftarrow \mathrm{dist}()$
if $p_1 + p_2 = 1$ **then** return "n is EVEN" **else** return "n is ODD"

Proposition 52. *The following condition holds before the $(i+1)$st iteration of the for-loop of the algorithm Balance: MRI$(L) < n/2$, MRI$(L \cup X) > n/2$, and there exist values $b_1, \ldots, b_i \in \{0,1\}$ such that $X \subseteq \{a \mid ID_a[j] = b_j$ for each $j \in [i]\}$.*

Proof. All these conditions are certainly satisfied for $i = 0$. Assume that they are satisfied before the $(i+1)$st iteration for $i > 0$. Then, they are also satisfied after that iteration of the for-loop, which follows from the way in which L and X are modified and the fact that TestMaj gives correct answers (Proposition 51). □

Theorem 53. *The algorithm Balance determines parity of the number of agents in $O(\log^2 N)$ rounds in the basic model and in $O(\log N)$ rounds in other models.*

Proof. Time complexity follows directly from the pseudo-code and the complexity of testing emptiness in various models (see Lemma 13). The answer given in line 2 is correct, since Proposition 20 assures a nontrivial move in NMove for odd value of n. Similarly, the answer given in line 10 is correct, since it confirms that the rotation index of $L \cup X_1$ or $L \cup X_2$ is equal to $n/2$. Thus, n is even, since rotation index is a natural number.

Proposition 52 implies that, after the for-loop, $\mathrm{MRI}(L) < n/2$, $\mathrm{MRI}(L \cup X) > n/2$ and $|X| = 1$ (since only one number satisfies conditions imposed on IDs of members of X). Let $\mathrm{MRI}(L) = r$. Then, $\mathrm{MRI}(L \cup X) \equiv (r+2) \bmod n$. Since $r < n/2$, we have $r + 2 < n$ and therefore $\mathrm{MRI}(L \cup X) = r + 2$. Thus, the sum of the rotation indexes from lines 19 and 21 is even. Thus, for odd n, this sum is not equal to n and the sum of traversed distances $p_1 + p_2$ cannot be equal to 1. This shows that, if n is odd, the algorithm gives a correct answer.

Assume that n is even. The only value of natural r such that $r < n/2$ and $r + 2 > n/2$ is $r = n/2 - 1$. Thus, for even n, the sum of rotation indexes from lines 19 and 21 is equal to n and the sum of traversed distances $p_1 + p_2$ is equal to 1 – the algorithm gives a correct result. $\square$

8. Randomized Algorithms

We say that a randomized algorithm solves a problem *with high probability* (whp) in time $O(f(n))$ if for each (large enough) problem of size n and each instance of this size, the problem is solved in time $O(f(n))$ with probability at least $1 - \frac{1}{n^c}$ for some constant $c \geq 1$.

Friedetzky et al. [5] considered the position discovery problem in anonymous geometric ring networks, i.e., under assumption that agents are indistinguishable (they do not have unique IDs).

Assume that a polynomial upper bound N on the actual number of agents n is given to agents. (That is, $n < N < n^c$ for some constant c.) Then, we can obtain randomized solutions to coordination problems and location discovery in the following way. First, each agent chooses randomly and independently its ID in the range $[N^2]$ with uniform distribution. Then, a deterministic solution presented in this paper to a considered problem is applied. Note that assigned IDs are unique whp. More precisely, the probability that a pair of agents choose a fixed ID $i \in [N^2]$ is $1/N^4$. By the union bound, the probability that a fixed pair of agents choose the same ID is $N \cdot \frac{1}{N^4} = \frac{1}{N^3}$. As there are $\binom{N}{2} < N^2$ pairs of agents, the probability that all IDs are different is at least $1 - 1/N > 1 - 1/n$. Thus, we obtain a solution with the same complexity as in the deterministic case with IDs, with high probability.

Now, assume that no estimation nor upper bound on n is provided. Consider a round in which each agent starts moving to right/left with probability 1/2. We call such a round *random*. The probability that such a round is a trivial move is $O(1/\sqrt{n})$, as shown in the proof of Theorem 29. Thus, after $O(1)$ rounds, we have a nontrivial move whp. Given a nontrivial move, we can solve the direction agreement problem in $O(1)$ rounds as well (Lemma 8).

As noticed in [5], the rotation index r of a random round satisfies $r \leq \sqrt{n}$ with probability $1 - 1/2^{\Theta(n)}$. (This fact can be verified using a Chernoff bound

[27]). Let the *normalized* rotation index of a round with rotation index r be equal to $\min\{r, n-r\}$. Recall that, with common sense of direction, agents can verify whether $r > n/2$ (see Lemma 2). And, by reversing directions of all agents, a round with rotation index $r > n/2$ can be changed into a round with rotation index $n-r < n/2$. Observe that, given the common sense of direction, the probability that a random round is $(n/2+c, n/2-c)$-round is

$$\binom{n}{n/2+c} 2^{-n} < \binom{n}{n/2} 2^{-n} = O(1/\sqrt{n})$$

where the last relationship is obtained by the Stirling formula. Using this fact we can show that, for any constant $c > 0$, there exists a constant c_1 such that the largest modified rotation index in c_1 random rounds is in the range $[c, \sqrt{n}]$ (for n large enough), with high probability.

Given the common sense of direction and a nontrivial move R with the rotation index $r \in [c, \sqrt{n}]$, we get an estimation of n in the following way. The round R is executed several times, until the sum of distances on which an agent is shifted gets at least 1. Then, the number m of repetitions of R is the estimate of n. As the rotation index r of R satisfies $r \in [c, \sqrt{n}]$ whp, we have $m \in [n/\sqrt{n}, n/c]$ whp. Let the above procedure be called *size estimation algorithm.*

Now, consider the following algorithm for location discovery:

1. Run random rounds until a nontrivial move is obtained. Solve the direction agreement problem, using a nontrivial move.

2. Execute the size estimation algorithm to determine an estimation m of the size n of the network.

3. Assign ID to each agent randomly in the range $[m^6]$, with uniform distribution.

4. Determine parity of n (see Theorem 53).

5. Solve the location discovery problem (provided a network works under assumptions of the lazy or perceptive model or n is odd), using the appropriate deterministic algorithm.

By the above discussion, this algorithm gives the following (almost) optimal result, which improves the $O(n \log^2 n)$ round algorithm for the basic and the perceptive model [5].

Theorem 54. *For each constant $c > 0$, the location discovery problem can be solved in anonymous networks:*

- *in time $n + n/c + O(\log n)$ in the lazy model;*
- *in time $n + n/c + O(\log^2 n)$ in the basic model, provided n is odd;*
- *in time $n/2 + n/c + O(\sqrt{n})$ in the perceptive model*

with high probability, where the actual number of rounds corresponding to the summand in the big-O notation depends on the constant c.

Proof. Steps 1 and 2 of the presented above algorithm work in $O(1)$ and $O(n/c)$ rounds respectively with high probability, as discussed before. Moreover, the estimation m of the size of the network obtained in step 2 belongs to the interval $[\sqrt{n}, n/c]$ with high probability. Steps 4 and 5 of the algorithm are executed for $N = m^6$, thus $N \geq n$ and $\log N = O(\log n)$ with high probability. Therefore, step 4 of the algorithm works in $O(\log^2 n)$ rounds in the basic model and in $O(\log n)$ rounds in other models with high probability, by Theorem 53.

At the beginning of step 5, the nontrivial move problem is solved with high probability. Therefore, the location discovery problem can be solved in $n + O(\log n)$ rounds in the basic model and in the lazy model, by Lemma 17. In the perceptive model, the location discovery problem can be solved in step 5 in $n/2 + O(\sqrt{n})$ rounds, by Lemmas 9, 44 and 46. □

For completeness, we also summarize the obtained complexities for coordination problems.

Theorem 55. *For each constant $c > 0$, and each of the models basic, lazy and perceptive of anonymous networks,*

- *the nontrivial move problem and the direction agreement problem can be solved in $O(1)$ rounds,*
- *the leader election problem can be solved in time $n/c + O(\log n)$,*

with high probability.

9. Distinguishers in the framework of combinatorial search problems

In this section we discuss relationships between distinguishers introduced in Section 4 and other (well established) combinatorial structures.

Suppose we have N coins with IDs $1, 2, \ldots, N$, at most n out of all N coins are counterfeit. We are allowed to ask queries defined by subsets of $[N]$. The answer to the query $Q \subseteq [N]$ is equal to the number of couterfeit coins in the set Q, i.e., the size of $Q \cap X$ where X is the set of IDs of counterfeit coins. The sequence of queries $Q_1, \ldots, Q_p \subseteq [N]$ solves the *nonadaptive conterfeit coin problem* with parameters n and N ((N, n)-CC for short) if the answers to queries $Q_1, \ldots, Q_p$ determine the set of counterfeit coins. In other words, the sequence of queries $Q_1, \ldots, Q_p \subseteq [N]$ solves (N, n)-CC if, for each $X_1, X_2 \subseteq [N]$ such that $|X_1|, |X_2| \leq n$, there exists $i \in [p]$ such that $|Q_i \cap X_1| \neq |Q_i \cap X_2|$.

Observe that the definition of (N, n)-distinguisher is a relaxed variant of (N, n)-CC, where we require that the queries differentiate **only** pairs of sets which are:
(a) *disjoint* and
(b) of size *exactly* n.
Therefore, each solution for (N, n)-CC is also a (N, n)-distinguisher. However,

a (N,n)-distinguisher does not necessarily satisfy requirements of a solution for (N,n)-CC. (Note that (a) or (b) alone does not impose a real relaxation with respect to the nonadaptive counterfeit coin problem.)

Intuitively, the relaxation of (N,n)-distinguisher with respect to (N,n)-CC is as follows. A solution for (N,n)-CC is supposed to determine the set of counterfeit coins uniquely based on exact sizes of intersections of query sets with the set of counterfeit coins. A (N,n)-distinguisher on the other hand deals with a set of $2n$ counterfeit coins split in two sets X_1, X_2 of size n. Then, the only outcome of the queries says whether the sizes of intersections of the query sets with X_1 and X_2 were equal or not. (One can think that we have just a beam balance instead of exact weight to compare the results of the query on X_1 and X_2.) And the goal is just to find a query set which makes the size of these intersections not equal.

The counterfeit coin problem and its various variants have been extensively studied, see e.g. [24, 28, 29]. The optimal size of a solution of (N,n)-CC is well known.

Theorem 56. *[24] The smallest number of queries solving (N,n)-CC for $n>1$ is $\Theta\left(\frac{nlog(N/n)}{\log n}\right)$.*

Theorem 56 directly implies that there exist (N,n)-distinguisher of size $\Theta\left(\frac{nlog(N/n)}{\log n}\right)$. Moreover, by concatening optimal solutions of $(N,2^i)$-CC for $i=1,2,\ldots,\lceil\log N\rceil$, we also obtain a strong (N,f)-distinguisher for $f(N,n)=O(n\log(N/n)/\log n)$. These observations give an alternative way to prove upper bounds from Corollary 31.[4] (Note that, similarly to our result, the upper bounds in [24] are non-constructive.) However, the lower bound on the size of an optimal solution for (N,n)-CC from Theorem 56 does not imply an analogous lower bound for distinguishers. Therefore, Lemma 25 extends the known bounds for the counterfeit coin problems on the relaxed variant defined by distinguishers.

Interestingly, there is also a close relationship between distinguishers and selective families studied e.g. in [25] and applied in various areas of computer science. Namely, one can show that every $(N,2n)$-selective family is a (standard) (N,n)-distinguisher. As the optimal size of a (N,n)-selective family is $\Theta(n\log(N/n))$ [25], direct application of a $(N,2n)$-selective family as a (N,n)-distinguisher gives a solution whose size is $\Theta(\log n)$ apart from the optimum.

10. Conclusions and open problems

In this paper, we evaluated complexity of location discovery and some coordination problems in synchronous geometric ring networks in considered settings.

[4] One can also try to apply a solution of (N,n)-CC from [24] to solve the nontrivial move problem. However, as the nontrivial move problem has to deal with specific constraints not present in the definition of distinguishers, we prefer a direct proof of Theorem 29.

In the basic and lazy model, we have shown an exponential gap in complexity of coordination problems between the cases that n is odd and even. Moreover, we established a relationship of these problems with some combinatorial search problems. In all cases in which the location discovery problem is feasible, we provided efficient, (almost) optimal solutions. An interesting open problem is to give explicit constructions for the nontrivial move problem for even n in the basic and lazy model. (In this paper, their existence is merely shown by the probabilistic method.) The exact complexity of the nontrivial move problem in the perceptive model is also not known; we have only provided the upper bound $O\left(\sqrt{n \log N} + (\log N)(\log n)\right)$.

We also do not know whether the additive term $\frac{n}{c}$ in solutions for the location discovery for anonymous networks is necessary (see Theorems 54, 55).

Acknowledgements

This work was supported by Polish National Science Centre grant DEC-2012/06/M/ST6/00459 and, in the final stage, by grant 2017/25/B/ST6/02010.

L. Gąsieniec and R. Martin acknowledge the support of the Networks Sciences and Technology (NeST) Initiative in the School of Electrical Engineering, Electronics and Computer Science at the University of Liverpool.

The authors also gratefully acknowledge anonymous reviewers for their valuable comments and suggestions improving the results and quality of presentation.

References

[1] H. Attiya, J. Welch, Distributed Computing, McGraw-Hill, 1998.

[2] E. Kranakis, D. Krizanc, E. Markou, The Mobile Agent Rendezvous Problem in the Ring, Morgan and Claypool Publishers, 2010.

[3] N. Lynch, Distributed Algorithms, Morgan Kaufmann Publishers, 1996.

[4] J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, O. M. Ponce, E. Pacheco, Position discovery for a system of bouncing robots, Inf. Comput. 244 (2015) 122–133. `doi:10.1016/j.ic.2015.07.005`.
URL `https://doi.org/10.1016/j.ic.2015.07.005`

[5] T. Friedetzky, L. Gąsieniec, T. Gorry, R. Martin, Observe and remain silent (communication-less agent location discovery), in: B. Rovan, V. Sassone, P. Widmayer (Eds.), MFCS, Vol. 7464 of Lecture Notes in Computer Science, Springer, 2012, pp. 407–418.

[6] I. Suzuki, M. Yamashita, Distributed anonymous mobile robots: Formation of geometric patterns, SIAM J. Comput. 28 (4) (1999) 1347–1363. `doi:10.1137/S009753979628292X`.

[7] S. Das, P. Flocchini, N. Santoro, M. Yamashita, On the computational power of oblivious robots: forming a series of geometric patterns, in: A. W. Richa, R. Guerraoui (Eds.), Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing, PODC 2010, Zurich, Switzerland, July 25-28, 2010, ACM, 2010, pp. 267–276. doi:10.1145/1835698.1835761.

[8] P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, Arbitrary pattern formation by asynchronous, anonymous, oblivious robots, Theor. Comput. Sci. 407 (1-3) (2008) 412–447. doi:10.1016/j.tcs.2008.07.026.

[9] M. Cieliebak, P. Flocchini, G. Prencipe, N. Santoro, Distributed computing by mobile robots: Gathering, SIAM J. Comput. 41 (4) (2012) 829–879. doi:10.1137/100796534.

[10] S. Susca, F. Bullo, S. Martínez, Synchronization of beads on a ring, in: Proc. IEE Conference on Decision and Control, 2007, pp. 4845–4850.

[11] B. Cooley, P. Newton, Iterated impact dynamics of n-beads on a ring, SIAM Review 47 (2) (2005) 273–300.

[12] M. A. Bender, D. K. Slonim, The power of team exploration: Two robots can learn unlabeled directed graphs, in: FOCS, IEEE Computer Society, 1994, pp. 75–85.

[13] J. Chalopin, P. Flocchini, B. Mans, N. Santoro, Network exploration by silent and oblivious robots, in: D. Thilikos (Ed.), Graph Theoretic Concepts in Computer Science - 36th International Workshop, WG 2010, Zarós, Crete, Greece, June 28-30, 2010 Revised Papers, Vol. 6410 of Lecture Notes in Computer Science, 2010, pp. 208–219. doi:10.1007/978-3-642-16926-7_20.
URL http://dx.doi.org/10.1007/978-3-642-16926-7

[14] P. Fraigniaud, L. Gąsieniec, D. Kowalski, A. Pelc, Collective tree exploration, Networks 48 (3) (2006) 166–177. doi:10.1002/net.20127.
URL http://dx.doi.org/10.1002/net.20127

[15] E. Arkin, S. Fekete, J. Mitchell, Approximation algorithms for lawn mowing and milling, Comput. Geom. 17 (1-2) (2000) 25–50. doi:10.1016/S0925-7721(00)00015-8.
URL http://dx.doi.org/10.1016/S0925-7721(00)00015-8

[16] J. Czyzowicz, A. Labourel, A. Pelc, Optimality and competitiveness of exploring polygons by mobile robots, Inf. Comput. 209 (1) (2011) 74–88. doi:10.1016/j.ic.2010.09.005.
URL http://dx.doi.org/10.1016/j.ic.2010.09.005

[17] X. Deng, T. Kameda, C. Papadimitriou, How to learn an unknown environment I: the rectilinear case, J. ACM 45 (2) (1998) 215–245. doi:

10.1145/274787.274788.
URL http://doi.acm.org/10.1145/274787.274788

[18] F. Hoffmann, C. Icking, R. Klein, K. Kriegel, The polygon exploration problem, SIAM J. Comput. 31 (2) (2001) 577–600. doi:10.1137/S0097539799348670.
URL http://dx.doi.org/10.1137/S0097539799348670

[19] J. Czyzowicz, L. Gąsieniec, A. Kosowski, E. Kranakis, O. M. Ponce, E. Pacheco, Position discovery for a system of bouncing robots, in: M. Aguilera (Ed.), Distributed Computing - 26th International Symposium, DISC 2012, Salvador, Brazil, October 16-18, 2012. Proceedings, Vol. 7611 of Lecture Notes in Computer Science, Springer, 2012, pp. 341–355. doi:10.1007/978-3-642-33651-5_24.

[20] J. Czyzowicz, E. Kranakis, E. Pacheco, Localization for a system of colliding robots, in: F. Fomin, R. Freivalds, M. Kwiatkowska, D. Peleg (Eds.), Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II, Vol. 7966 of Lecture Notes in Computer Science, Springer, 2013, pp. 508–519. doi:10.1007/978-3-642-39212-2_45.
URL http://dx.doi.org/10.1007/978-3-642-39212-2

[21] J. Czyzowicz, S. Dobrev, E. Kranakis, E. Pacheco, Survivability of swarms of bouncing robots, in: A. Pardo, A. Viola (Eds.), LATIN 2014: Theoretical Informatics - 11th Latin American Symposium, Montevideo, Uruguay, March 31 - April 4, 2014. Proceedings, Vol. 8392 of Lecture Notes in Computer Science, Springer, 2014, pp. 622–633. doi:10.1007/978-3-642-54423-1_54.
URL http://dx.doi.org/10.1007/978-3-642-54423-1

[22] J. Czyzowicz, E. Kranakis, E. Pacheco, D. Pajak, Information spreading by mobile particles on a line, in: Structural Information and Communication Complexity - 22nd International Colloquium, SIROCCO 2015, Montserrat, Spain, July 14-16, 2015, Post-Proceedings, 2015, pp. 285–298. doi:10.1007/978-3-319-25258-2_20.
URL https://doi.org/10.1007/978-3-319-25258-2_20

[23] H. Ando, I. Suzuki, M. Yamashita, Formation and agreement problems for synchronous mobile robots with limited visibility, in: Intelligent Control, 1995., Proceedings of the 1995 IEEE International Symposium on, 1995, pp. 453–460. doi:10.1109/ISIC.1995.525098.

[24] V. Grebinski, G. Kucherov, Optimal reconstruction of graphs under the additive model, Algorithmica 28 (1) (2000) 104–124.

[25] A. Clementi, A. Monti, R. Silvestri, Distributed broadcast in radio networks of unknown topology, Theor. Comput. Sci. 302 (1-3) (2003) 337–364.

[26] P. Frankl, Z. Füredi, Forbidding just one intersection, J. Comb. Theory, Ser. A 39 (2) (1985) 160–176.

[27] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.

[28] N. H. Bshouty, Optimal algorithms for the coin weighing problem with a spring scale, in: COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009, 2009.
URL `http://www.cs.mcgill.ca/%7Ecolt2009/papers/004.pdf#page=1`

[29] G. D. Marco, D. R. Kowalski, Searching for a subset of counterfeit coins: Randomization vs determinism and adaptiveness vs non-adaptiveness, Random Struct. Algorithms 42 (1) (2013) 97–109. `doi:10.1002/rsa.20417`.
URL `https://doi.org/10.1002/rsa.20417`