Optimal Gathering by Asynchronous Oblivious Robots in Hypercubes

Kaustav Bose, Manash Kumar Kundu(✉), Ranendu Adhikary, and Buddhadeb Sau

Department of Mathematics, Jadavpur University, Kolkata, India
{kaustavbose.rs,manashkrkundu.rs,
ranenduadhikary.rs}@jadavpuruniversity.in, bsau@math.jdvu.ac.in

Abstract. We consider the problem of gathering a set of autonomous, identical, oblivious, asynchronous, mobile robots at a vertex of an anonymous hypercube. The robots operate in Look-Compute-Move cycles. In each cycle, a robot takes a snapshot of the current configuration (*Look*), then based on the perceived configuration, decides whether to stay idle or to move to an adjacent vertex (*Compute*), and in the later case makes an instantaneous move accordingly (*Move*). We have shown that the problem is unsolvable if the robots do not have multiplicity detection capability. With weak multiplicity detection capability, the problem is solvable in an oriented hypercube for any initial configuration of $2k+1 (k > 0)$ number of robots. For $4k (k > 0)$ number of robots, the problem is solvable under the same assumptions if and only if the group of automorphism of the configuration is trivial. Our proposed algorithms are optimal with respect to the total number of moves executed by the robots.

Keywords: Distributed computing · Autonomous robots · Gathering · Hypercube · Weber point · Asynchronous · Look-Compute-Move cycle

1 Introduction

The *gathering* problem requires a set of n mobile computational entities, usually called *robots* or *agents*, initially situated at different locations in a spatial universe, to gather at some unspecified location within finite time. When only two robots are involved, the problem is usually referred to as the *rendezvous* problem. In distributed computing, gathering has been extensively studied both in continuous and in discrete domains. In the continuous setting, the robots operate in the two-dimensional Euclidean space and in the discrete case, they operate in a network modeled as a graph. In the discrete setting, the problem has been previously studied in different graph topologies, e.g. rings [10,18,22,23], grids [8,31], trees [8] etc. The problem is particularly difficult in graphs that are highly symmetric and is solvable only in very limited cases. Hence, for characterization of gatherability, it is important to investigate the problem in highly symmetric graphs. In this paper, we investigate the problem in a hypercube graph.

S. Gilbert et al. (Eds.): ALGOSENSORS 2018, LNCS 11410, pp. 102–117, 2019.
https://doi.org/10.1007/978-3-030-14094-6_7

1.1 The Model

A set of autonomous mobile robots is randomly deployed on the vertices of a d-dimensional hypercube network. The *d-dimensional hypercube* Q_d is an undirected graph with vertex set $V(Q_d) = \mathbb{Z}_2^d = \{0,1\}^d$, and two vertices are adjacent if and only if the two binary strings differ in exactly one coordinate. An *oriented hypercube* is an edge-labeled hypercube with the so-called *dimensional labeling* $\lambda : E(Q_d) \rightarrow \{1, \ldots, d\}$ where $\lambda(uv) = i$, if u and v differ in the ith coordinate. We shall denote an oriented hypercube by $Q_d^{\mathcal{O}}$, and an unoriented hypercube by simply Q_d. The binary string labels of the vertices are for descriptive purposes, and are not known to the robots. However, in an oriented hypercube, the edge-labels are known to the robots. It is traditionally assumed that the robots can only perceive the labels of the edges adjacent to the vertex on which it resides. But since the labels of edges adjacent to a single vertex determine the dimensional labels of all the edges in a hypercube (See Theorem 3.1 in [33]), we assume without loss of generality that the robots know the labels of all the edges.

The robots are *oblivious* (they have no memory of past configurations and previous actions), *autonomous* (there is no central control), *homogeneous* (they execute the same distributed algorithm), *anonymous* (they have no unique identifiers) and *identical* (they are indistinguishable by their appearance). The robots have *global visibility*, i.e., they are able to perceive the entire graph. The robots do not agree on any global coordinate system. Furthermore, there are no means of communication between the robots.

The robots, when active, operate according to the so-called LOOK-COMPUTE-MOVE cycle. In each cycle, a previously idle or inactive robot wakes up and executes the following steps. In the LOOK phase, the robot takes the snapshot of the positions of all the robots, represented in its own coordinate system. Based on the perceived configuration, the robot performs computations according to a deterministic algorithm to decide whether to stay idle or to move to an adjacent vertex. Based on the outcome of the algorithm, the robot either remains stationary or makes an instantaneous move to an adjacent vertex. Since the moves are instantaneous, it implies that the robots are always seen on vertices, not on edges. In the fully synchronous setting (FSYNC), the activation phase of all robots can be logically divided into global rounds, where all the robots are activated in each round. The semi-synchronous (SSYNC) model coincides with the FSYNC model with the only difference that not all robots are necessarily activated in each round. The most general type of scheduler is the asynchronous scheduler (ASYNC). In ASYNC, the robots are activated independently, and the amount of time spent in LOOK, COMPUTE, MOVE and inactive states are finite but unbounded and unpredictable. As a result, the robots do not have a common notion of time.

An important capability associated to the robots is multiplicity detection. During the LOOK phase, a robot may perceive a vertex occupied by more than one robot in different ways. In *strong multiplicity detection*, the robots perceive the actual number of robots in each vertex. In *weak multiplicity detection*, the robots are only able to detect whether a vertex is occupied by more than one robot, but not the exact number. If the robots have no multiplicity detection capability, they can only decide if a vertex is occupied or empty.

1.2 Related Works

The gathering problem has been extensively studied in continuous domain under various assumptions [1,4–7,16,29,30]. In discrete domains, both gathering and rendezvous have been studied in different graph topologies [2,8,11,13–15,22,23,25,27,28,32]. The problem of gathering two robots on an anonymous ring was studied in [12,26,28]. The problem for more than two robots was studied in [15]. In [15], the robots had memory and used tokens to break symmetry. In [23], the problem was first considered in a very minimal setting with identical, asynchronous, memoryless robots without tokens or any kind of communication capability. They proved that without multiplicity detection, gathering is impossible on rings for $n \geq 2$ robots. With weak multiplicity detection capability, they solved the problem for all configurations with an odd number of robots, and all the asymmetric configurations with an even number of robots by different algorithms. In [22], symmetric configurations with an even number of robots were studied, and the problem was solved for more than 18 robots. Some of the remaining configurations were solved in [9,18,24] in separate algorithms. In [10], a single unified algorithm was proposed, that achieves gathering for all gatherable initial configurations except some potentially gatherable configurations with 4 robots. The problem was studied with weak local multiplicity detection in [19–21]. A full characterization of gatherable configurations for finite grids and trees with weak multiplicity detection was provided in [8]. Gathering in finite grids in presence of crash-faults was studied in [3]. Optimal gathering in infinite grid with strong multiplicity detection was studied in [31].

2 Theoretical Preliminaries

2.1 Group of Automorphisms

An *automorphism* of a graph $G = (V, E)$ is a bijection $\varphi : V \longrightarrow V$ such that for all $u, v \in V$, u, v are adjacent if and only if $\varphi(u), \varphi(v)$ are adjacent. The set of all automorphisms of G forms a group, called the *automorphism group* of G and is denoted by $Aut(G)$.

The automorphism group of a hypercube is generated by two types of automorphisms, namely *translation* and *rotation*.

Translation: For $a \in \mathbb{Z}_2^d$, the map $\tau_a : V(Q_d) \longrightarrow V(Q_d)$ given by $u \mapsto u \oplus a$ is called *translation by* a. Here, $u \oplus a$ is the vertex obtained by adding the binary strings u and a componentwise. The set $T = \{\tau_a \mid a \in \mathbb{Z}_2^d\}$ of all translations forms a subgroup of $Aut(Q_d)$.

Rotation: For $\sigma \in S_d$, the map $r_\sigma : V(Q_d) \longrightarrow V(Q_d)$ given by $u \mapsto \sigma(u)$ is called *rotation by* σ, where $\sigma(u)$ is the vertex obtained by permuting the binary string u by $\sigma : \{1, \ldots, d\} \longrightarrow \{1, \ldots, d\}$. The set $R = \{r_\sigma \mid \sigma \in S_d\}$ of all rotations forms a subgroup of $Aut(Q_d)$.

Theorem 1 [17]. *$Aut(Q_d) = TR$.*

Hence, for any automorphism $\varphi \in Aut(Q_d)$, $\exists$ a unique pair $(a, \sigma) \in (\mathbb{Z}_2^d, S_d)$, such that $\varphi : V \longrightarrow V$ can be written as $u \mapsto \sigma(u) \oplus a$.

It is easy to see that $|T| = 2^d$ and $|R| = d!$. Since $T \cap R$ is trivial, $|TR| = |T||R|/|T \cap R| = 2^d d!$. Therefore, we have the following corollary.

Corollary 1 [17]. *$|Aut(Q_d)| = 2^d d!$.*

The definition of automorphism of graphs can be extended to edge-labeled graphs in a natural way. Given an edge-labeled graph $G = (V, E, \lambda)$ with edge-labeling $\lambda : E \longrightarrow \mathbb{N}$, an automorphism of G is a bijection $\varphi : V \longrightarrow V$ such that for all $u, v \in V$, $\varphi(u)\varphi(v) \in E$ if and only if *(1)* $uv \in E$ and *(2)* $\lambda(\varphi(u)\varphi(v)) = \lambda(uv)$. In view of this definition, it is easy to see that the dimensional labeling of a hypercube kills all rotational automorphisms. Thus the automorphism group of an oriented hypercube consists of only translations.

Theorem 2. *$Aut(Q_d^{\mathcal{O}}) = T$.*

2.2 Feasibility of Gathering

Consider a set of robots placed on the vertices of a simple undirected connected graph $G = (V, E)$. Define a function $f : V \longrightarrow \mathbb{N} \cup \{0\}$, where $f(v)$ is the number of robots on the vertex v. The pair (G, f) is called a *configuration of robots on* G, or simply a *configuration*. If all the robots in a configuration reside on a single vertex, then it is called *final configuration*; otherwise it is called a *non-final configuration*. Given a configuration (G, f), we define the *multiplicity function* $\tilde{f}$ in the following way. If the model assumes robots with strong multiplicity detection capability, then $\tilde{f}(v) = f(v)$ for all $v \in V$. If the robots have weak multiplicity detection capability, then $\tilde{f} : V \longrightarrow \{0, 1, 2\}$ is defined as,

$$\tilde{f}(v) = \begin{cases} 0 & \text{if } v \text{ is an empty vertex} \\ 1 & \text{if } v \text{ is occupied by exactly one robot} \\ 2 & \text{if } v \text{ is a multiplicity.} \end{cases}$$

If the robots have no multiplicity detection capability, then $\tilde{f} : V \longrightarrow \{0, 1\}$ is defined as,

$$\tilde{f}(v) = \begin{cases} 0 & \text{if } v \text{ is an empty vertex} \\ 1 & \text{if } v \text{ is occupied by at least one robot.} \end{cases}$$

Given a configuration (G, f), the pair $(G, \tilde{f})$ is called the *perceived configuration*.

An *automorphism of a perceived configuration* $(G, \tilde{f})$ is a graph automorphism $\varphi \in Aut(G)$ such that $\tilde{f}(v) = \tilde{f}(\varphi(v))$ for all $v \in V$. The set of all automorphisms of $(G, \tilde{f})$ also forms a group that will be denoted by $Aut(G, \tilde{f})$.

If $|Aut(G, \tilde{f})| = 1$, we say that $(G, \tilde{f})$ is *asymmetric*, otherwise it is said to be *symmetric*.

For an automorphism $\varphi \in Aut(G, \tilde{f})$, let $<\varphi> \subseteq Aut(G, \tilde{f})$ be the cyclic subgroup generated by φ. Elements of this group are $\{\varphi^0, \varphi^1, \varphi^2, \ldots, \varphi^{p-1}\}$, where φ^0 is the identity, $\varphi^k = \underbrace{\varphi \circ \varphi \circ \cdots \circ \varphi}_{k\,\text{times}}$ and p is the order of φ.

For any subgroup H of $Aut(G, \tilde{f})$, define the equivalence relation on V given by: $x \sim y$ if and only if $x = \varphi(y)$ for some $\varphi \in H$. This equivalence relation induces a partition on V. The *orbit of a vertex* $v \in V$ *under the action of* H is the set $H_v = \{\sigma(v) | \sigma \in H\}$, which is the corresponding equivalence class containing v.

Partitive Automorphism: Let $\mathcal{C} = ((V, E), \tilde{f})$ be a perceived configuration. A non-trivial automorphism $\varphi \in Aut(\mathcal{C})$ is said to be *partitive* on V if $|H_v| = p$ for all $v \in V$, where $p > 1$ is the order of φ and $H = <\varphi>$.

Lemma 1. *In Q_d, any non-trivial translation is partitive.*

Theorem 2 in [32], stated for configurations of robots with strong multiplicity detection capability can be easily generalized to the following theorem.

Theorem 3. *Let $\mathcal{C} = ((V, E), \tilde{f})$ be a non-final perceived configuration. If there exists a $\varphi \in Aut(\mathcal{C})$ partitive on V, then $\mathcal{C}$ is not gatherable.*

Theorem 4. *Without multiplicity detection capability, gathering in (both oriented and unoriented) hypercubes is not deterministically solvable in SSYNC.*

Proof. Assume that there exists a correct gathering algorithm $\mathcal{A}$. In the SSYNC model, time can be logically divided into discrete global rounds. So, starting from some non-final initial configuration, consider a synchronous execution of algorithm $\mathcal{A}$, in which gathering is achieved in round t.

Case 1: Suppose that in round $t - 1$, exactly two vertices in Q_d are occupied. Hence, the perceived configuration in round $t - 1$ is $(Q_d, \tilde{f})$ where $\tilde{f}(v) = \tilde{f}(w) = 1$ for two distinct vertices $v, w \in V(Q_d)$, and $\tilde{f}(u) = 0 \; \forall u \in V(Q_d) \setminus \{v, w\}$. But, then the perceived configuration $(Q_d, \tilde{f})$ admits a partitive automorphism given by $x \mapsto x \oplus v \oplus w$. Hence by Theorem 3, gathering can not be deterministically achieved from this configuration.

Case 2: Assume that at least three vertices in Q_d are occupied in round $t - 1$. Then algorithm $\mathcal{A}$ brings all the robots to a common vertex, say u, in one step. But the adversary can choose to activate all the robots except one that is not placed at u. Then all but one robot will reach u. This will create a configuration with exactly two vertices occupied. Since this configuration admits a partitive automorphism, gathering can not be deterministically achieved from here by Theorem 3. □

Corollary 2. *Without multiplicity detection capability, gathering in (both oriented and unoriented) hypercubes is not deterministically solvable in ASYNC.*

2.3 Weber Point

Given a configuration (G, f), with $G = (V, E)$, the centrality of $v \in V$ is defined as $c_{G,f}(v) = \sum\limits_{u \in V} d(u, v) \cdot f(u)$. When there is no ambiguity, we shall write $c_f(v)$, or simply $c(v)$.

Weber Point: Given a configuration $\mathcal{C} = (G = (V, E), f)$, a vertex $v \in V$ is a *Weber point* of $\mathcal{C}$, if $c(v) = min\{c(u) | u \in V\}$.

By definition, a Weber point is a vertex with minimum centrality. In other words, a vertex $w \in V$ is a Weber point if the sum of the lengths of the shortest paths from all robots to w is minimum. Therefore, an algorithm that gathers all the robots at a Weber point via the shortest paths is optimal with respect to the total number of moves performed by the robots. However, a configuration may have more than one Weber point. Given a configuration (G, f), we shall denote the set of Weber points by $\mathcal{W}_{G,f}$, or simply $\mathcal{W}_f$ or $\mathcal{W}$ when there is no ambiguity.

Theorem 5 [32]**.** *Let (G, f) be a configuration with Weber points $\mathcal{W}_f$. If a robot moves towards a Weber point $w \in \mathcal{W}_f$, resulting in a new configuration (G, f'), then*

1. $c_{f'}(v) = c_f(v) - 1$ *for each* $v \in \mathcal{W}_{f'}$
2. $w \in \mathcal{W}_{f'}$
3. $\mathcal{W}_{f'} \subseteq \mathcal{W}_f$.

We shall now discuss about the Weber points of configurations on a hypercube. Consider a set of n robots $\{r_1, r_2, \ldots, r_n\}$ on a d-dimensional hypercube Q_d. Suppose that the robots $r_1, r_2, \ldots, r_n$ are placed on the vertices $v_1, v_2, \ldots, v_n$ respectively. For $i = 1, 2, \ldots, n$, let the binary string representation of v_i be $b_{i1}b_{i2}\ldots b_{id}$, where $b_{ij} \in \{0, 1\}$. For $j = 1, 2, \ldots, d$, let us define sets $[b]_j \subseteq \{0, 1\}$ in the following way.

$$[b]_j = \begin{cases} \{0\}, & \text{if the number of 0's in the multiset } \{b_{1j}, b_{2j}, \ldots, b_{nj}\} \text{ is more} \\ & \text{than the number of 1's} \\ \{1\}, & \text{if the number of 1's in the multiset } \{b_{1j}, b_{2j}, \ldots, b_{nj}\} \text{ is more} \\ & \text{than the number of 0's} \\ \{0, 1\}, & \text{if the multiset } \{b_{1j}, b_{2j}, \ldots, b_{nj}\} \text{ has equal number of 0's and 1's.} \end{cases}$$

In Theorem 6, we show that the set of Weber points of the configuration is

$$\mathcal{W} = [b]_1 \times [b]_2 \times \ldots \times [b]_d.$$

For instance, consider a configuration of a set of 8 robots $\{r_1, r_2, \ldots, r_8\}$ on a 4-dimensional hypercube Q_4. Suppose that the robots are positioned on the following vertices respectively: 0110, 0111, 1000, 1110, 0000, 0001, 1110, 1101. Then the set of Weber points of this configuration is given by

$$\begin{aligned} \mathcal{W} &= \{0, 1\} \times \{1\} \times \{0, 1\} \times \{0\} \\ &= \{0100, 0110, 1100, 1110\}. \end{aligned}$$

Theorem 6. *Let $\{r_i\}_{i=1}^n$ be a set of robots placed on the vertices of Q_d with binary string representations $\{b_{i1}b_{i2}\dots b_{id}\}_{i=1}^n$ respectively. Then the set of Weber points of the configuration is $\mathcal{W} = [b]_1 \times [b]_2 \times \dots \times [b]_d$.*

Proof. The distance between any two vertices in Q_d is the number of positions in which their binary string representations differ. Then it can be easily seen that (1) the centrality of all $w \in [b]_1 \times [b]_2 \times \dots \times [b]_d$ are equal, (2) the centrality of any $v \in V(Q_d) \backslash [b]_1 \times [b]_2 \times \dots \times [b]_d$ is strictly greater than the centrality of $w \in [b]_1 \times [b]_2 \times \dots \times [b]_d$. □

Corollary 3. *The subgraph induced by the set of Weber points $\mathcal{W}$ of a configuration on a hypercube Q_d is also a hypercube.*

Corollary 4. *The number of Weber points of a configuration on a hypercube Q_d is 2^k, where $0 \leq k \leq d$.*

2.4 Leading Weber Point

A configuration of robots on a hypercube can have more than one Weber point. We want to devise an algorithm that gathers all the robots at one of the Weber points via the shortest paths. Our proposed algorithm requires to solve a subproblem called LeadingWeberPoint. Let us formally define the problem LeadingWeberPoint. Consider a configuration in which no vertex contains more than one robot, and that has no partitive automorphism. Let $\mathcal{W}$ be the set of Weber points of this configuration. The problem LeadingWeberPoint asks to devise an algorithm so that every robot that perceives this configuration in its local view, deterministically elects a unique Weber point $w_\ell \in \mathcal{W}$. We shall call the vertex w_ℓ the *leading Weber point.*

Since we have assumed that the robots are positioned at distinct vertices, there is no distinction between the configuration and the perceived configuration. In other words, given such a configuration (G, f), we have $\tilde{f} = f$. A vertex $v \in V$ is called a *fixed vertex* if $\varphi(v) = v$, $\forall \varphi \in Aut(G, f)$.

Theorem 7. LeadingWeberPoint *can be deterministically solved only if $\mathcal{W}$ has at least one fixed vertex.*

Proof. See Appendix A.

Theorem 8. LeadingWeberPoint *may not be deterministically solvable in an unoriented hypercube.*

Proof. Consider a configuration (Q_5, f) of a set of 14 robots on the 5-dimensional unoriented hypercube Q_5. The robots are placed on the following vertices: 00100, 00001, 11000, 10010, 01100, 01010, 00101, 00011, 11010, 10110, 11001, 10101, 01111, 11111. It is easy to see that $\mathcal{W}_f = V(Q_5)$. It can be shown that $Aut(Q_5, f)$ $= \{e, \varphi_1, \varphi_2, \varphi_3\}$, with each φ_i given by $u \mapsto \sigma_i(u) \oplus a_i$, where $a_i \in \mathbb{Z}_2^5, \sigma_i \in S_5$ are the following: $a_1 = 00000$, $\sigma_1 = (1)(24)(35)$, $a_2 = 10000$, $\sigma_2 = (1)(2543)$, $a_3 = 10000$, $\sigma_3 = (1)(2345)$. Then it can be easily verified that (1) there is no partitive automorphism in $Aut(Q_5, f)$, (2) there is no fixed vertex in $\mathcal{W}_f = V(Q_5)$. So by Theorem 7, LeadingWeberPoint is deterministically unsolvable. □

Now we show that LeadingWeberPoint can be deterministically solved in an oriented hypercube.

Lemma 2. *Given a vertex u_0 in an oriented hypercube $Q_d^{\mathcal{O}}$, $\exists$ exactly one coordinate assignment (bijection) $\Psi : V(Q_d^{\mathcal{O}}) \longrightarrow \{0,1\}^d$ such that*

1. $\Psi(u_0) = 00\ldots0 \in \{0,1\}^d$
2. u, v are adjacent if and only if $\Psi(u), \Psi(v)$ differ in exactly one bit
3. for $uv \in E(Q_d^{\mathcal{O}})$, $\lambda(uv) = i$ if and only if $\Psi(u), \Psi(v)$ differ in the ith position.

Proof. The coordinates given to u_0 are $00\ldots0$. Then by rule (2) and (3), the coordinates of all its neighbors are uniquely determined. If the coordinates of all vertices at distance $i (< d)$ from u_0 are uniquely determined, then again by rule (2) and (3), the coordinates of all vertices at distance $i+1$ can be determined uniquely. Hence by induction, the coordinates assigned to all the vertices are unique. □

Now for any $w \in V(Q_d^{\mathcal{O}})$, we define a binary string $\zeta(w)$ of length 2^d in the following the way:

1. First, give $Q_d^{\mathcal{O}}$ the unique coordinate assignment $\Psi : V(Q_d^{\mathcal{O}}) \longrightarrow \{0,1\}^d$ with $\Psi(w) = 00\ldots0$.
2. Now we define a total ordering $\prec$ on $V(Q_d^{\mathcal{O}})$ as: $u \prec v$

$$u \prec v \Leftrightarrow \begin{cases} d(u,w) < d(v,w) \\ Or, \\ d(u,w) = d(v,w), \text{and } \Psi(u) \text{ is lexicographically larger that } \Psi(v), \end{cases}$$

   where $d(u,w)$ is the distance of u from w. For example, when $d = 4$, the assigned coordinates of the vertices written in increasing order will be: $\underbrace{0000}_{\text{distance 0}}$, $\underbrace{1000, 0100, 0010, 0001}_{\text{distance 1}}$, $\underbrace{1100, 1010, 1001, 0110, 0101, 0011}_{\text{distance 2}}$, $\underbrace{1110, 1101, 1011, 0111}_{\text{distance 3}}$, $\underbrace{1111}_{\text{distance 4}}$.
3. Finally, scan the vertices of the hypercube according to the above ordering. For each vertex, put a 0 if it is empty, or 1 if it is occupied by a robot. Recall that any vertex can be occupied by at most one robot. The string of length 2^d thus obtained is $\zeta(w)$. In the previous example, if the occupied vertices are 0000, 1000, 0010, 1001, 0011, 1011, 1111, then $\zeta(w) = 1101000100100101$.

Lemma 3. *For any two distinct vertices $u, v \in V(Q_d^{\mathcal{O}})$, if $\zeta(u) = \zeta(v)$, then the configuration has a partitive automorphism.*

Proof. It can be easily seen that if $\zeta(u) = \zeta(v)$, then the configuration has the automorphism (translation) given by $x \mapsto x \oplus u \oplus v$. □

Theorem 9. LeadingWeberPoint *is solvable in an oriented hypercube.*

Proof. Since the configuration has no partitive automorphism, $\zeta(w_1) \neq \zeta(w_2)$ for any distinct $w_1, w_2 \in \mathcal{W}$. Hence the robots can unanimously elect $w \in \mathcal{W}$ with the lexicographically (strictly) largest $\zeta(w)$ as the leading Weber point. □

3 The Algorithm

Our plan is to solve the problem in two stages. In stage 1, we create a multiplicity at a Weber point and then in stage 2, we sequentially bring the remaining robots to that vertex. Before describing the algorithm, we first give two definitions.

Anchor: Let $(Q_d^{\mathcal{O}}, \tilde{f})$ be a non-final perceived configuration on an oriented hypercube with at most one multiplicity and no partitive automorphism. The *anchor* of $(Q_d^{\mathcal{O}}, \tilde{f})$ is a vertex $\alpha \in V(Q_d^{\mathcal{O}})$ defined as the following. If $(Q_d^{\mathcal{O}}, \tilde{f})$ has no multiplicity, then α is the leading Weber point; otherwise α is the unique vertex with multiplicity. Note that all the robots observing the configuration $(Q_d^{\mathcal{O}}, \tilde{f})$ agree on which vertex is the anchor.

Leader: Since all the robots observing the configuration $(Q_d^{\mathcal{O}}, \tilde{f})$ agree on the anchor α, they also agree on a common coordinate system, which is the unique coordinate system Ψ described in Lemma 2 with $\Psi(\alpha) = 00\dots0$. This also allows the robots to order the vertices of the hypercube as described in the previous section. In this ordering, the first robot appearing on a non-anchor vertex will be called the *leader*.

3.1 $\mathbf{2k+1}$ $(\mathbf{k > 0})$ Robots

Theorem 10. *Any configuration on a hypercube with odd number of robots has exactly one Weber point.*

Proof. Using the same notations as in Theorem 6, the set of Weber points is given by $\mathcal{W} = [b]_1 \times [b]_2 \times \dots \times [b]_d$. Since there are odd number of robots, the multiset $\{b_{1j}, b_{2j}, \dots, b_{nj}\}$ can never have equal number of 0's and 1's. Hence, $[b]_j = \{0\}$ or $\{1\}$, $\forall j \in \{1, \dots, d\}$. Thus $|\mathcal{W}| = 1$. □

Theorem 11. *Gathering in $Q_d^{\mathcal{O}}$ is optimally solvable in ASYNC with weak multiplicity detection for any configuration of odd number of robots.*

Proof. We simply ask only the leader to move towards the anchor. The anchor α is the unique Weber point of the configuration. As the leader moves towards it, the Weber point remains invariant by Theorem 5. After one or two robots reach α, a multiplicity is created at α. Throughout stage 2, α remains the unique multiplicity in the configuration, since only the leader moves. Thus, all the remaining robots will sequentially reach α. The algorithm is clearly optimal with respect to the total number of moves executed by the robots. □

Algorithm 1. Gathering for $4k$ $(k > 0)$ robots

```
Procedure GATHER()
    $\alpha \leftarrow$ anchor
    if $\alpha$ is a multiplicity then
        if I am leader then
            Move towards $\alpha$
    else
        if I am leader then
            if I am on a Type 1 vertex then
                Move towards $\alpha$ via a Type 1 edge
            else if I am on a Type 2 vertex then
                Move towards $\alpha$
```

3.2 4k (k > 0) Robots

In view of Theorems 2, 3 and Lemma 1, any configuration with non-trivial automorphism group is ungatherable. We show that for all the remaining configurations, gathering can be optimally solved. Again our strategy is to move the leader towards the anchor. However, in this case the leader has to judiciously choose the edge via which it should approach the anchor. Unlike the previous case, the anchor may change after a move.

Consider the first stage of the algorithm, when there is no multiplicity in the configuration. Then the anchor is the leading Weber point w_ℓ. We classify all the non-anchor vertices into two types: *type 1* and *type 2*. If the configuration has 2^m $(0 \leq m \leq d)$ Weber points, then among the d neighbors of w_ℓ, m are also Weber points. This is because of Lemma 3. Let us call these Weber points $w_1, \ldots, w_m$. Since the coordinates assigned to w_ℓ are $0 \ldots 0$, the coordinates of each $w_i \in \{w_1, \ldots, w_m\}$ have exactly one 1. For each w_i, assume that its assigned coordinates have the 1 at the p_ith place, which implies that the edge joining w_i and w_ℓ has label p_i. Also, the set of Weber points, in the assigned coordinates, is given by $\mathcal{W} = [b]_1 \times [b]_2 \times \ldots \times [b]_d$, where $[b]_l$ is $\{0, 1\}$ if $l \in \{p_1, \ldots, p_m\}$, and $\{0\}$ otherwise.

Type 1 Vertex: A non-anchor vertex v will be called a *type 1 vertex* if the following holds: there is at least one $p_i \in \{p_1 \ldots p_m\}$ such that the assigned coordinates of v have 1 at p_ith place. Also the edge incident to v with label p_i will be called a *type 1 edge.*

Type 2 Vertex: If a non-anchor vertex v is not type 1, then it will be called a *type 2 vertex.* This implies that the p_1th, $\ldots p_m$th terms of the assigned coordinates of v are 0. Note that if the configuration has only one Weber point, i.e., $m = 0$, then all non-anchor vertices are vacuously type 2.

Theorem 12. *Let $(Q_d^{\mathcal{O}}, f)$ be a configuration of $4k$ $(k > 0)$ robots with no multiplicities and no partitive automorphisms. Let w_ℓ be the leading Weber point, and hence the anchor. Assume that the leader r is placed at a type 1 vertex u. Suppose that it moves via a type 1 edge with label p_i to an empty vertex v, and gives rise to configuration $(Q_d^{\mathcal{O}}, f')$. Then the following holds.*

1. $w_\ell \in \mathcal{W}_{f'}$
2. If $|\mathcal{W}_f| = 2^m (m > 0)$*, then* $|\mathcal{W}_{f'}| = 2^{m-1}$
3. $(Q_d^{\mathcal{O}}, f')$ *has no partitive automorphism.*

Proof

(1) Assume that r moves via a type 1 edge with label p_i. Then the assigned coordinates of u and v differ in exactly one bit at the p_ith position. At the p_ith place, u has 1, while v has 0. Then v has less 1's than u, and hence v is closer to w_ℓ than u, i.e., r has moved towards w_ℓ. Hence, by Lemma 5, $w_\ell \in \mathcal{W}_{f'}$.

(2) It is easy to see that, as r moves from u to v, (i) its distance from all Weber points whose coordinates have 1 at p_ith place (there are 2^{m-1} of them), increases by one, and (ii) its distance from all Weber points whose coordinates have 0 at p_ith place, decreases by one. Hence the move reduces the set of Weber points by half.

(3) If possible, assume that $(Q_d^{\mathcal{O}}, f')$ admits a partitive automorphism, i.e., a non-trivial translation τ. Assume that the translation, in the assigned coordinate system, is given by $x \mapsto x \oplus a$, for some $a \in \{0,1\}^d$. Let $\mathcal{R}$ and $\mathcal{R}'$ be the set of vertices occupied by robots in $(Q_d^{\mathcal{O}}, f)$ and $(Q_d^{\mathcal{O}}, f')$ respectively. Since τ maps any vertex of $\mathcal{R}'$ to another vertex of $\mathcal{R}'$, the group $<\tau>$ induces an equivalence relation on $\mathcal{R}'$, partitioning it into $2k$ disjoint sets of cardinality 2: $\mathcal{R}' = \bigcup\limits_{j=1}^{2k} \{x_j, \tau(x_j)\}$. Let $\mathcal{R}_{p_i}$ and $\mathcal{R'}_{p_i}$ be the multiset containing the p_ith terms of the assigned coordinates of vertices of $\mathcal{R}$ and $\mathcal{R}'$ respectively. Clearly $\mathcal{R}_{p_i}$ contains $2k$ number of 0's and $2k$ number of 1's. In $\mathcal{R}'_{p_i}$, we have $2k+1$ number of 0's and $2k-1$ number of 1's.

Case 1: Let the p_ith term of a be 0. Hence, if p_ith term of x is $b \in \{0,1\}$, then the p_ith term of $\tau(x) = x \oplus a$ is also b. This implies that $\mathcal{R'}_{p_i}$ has even number of 0's and 1's. This is a contradiction, as we have shown that number of 0's and 1's in $\mathcal{R'}_{p_i}$ is $2k+1$ and $2k-1$ respectively.
Case 2: Let the p_ith term of a be 1. So, if p_ith term of x is $b \in \{0,1\}$, then the p_ith term of $\tau(x) = x \oplus a$ is $\bar{b}$. This implies that $\mathcal{R'}_{p_i}$ has equal number of 0's and 1's. This is again a contradiction. □

Theorem 13. *Let* $(Q_d^{\mathcal{O}}, f)$ *be a configuration of* $4k$ $(k > 0)$ *robots with no multiplicities and no partitive automorphisms. Let* w_ℓ *be the leading Weber point, and hence the anchor. Suppose that the leader* r *is placed at a type 2 vertex* u*. If it moves towards* w_ℓ *to an empty vertex, then*

1. the new configuration $(Q_d^{\mathcal{O}}, f')$ *has no partitive automorphism*
2. $\mathcal{W}_f = \mathcal{W}_{f'}$
3. w_ℓ *is the leading Weber point of* $(Q_d^{\mathcal{O}}, f')$
4. r *is the leader in* $(Q_d^{\mathcal{O}}, f')$*.*

Proof. We use the same notations as in the proof of the previous theorem. For each $w \in \mathcal{W}_f \setminus \{w_\ell\}$ the following holds: (i) among the p_1th, $\ldots, p_m$th terms of the assigned coordinates, there is at least one 1, (ii) all the terms except the p_1th, $\ldots, p_m$th ones are 0. Exactly the opposite is true for the type 2 vertex u. It implies that the distance of r from w_ℓ is strictly less than its distance from any other Weber point in $\mathcal{W}_f) \setminus \{w_\ell\}$. Also, after the move, its distances from all Weber points reduce by exactly 1. Hence, after the move, $\zeta(w_\ell)$ remains lexicographically strictly largest among $\{\zeta(w) \mid w \in \mathcal{W}_{f'}\}$. All the statements of the theorem easily follow from these observations. □

Lemma 4. *Let $G = (V, E)$ be an arbitrary graph. Let $\mathcal{P} = v_0 e_0 v_1 e_1 v_2 \ldots v_{l-1} e_{l-1} v_l$ be a path from v_0 to v_l. Suppose that for any e_j, a move through it by a robot from v_j to v_{j+1} reduces its distance from v_l by 1. Then $\mathcal{P}$ is a shortest path from v_0 to v_l in G.*

Lemma 5. *Suppose that a robot moves from a vertex u to an adjacent vertex v in a hypercube Q_d. Then for any $w \in V(Q_d)$, either its distance from w reduces by 1 or increases by 1, i.e., its distance from w does not remain unchanged.*

Theorem 14. *Algorithm 1 achieves optimal gathering in ASYNC, for all asymmetric configurations of $4k$ $(k > 0)$ robots with weak multiplicity detection.*

Proof. By Theorems 12 and 13, no move in the first stage creates a partitive automorphism. Since at any time, only the leader is allowed to move, a multiplicity can only be created at the anchor. Since throughout stage 2, there is a unique multiplicity, a configuration with a partitive automorphism is never created. Notice that in both stages, an anchor is always a Weber point. Hence, the robots always move towards some Weber point. So, after each move, the centrality of the surviving set of Weber points is reduced by 1. Therefore, eventually the centrality of one Weber point becomes 0, which implies that gathering is accomplished.

It remains to prove that Algorithm 1 is optimal with respect to the total number of moves executed by the robots. Suppose that the algorithm gathers all the robots at w, which was a Weber point of the initial configuration. In view of Theorem 12, it is sufficient to show that every movement executed by any robot is towards w. Since every movement executed by a robot is towards some Weber point, according to Theorem 5, the set of Weber points of the configurations starting from the initial to the final configuration form the following nested series: $\mathcal{W}_0 \supseteq \mathcal{W}_1 \supseteq \ldots \supseteq \mathcal{W}_{final} = \{w\}$. In other words, w remains a Weber point throughout the progress of the algorithm. If at some step, a move by a robot is not towards w, then by Theorem 13, it moves away from w. Then the centrality of w is increased by 1, while the centrality of some other Weber point is decreases by 1. This means that w does not remain a Weber point after the move. This is a contradiction. □

4 Concluding Remarks

This is the first paper that investigates the gathering problem on a hypercube graph. We have provided a complete characterization of all gatherable configurations in ASYNC for $2k + 1$ and $4k$ $(k > 0)$ number of robots with weak multiplicity detection in an oriented hypercube. This leaves unsettled only the configurations with $4k + 2$ $(k > 0)$ number of robots. Note that our strategy for $4k$ robots does not work for $4k + 2$ robots. To see this, consider a configuration in Q_9 of 10 robots placed on the following vertices: 000000000, 110111000, 101111000, 011111000, 000111000, 001000111, 010000111, 100000111, 111000111, 111000000. Here, the anchor, i.e., the leading Weber point is 000000000 and the leader is 111000000. It can be seen that a move by the leader towards the anchor via any edge creates a configuration with a partitive automorphism. Another challenging direction of future research would be to study the problem with limited visibility. It would also be interesting to consider randomized algorithms to bypass the impossibility results.

Acknowledgements. The first three authors are supported by NBHM, DAE, Govt. of India, UGC, Govt. of India and CSIR, Govt. of India respectively. We would like to thank the anonymous reviewers for their valuable comments which helped us improve the quality and presentation of this paper.

Appendix A Proof of Theorem 7

Consider a configuration (Q_d, f) that has no partitive automorphism. Since we have assumed that the robots are positioned at distinct vertices, there is no distinction between the configuration and the perceived configuration. In other words, we have $\tilde{f} = f$. Assume that the configuration has no fixed Weber point. Let us assume that there is an algorithm $\mathcal{A}$ that deterministically solves LeadingWeberPoint. Let $w_1 \in \mathcal{W}$ be the leading Weber point elected by the robots. Since w_1 is not a fixed vertex, there is a $w_2 \neq w_1$ such that $\varphi(w_1) = w_2$, for some $\varphi \in Aut(G, f)$.

Each robot observes the positions of other robots in its local coordinate system. A local coordinate system of a robot is just an assignment $\Psi : V(Q_d) \longrightarrow \{0,1\}^d$, respecting the rule that $u, v \in V(Q_d)$ are adjacent if and only if $\Psi(u), \Psi(v)$ differ in precisely one bit. Since there is no global agreement, the local coordinate system of each robot is arbitrary, and is chosen by the adversary. Let us formally define the view of a robot. The view of a robot is given by the triplet $\mathcal{V}_\Psi = (\Psi, \tilde{f}, me)$, where $\Psi : V(Q_d) \longrightarrow \{0,1\}^d$ is the local coordinate system, $\tilde{f} : \{0,1\}^d \longrightarrow \{0,1\}$ is the multiplicity function defined on the set of vertices expressed in local coordinates, and $me \in \{0,1\}^d$ is the coordinates of the vertex on which the robot resides. The view $\mathcal{V}_\Psi$ is the input for algorithm $\mathcal{A}$. The output $\mathcal{A}(\mathcal{V}_\Psi) \in \{0,1\}^d$ is the coordinates of the required leading Weber point, i.e., the returned leading Weber point is the vertex $\Psi^{-1}(\mathcal{A}(\mathcal{V}_\Psi))$.

Consider a robot r_1 in the configuration residing at vertex v_1. The robot r_1, using a local coordinate system $\Psi_1 : V(Q_d) \longrightarrow \{0,1\}^d$, elects w_1 as the leading

Weber point. That is, given the input in the coordinate system Ψ_1, the output of $\mathcal{A}$ is $\Psi_1(w_1)$. Now consider the following cases.

Case 1: Suppose that $\varphi(v_1) = v_1$. Consider the local coordinate system $\Psi_2 = \Psi_1 \circ \varphi^{-1}$. Note that the view of r_1 in coordinate systems Ψ_1 and Ψ_2 are exactly the same, i.e., $\mathcal{V}_{\Psi_1} = \mathcal{V}_{\Psi_2}$. Since $\mathcal{A}$ is a deterministic algorithm, $\mathcal{A}(\mathcal{V}_{\Psi_1}) = \mathcal{A}(\mathcal{V}_{\Psi_2})$. Since the elected leading Weber point in local coordinate system Ψ_1 is w_1, we have $\mathcal{A}(\mathcal{V}_{\Psi_1}) = \Psi_1(w_1)$. So we have,

$$\begin{aligned} & \mathcal{A}(\mathcal{V}_{\Psi_2}) = \mathcal{A}(\mathcal{V}_{\Psi_1}) = \Psi_1(w_1) \\ \Rightarrow \quad & \Psi_2^{-1}(\mathcal{A}(\mathcal{V}_{\Psi_2})) = \Psi_2^{-1}(\Psi_1(w_1)) = \varphi \circ \Psi_1^{-1} \circ \Psi_1(w_1) = \varphi(w_1) = w_2 \end{aligned}$$

Hence, we see that in local coordinate system Ψ_1 the robot r_1 elects w_1 as the leading Weber point, while in Ψ_2 it elects w_2. This is a contradiction.

Case 2: Now assume that $\varphi(v_1) = v_2 \neq v_1$. Then there must be a robot r_2 in v_2. Suppose that the adversary sets the local coordinate system of r_2 as $\Psi_2 = \Psi_1 \circ \varphi^{-1}$. Then the view of r_1 and r_2 will be identical, i.e., $\mathcal{V}_{\Psi_1} = \mathcal{V}_{\Psi_2}$. Again we have, $\mathcal{A}(\mathcal{V}_{\Psi_2}) = \mathcal{A}(\mathcal{V}_{\Psi_1}) = \Psi_1(w_1)$ and hence, $\Psi_2^{-1}(\mathcal{A}(\mathcal{V}_{\Psi_2})) = w_2$. Therefore, r_2 will elect w_2, while r_1 elects w_1 as the leading Weber point. This is again a contradiction. □

References

1. Agathangelou, C., Georgiou, C., Mavronicolas, M.: A distributed algorithm for gathering many fat mobile robots in the plane. In: ACM Symposium on Principles of Distributed Computing, PODC 2013, Montreal, QC, Canada, 22–24 July 2013, pp. 250–259 (2013). https://doi.org/10.1145/2484239.2484266
2. Barrière, L., Flocchini, P., Fraigniaud, P., Santoro, N.: Rendezvous and election of mobile agents: impact of sense of direction. Theory Comput. Syst. **40**(2), 143–162 (2007). https://doi.org/10.1007/s00224-005-1223-5
3. Bose, K., Adhikary, R., Chaudhuri, S.G., Sau, B.: Crash tolerant gathering on grid by asynchronous oblivious robots. CoRR abs/1709.00877 (2017). http://arxiv.org/abs/1709.00877
4. Cieliebak, M., Flocchini, P., Prencipe, G., Santoro, N.: Solving the robots gathering problem. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 1181–1196. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45061-0_90
5. Cieliebak, M., Flocchini, P., Prencipe, G., Santoro, N.: Distributed computing by mobile robots: gathering. SIAM J. Comput. **41**(4), 829–879 (2012). https://doi.org/10.1137/100796534
6. Cohen, R., Peleg, D.: Convergence properties of the gravitational algorithm in asynchronous robot systems. SIAM J. Comput. **34**(6), 1516–1528 (2005). https://doi.org/10.1137/S0097539704446475
7. Czyzowicz, J., Gasieniec, L., Pelc, A.: Gathering few fat mobile robots in the plane. Theor. Comput. Sci. **410**(6–7), 481–499 (2009). https://doi.org/10.1016/j.tcs.2008.10.005

8. D'Angelo, G., Stefano, G.D., Klasing, R., Navarra, A.: Gathering of robots on anonymous grids and trees without multiplicity detection. Theor. Comput. Sci. **610**, 158–168 (2016). https://doi.org/10.1016/j.tcs.2014.06.045
9. D'Angelo, G., Di Stefano, G., Navarra, A.: Gathering of six robots on anonymous symmetric rings. In: Kosowski, A., Yamashita, M. (eds.) SIROCCO 2011. LNCS, vol. 6796, pp. 174–185. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22212-2_16
10. D'Angelo, G., Stefano, G.D., Navarra, A.: Gathering on rings under the look-compute-move model. Distrib. Comput. **27**(4), 255–285 (2014). https://doi.org/10.1007/s00446-014-0212-9
11. Das, S., Mihalák, M., Šrámek, R., Vicari, E., Widmayer, P.: Rendezvous of mobile agents when tokens fail anytime. In: Baker, T.P., Bui, A., Tixeuil, S. (eds.) OPODIS 2008. LNCS, vol. 5401, pp. 463–480. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-92221-6_29
12. Dessmark, A., Fraigniaud, P., Kowalski, D.R., Pelc, A.: Deterministic rendezvous in graphs. Algorithmica **46**(1), 69–96 (2006). https://doi.org/10.1007/s00453-006-0074-2
13. Dobrev, S., Flocchini, P., Prencipe, G., Santoro, N.: Multiple agents rendezvous in a ring in spite of a black hole. In: Papatriantafilou, M., Hunel, P. (eds.) OPODIS 2003. LNCS, vol. 3144, pp. 34–46. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27860-3_6
14. Flocchini, P., Kranakis, E., Krizanc, D., Luccio, F.L., Santoro, N., Sawchuk, C.: Mobile agents rendezvous when tokens fail. In: Královič, R., Sýkora, O. (eds.) SIROCCO 2004. LNCS, vol. 3104, pp. 161–172. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27796-5_15
15. Flocchini, P., Kranakis, E., Krizanc, D., Santoro, N., Sawchuk, C.: Multiple mobile agent rendezvous in a ring. In: Farach-Colton, M. (ed.) LATIN 2004. LNCS, vol. 2976, pp. 599–608. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24698-5_62
16. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of asynchronous robots with limited visibility. Theor. Comput. Sci. **337**(1–3), 147–168 (2005). https://doi.org/10.1016/j.tcs.2005.01.001
17. Godsil, C., Royle, G.F.: Algebraic Graph Theory. GTM, vol. 207. Springer Science & Business Media, New York (2001). https://doi.org/10.1007/978-1-4613-0163-9
18. Haba, K., Izumi, T., Katayama, Y., Inuzuka, N., Wada, K.: On gathering problem in a ring for 2n autonomous mobile robots. In: Proceedings of the 10th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS), poster (2008)
19. Izumi, T., Izumi, T., Kamei, S., Ooshita, F.: Mobile robots gathering algorithm with local weak multiplicity in rings. In: Patt-Shamir, B., Ekim, T. (eds.) SIROCCO 2010. LNCS, vol. 6058, pp. 101–113. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13284-1_9
20. Kamei, S., Lamani, A., Ooshita, F., Tixeuil, S.: Asynchronous mobile robot gathering from symmetric configurations without global multiplicity detection. In: Kosowski, A., Yamashita, M. (eds.) SIROCCO 2011. LNCS, vol. 6796, pp. 150–161. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22212-2_14
21. Kamei, S., Lamani, A., Ooshita, F., Tixeuil, S.: Gathering an even number of robots in an odd ring without global multiplicity detection. In: Rovan, B., Sassone, V., Widmayer, P. (eds.) MFCS 2012. LNCS, vol. 7464, pp. 542–553. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32589-2_48

22. Klasing, R., Kosowski, A., Navarra, A.: Taking advantage of symmetries: gathering of many asynchronous oblivious robots on a ring. Theor. Comput. Sci. **411**(34–36), 3235–3246 (2010). https://doi.org/10.1016/j.tcs.2010.05.020
23. Klasing, R., Markou, E., Pelc, A.: Gathering asynchronous oblivious mobile robots in a ring. Theor. Comput. Sci. **390**(1), 27–39 (2008). https://doi.org/10.1016/j.tcs.2007.09.032
24. Koreń, M.: Gathering small number of mobile asynchronous robots on ring. Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej. Technologie Informacyjne **18**, 325–331 (2010)
25. Kranakis, E., Krizanc, D., Markou, E.: Deterministic symmetric rendezvous with tokens in a synchronous torus. Discrete Appl. Math. **159**(9), 896–923 (2011). https://doi.org/10.1016/j.dam.2011.01.020
26. Kranakis, E., Santoro, N., Sawchuk, C., Krizanc, D.: Mobile agent rendezvous in a ring. In: 23rd International Conference on Distributed Computing Systems, ICDCS 2003, Providence, RI, USA, 19–22 May 2003, pp. 592–599 (2003). https://doi.org/10.1109/ICDCS.2003.1203510
27. Di Luna, G.A., Flocchini, P., Pagli, L., Prencipe, G., Santoro, N., Viglietta, G.: Gathering in dynamic rings. In: Das, S., Tixeuil, S. (eds.) SIROCCO 2017. LNCS, vol. 10641, pp. 339–355. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72050-0_20
28. Marco, G.D., Gargano, L., Kranakis, E., Krizanc, D., Pelc, A., Vaccaro, U.: Asynchronous deterministic rendezvous in graphs. Theor. Comput. Sci. **355**(3), 315–326 (2006). https://doi.org/10.1016/j.tcs.2005.12.016
29. Pagli, L., Prencipe, G., Viglietta, G.: Getting close without touching: near-gathering for autonomous mobile robots. Distrib. Comput. **28**(5), 333–349 (2015). https://doi.org/10.1007/s00446-015-0248-5
30. Prencipe, G.: Impossibility of gathering by a set of autonomous mobile robots. Theor. Comput. Sci. **384**(2–3), 222–231 (2007). https://doi.org/10.1016/j.tcs.2007.04.023
31. Stefano, G.D., Navarra, A.: Gathering of oblivious robots on infinite grids with minimum traveled distance. Inf. Comput. **254**, 377–391 (2017). https://doi.org/10.1016/j.ic.2016.09.004
32. Stefano, G.D., Navarra, A.: Optimal gathering of oblivious robots in anonymous graphs and its application on trees and rings. Distrib. Comput. **30**(2), 75–86 (2017). https://doi.org/10.1007/s00446-016-0278-7
33. Tel, G.: Network orientation. Int. J. Found. Comput. Sci. **5**(1), 23–57 (1994). https://doi.org/10.1142/S0129054194000037