Asynchronous Ring Gathering by Oblivious Robots with Limited Vision

Sayaka Kamei, Anissa Lamani and Fukuhito Ooshita

Abstract—We investigate in this paper the gathering problem where the aim is to ensure that a collection of identical, oblivious and asynchronous mobile robots meet (gather) in one location not known in advance. Robots operate in cycles that consist of three phases: Look, Compute and Move. During the first phase, robots take a snapshot to see the positions of the other robots. In the second phase, they make a decision to move or to stay idle. In the last phase, they move towards a computed destination if any. While most of the previous works in the discrete universe, under this setting consider that robots have an unlimited visibility i.e., each robot is able to see the position of all the other robots in the system, only few researches have been devoted recently to the case where robots have a restricted visibility. In our work, we continue the investigation aiming to characterize the cases in which the problem can be solved.

Index Terms—Mobile robots, Gathering, limited vision, Distributed algorithm, Ring

I. INTRODUCTION

We consider in this paper, robot systems in which a collection of weak robots need to collaborate in order to solve a common task. We investigate the gathering problem where the aim for the robots is to gather in one location not known in advance. Robots are endowed with visibility sensors, they are autonomous, oblivious, mobile and unable to communicate with each other. Robots operate in cycles that comprise three phases: Look, Compute and Move. During the first phase (Look), robots take a snapshot of the other robots positions using their visibility sensors. In the second phase (Compute), robots compute a neighboring destination according to the snapshot results. In the last phase (Move), they move towards the computed destination. We assume that the time between any two phases is finite but unbounded i.e., some robots might move according to an outdated snapshot while some others might be in different phases of their cycles.

Several studies have been devoted to the gathering problem under such a weak scenario. Two universes are usually considered i.e., the continuous two dimensional Euclidean space in which robots can move freely [1], [2], [3] and the discrete universe [4], [5], [6], [7], [8], [9], [10] in which the space is partitioned into a countable number of locations, conventionally represented by a graph, where nodes represent locations that can be sensed, and the edges represent the possibility for a robot to move from one location to another. In this paper, we consider the discrete universe, more precisely, we consider an anonymous and un-oriented ring.

Most of the previous works on the gathering problem in the discrete universe under the above scenario, assume that robots have an unlimited visibility, that is, they can sense the robots on every node of the system, whatever its size.

Only few recent researches have been investigating robots with limited visibility [11], [12]. In [11], bipartite graphs has been considered assuming that robots are only able to sense their direct neighboring nodes. The gathering is in this case, only possible when the initial configuration is a star (an agent a with all other agents adjacent to it). In [12], infinite lines has been studied, the authors prove that no universal algorithm solves the gathering problem if robots have a restricted vision (they cannot sense all the nodes) with the global weak multiplicity detection.

We continue in this paper the investigation on the gathering problem on ring shaped networks assuming that robots have a limited visibility. We aim to fully characterize the conditions under which the problem can be solved.

II. PRELIMINARIES

We consider in this paper the case of an anonymous, un-oriented and undirected ring of n nodes u_0, u_1, . . . , u_n such that u_i is connected to both u_{i−1} and u_{i+1}. On the ring, k robots operate in a distributed way in order to achieve a task in a cooperative fashion that is, to gather in one location not known in advance. The set of robots considered are identical i.e., they execute the same program using no local parameters and one cannot distinguish them using their appearance, they are also oblivious i.e., they cannot remember their last observations or their last steps taken before. In addition, they are unable to communicate directly with each other, however, they are endowed with a multiplicity sensor allowing them to sense the environment including the position of the other robots. Based on the sensing result, robot decides whether to move or to stay idle. Each robot executes cycles infinitely many times, (i) first, it takes a snapshot of the environment to see the position of some other robots (look phase), (ii) according to the observation, it decides to move or to stay idle (compute phase), (iii) if it decides to move, it moves to its neighbor node towards a target destination (move phase).

At each instant t, a subset of robots are activated by an entity known as the scheduler. This scheduler is considered to be fair i.e., all robots must be activated infinitely many times. We consider the asynchronous model that allows the interleaving of phases by the scheduler with the following constraint: the Move operation is instantaneous i.e., when a robot takes a snapshot of its environment, it sees the other robots on nodes and not on edges. However, since the scheduler is allowed to interleave the different operations, robots can move according to an outdated view since during the Compute phase, some robots may have already moved. At each instant t, robots occupy nodes of the ring, their positions form a configuration of the system at that time. We assume that, at instant t = 0 (i.e., at the initial configuration), there is at most one robot on each node. We say that there is a tower at some node u_i, if there is more than one robot on u_i. Similarly, we say that there is a single robot on u_i if there is exactly one robot on u_i. If the tower can be viewed by any robot at any node we say that robots have a global multiplicity detection. If the tower can only be perceive by only the robots that are part of it, we say that robots have a local multiplicity detection. If robots can
determine the exact number of robots part of the tower we say that their have a strong multiplicity detection, otherwise, we say that they have a weak multiplicity detection.

We assume that each robot can only sense the nodes being within a certain distance, denoted by $\phi$ ($\phi \geq 0$), from the node where the robot is located (The distance between two nodes in a ring is the number of edges in a shortest path connecting them). All robots have the same value of $\phi$. We use the same notation as in [13], since the ring is un-oriented, no robot is able to give an orientation to its snapshot. More precisely, given a robot $r_j$ located at a node $u_i$, the multiplicity sensor of $r_j$ outputs a sequence, $s_j$, of $2\phi + 1$ integers $x_{-\phi}, x_{-1}, \ldots, x_{-1}, x_0, x_1, \ldots, x_{\phi-1}, x_\phi$ such that:

- $x_{-\phi} = d_{i-\phi}$, $x_0 = d_i$, $x_\phi = d_{i+\phi}$, or

If the sequence $x_1, \ldots, x_{\phi-1}, x_\phi$ is equal to the sequence $x_{-1}, \ldots, x_{-1}, x_{-\phi}$, then the view of $r_j$ is said to be symmetric. Otherwise, it is said to be asymmetric.

In this paper, each rule in the proposed algorithms is presented in the following manner:

$\langle$ Label $\rangle \langle$ Guard $\rangle :: \langle$ Statement $\rangle$. The guard is a possible sequence $s$ provided by the sensor of a robot $r_j$:

$s = (x_{-\phi}, \ldots, x_{-1}, (x_i), x_{1}, \ldots, x_{\phi-1}, x_\phi)$. A robot $r_j$ at node $u_i$ is enabled at time $t$ (or simply enabled when it is clear from the context) if:

$s = (d_{i-\phi}(t), \ldots, d_{i-1}(t), (d_i), d_{i+1}(t), \ldots, d_{i+\phi}(t))$, or

$s = (d_{i+\phi}(t), \ldots, d_{i+1}(t), (d_i), d_{i-1}(t), \ldots, d_{i-\phi}(t))$. The corresponding rule $\langle$ Label $\rangle$ is then also said to be enabled.

The statement describes the action to be performed by $r_j$. There are only two possible actions: ($i$) $\rightarrow$, meaning that $r_j$ moves towards the node $u_{i+1}$, ($ii$) $\leftarrow$, meaning that $r_j$ moves towards the node $u_{i-1}$. When the view of $r_j$ is symmetric, the scheduler chooses the action to be performed. In this case, we write statement: $\leftarrow \vee \rightarrow$. In each algorithm, Character "?' indicates any value. We use $x^y$ to denote a sequence of $y$ consecutive nodes of multiplicity $x$. Observe that $y \leq \phi$ and $(x \times y) \leq n$. In a given configuration, for a given orientation of the ring, if the view of some robot $r_j$ is equal to $0^\phi(x)^\phi$, then $r_j$ is said to be a border. In the case where $x > 0$, we say that there is a tower at the border. A sequence of consecutive nodes in which each occupied node is at most at distance $\phi$ is called a $\phi$-group. In a given configuration $C$, we say that the visibility graph is connected (respectively disconnected) if there exists exactly one $\phi$-group (respectively more than one $\phi$-group) in $C$.

III. ODD NUMBER OF ROBOTS

We assume in the following an odd number of robots. We first show that, starting from an arbitrary initial configuration and assuming that $\phi = 1$, no deterministic algorithm solves the gathering problem if robots do not know their number $k$. We then provide an deterministic solution that solves the problem assuming that $k$ is known by all the robots and starting from any configuration that does not contain any tower and in which the following conditions hold: 1) visibility graph is connected, 2) there exists borders, 3) each robot uses the local strong multiplicity detection, 4) $\phi \geq 1$ and 5) $k \geq 3$.

A. Impossibility result

We show in the following that there exists no deterministic algorithm that solves the gathering problem if $k$ is unknown and $\phi = 1$. Let us first assume that $k < n$ and that the initial configuration contains a single $\phi$-group i.e., the visibility graph is connected. We borrow some results already proved in [13].

Since robots are only able to sense their direct neighbor, each robot can perform only four actions as follow:

(i) $R_{\text{sw}}(0)(1):: \leftarrow \vee \rightarrow$,

(ii) $R_{\text{out}}(0)(1):: \leftarrow$,

(iii) $R_{\text{in}}(0)(1):: \rightarrow$,

(iv) $R_{\text{swp}}(1)(1):: \leftarrow \vee \rightarrow$.

Assume that the visibility graph is connected, that is, any initial configuration contains a $\phi$-group of size $k$.

Lemma 1 Let $A$ be an asynchronous protocol for $\phi = 1$. If $A$ solves the gathering problem, then $A$ includes Rule $R_{\text{in}}$ that is the only applicable rule in the initial configuration.

Proof. Can be deduced from Lemmas (3.2), (3.3), (3.4), (3.5) and (3.6) in [13] (even though [13] considers the exploration problem, the lemmas also holds for the gathering problem). □

Lemma 1 ensures that at, at least, one border, one tower is created. Hence, for any $x \geq 1$, the following actions can be executed:

(i) $R_{\text{in}}(x)(0)(1):: \leftarrow$, (ii) $R_{\text{out}}(x)(0):: \rightarrow$,

(iii) $R_{\text{sw}}(x)(1)(1):: \leftarrow \vee \rightarrow$.

Let consider in the following that $k \geq 5$ (recall that $k$ is odd). From [13], we can deduce the following lemma:

Lemma 2 Let $A$ be an asynchronous protocol for $\phi = 1$ and $k \geq 5$. If $A$ solves the gathering problem then for every $x \geq 1$, $A$ includes $R_{\text{out}}(x)$ only.

Proof. Can be proved the same way as proved in [13] using Lemmas (3.6), (3.7) and (3.8) in [13]. □

Theorem 1 There exists no deterministic algorithm in the asynchronous model that solves the gathering problem if the visibility graph is connected, $k$ is unknown, $3 < k \leq n$ and $\phi = 1$.

Proof. Assume by contradiction, that there exists a deterministic algorithm that solves the gathering problem in the asynchronous model when $k$ is unknown and $\phi = 1$. Let us first assume that $k < n$, that is, the configuration contains exactly two borders. According to Lemma 1, the only rule that is enabled in this case is Rule $R_{\text{in}}$, that is the robot being at the border of the $\phi$-group moves towards its adjacent occupied node, creating a tower. Assume that the scheduler activates only one border (recall that the scheduler is asynchronous that is, it activates a non empty subset of robots for the execution). From Lemma 2, the tower that was created at the border is enabled to move towards its adjacent occupied node ($R_{\text{out}}(x)$ is the only rule that is enabled). Assume that the scheduler activates all robots in the same tower at the same time. Since they have the same view, robots exhibit the same behavior and thus, move to their neighboring occupied node. Rule $R_{\text{sw}}(x)$ keeps being enabled, that is, the size of the tower keeps increasing while moving to its adjacent occupied node. Since $k$ is unknown, robots do not know how many robots are part of the $\phi$-group, that is, the tower cannot stop moving after reaching a given size. Suppose that the scheduler keeps activating all robots part of the tower. Eventually, a configuration in which there is only one tower of size $k - 1$ and one neighboring single robot is reached. The tower being not able to recognize such a configuration, continues to be enabled to move towards its adjacent occupied node. Three cases are then possible depending on whether the single robot is enabled to move or not, as follows:

1) The single robot is not allowed to move. In this case assume that the scheduler activates at each time $k - 2$
robots in the tower. When the $k - 2$ robots move to their adjacent occupied node, the resulting configuration contains a single robot and a tower of size $k - 1$. Observe that the configuration reached is indistinguishable from the previous one. Assume that the scheduler keeps activating $k - 2$ robots in the tower at each time. Thus, no progression is made in the completion of the gathering task. Hence, the gathering is impossible in this case. Contradiction.

2) The single robot moves towards the tower. In this case, assume that the scheduler always activates all robots part of the tower at the same time along with the single robot. When the robots move, the tower and the single robot simply exchange their positions (this case is similar to the one where there are only two robots in the system). Thus, no progress is made in the completion of the gathering task. Hence, the gathering is impossible. Contradiction.

3) The single robot moves in the opposite direction of the tower. Assume that the scheduler activates all robots at the same time, that is all robots in the tower move towards the single robot and the single robot moves in the opposite direction of the tower. The configuration reached in this case is indistinguishable from the previous one since robots keep having the same view. That is, assuming that the scheduler activates all robots at the same time, the configuration reached is always indistinguishable from the previous one and hence, no progression is made in the completion of the gathering task. Contradiction.

Let consider now the case where $k = n$. All robots have in this case the same view. Assume that the scheduler activates them all at the same time at each time making them move in the same direction. The configuration reached is indistinguishable from the initial configuration. Thus, no progression is made in the completion of the gathering task. No gathering is thus possible. Contradiction.

We can deduce that there exists no deterministic algorithm in the asynchronous model that solves the gathering problem if $k$ is unknown and $\phi = 1$ and starting from an arbitrary towerless configuration in which the visibility graph is connected. Hence, the theorem holds.

Observe that the results can be easily extended to any towerless initial configuration in which the visibility graph is not connected, hence:

**Theorem 2** There exists no deterministic algorithm in the asynchronous model that solves the gathering problem starting from an arbitrary towerless initial configuration if $k$ is unknown, $3 < k \leq n$ and $\phi = 1$.

**B. Positive result**

We present in the following, a deterministic algorithm that solves the gathering problem when $k$ is odd and known by all the robots in the system and starting from an arbitrary towerless configuration in which the following conditions holds: 1) visibility graph is connected, 2) there exist borders, 3) each robot uses the local strong multiplicity detection, 4) $\phi \geq 1$ and 5) $k \geq 3$.

The idea of the algorithm is quite simple, first the border robot moves inside the visibility graph until it creates a tower. The border tower keeps then moving in the same direction, increasing its size at each time it moves to an occupied node until its size becomes equal to $(k + 1)/2$, robots part of such a tower are then not allowed to move anymore. Note that since the scheduler is asynchronous, the scheduler can activate one border and ignore the other one, however, when the size of the border tower is equal to $(k + 1)/2$, no robot is allowed to move except for the other border that was ignored, hence, we are sure to reach a configuration in which there are only two neighboring towers, one of size $(k + 1)/2$ and the other of size $(k - 1)/2$. Observe that the robots that are part of the tower of size $(k - 1)/2$ can only see that one single robot on their neighboring occupied node (recall that robots are endowed by the local strong multiplicity detection), that is, they keep being enabled to move to their adjacent occupied node. By moving they join the biggest tower. Hence, eventually, all robots are part of the same node. The algorithm contains only one rule: $(0^x(1)0^y?^{\phi - m - 1}) \land (0 \leq m < \phi) \land (1 \leq x < (k + 1)/2) \land (y \geq 1) \rightarrow \neg \alpha$.

**IV. EVEN NUMBER OF ROBOTS**

We will use the following proposition which can be proved similarly to [5].

**Proposition 1** No deterministic algorithm solves the gathering problem if the configuration is edge-edge symmetric.

First, we prove the necessary conditions to achieve gathering. Assume that robots $r_0, r_1, \ldots, r_{k-1}$ reside in this order. We denote $r_{k/2}$ and $r_{k/2+1}$ by middle robots. If there exists a hole between two middle robots, we call it the center hole. If two middle robots reside on neighboring nodes, we assume there exists the central hole of size 0 between them. Then, we have the following lemma. In the proof of the lemma, we denote the reverse of sequences by $R$, that is, $(x_0x_1 \ldots x_{\ell-1})^R = x_{\ell-1}x_{\ell-2} \ldots x_0$ holds.

**Lemma 3** No deterministic algorithm solves the gathering problem if $k$ is even and the size of the central hole is even.

**Proof.** Let us consider an algorithm $A$. We show that, if we allow even-size central hole, there exists a non-symmetric and non-periodic configuration such that $A$ cannot solve gathering. Without loss of generality, we can assume that $A$ includes the rule $(o(0(1)0\beta) \rightarrow$, where $\alpha \in \{0,1\}^{\phi - 1}$ and $\beta \in \{0,1\}^\phi - 1$. Then, if configuration $C$ includes $o\alpha 0\beta$ and the middle robot is activated, the part of configuration changes to $o\alpha 001\beta$. Let us consider configuration $o\alpha 010\beta 100\alpha^{R0}\beta^{R0}$, where $m < \phi$ is even and $\ell > \phi$ is even. Note that this configuration is non-periodic and non-symmetric. Clearly, this configuration transits to $o\alpha 010\beta 100\alpha^{R0}\beta^{R0}$, Since this configuration is edge-edge symmetric, $A$ cannot solve gathering.

We then present a deterministic solution under the following conditions: 1) The visibility graph is connected, 2) there exist borders, 3) each robot uses the global strong multiplicity detection, 4) the size of the central hole is odd and at most $\phi - 2$, 5) $\phi \geq 3$, 6) $k \geq 4$, and 7) each robot knows $k$.

The basic idea of the solution is similar to the one used when the number of robots is odd: each border robot moves inside the $\phi$-group, and if two towers remain, the smaller tower joins the larger one. However, whereas the case when $k$ is odd, we need to avoid creating two towers of size $k/2$ because such configurations never achieve gathering (each tower can behave as a single robot that is, the configuration is similar to the one that contains only two robots). To prevent this to happen, we create two towers of size $(k/2 - 1)$ in both sides of the middle robots (robots that are neighbor of the central hole). When the middle robots confirm that two such towers have been created, they create a tower of size 2 between the two border towers,
we refer in the sequel to this tower by the middle tower. After that, the two border towers of size \((k/2 - 1)\) move to join the middle tower. When all robots of a given border towers join the middle tower, the size of the middle tower becomes equal to at least \((k/2 + 1)\). Therefore, the smaller tower can join it, and achieves gathering.

**Algorithm 1** Deterministic Even Gathering

1) \((0^m(x)0^n x' y^{\phi - m - 1}) \land (x < (k - 2)/2) \land (x' > 0) \land (m \geq 0) :\rightarrow \)

2) \((0^m(x)0^n x' y^{\phi - m - 1}) \land (x = (k - 2)/2) \land (m > 0) :\rightarrow \)

3) \((0^m(x)0^n x' y^{\phi - m - 1}) \land (x = (k - 2)/2) \land (i \geq 0) \land (j \geq 0) \land (l \geq 0) \land (p \geq 0) \land (i + j = \phi - 1) \land (i + l \leq \phi - 2) \land (j + l \leq \phi - 2) \land (j + l + m \text{ is even}) \land (j > m) :\rightarrow \)

4) \((0^m(x)0^n x' y^{\phi - m - 1}) \land (x = (k - 2)/2) \land (i \geq 0) \land (j \geq 0) \land (l \geq 0) \land (m \geq 0) \land (p \geq 0) \land (i + j = \phi - 1) \land (i + l \leq \phi - 2) \land (j + l \leq \phi - 2) \land (j + l + m \text{ is odd}) \land (j < (j + l + m + 1)/2) :\rightarrow \)

5) \((0^m(x)0^n x' y^{\phi - m - 1}) \land (x = (k - 2)/2) \land (m > 0) :\rightarrow \)

6) \((0^m(x)0^n x' y^{\phi - m - 1}) \land (x \geq 0) \land (x' > 0) \land (x'' > 0) \land (x + x' + x'' = k) :\rightarrow \)

7) \((0^m(x)0^n x' y^{\phi - m - 1}) \land (x + x' = k) \land (x' > k/2) :\rightarrow \)

The formal description of the algorithm is given in Algorithm 1. First, border robots move inside the visibility graph to create a tower (Rule 1). The border robots keep moving in the same direction and keep increasing their size at each time they meet with another robot. When the size of the tower is equal to \((k - 2)/2\), if the robots in such a tower can only see a single robot, the tower keeps moving towards the visible single robot (Rules 1 and 2). When the robots in the tower can see two single robots then the tower stops moving. Note that the configuration contains exactly two single robots and they are middle robots. When the border towers stop moving, the two single robots can see both towers. After that, the two single robots are now allowed to move to create a tower. To do so two cases are possible: (i) the number of nodes between the two towers is even (Rule 3), and (ii) the number of nodes between the two towers is odd (Rule 4). For both rules, two single robots are border robots and the hole between them is the central hole, and consequently \(l\) is odd and \(l \leq \phi - 2\) holds. The condition \(j + l \leq \phi - 2\) in Rules 3 and 4 implies that middle robots move after each tower sees two middle robots. Consequently, since each tower sees two single robots, towers do not move until middle robots create a tower of size 2 by Rules 3 or 4. In case (i), since \(j + m\) is odd, \(j\) and \(m\) are different. This means we can select one single robot such that \(j > m\) and move it to another single robot (Rule 3). Eventually they create a tower of size two. In case (ii), since \(j + l + m\) is odd, we can select one central node between two towers. Then, two single robots move to the central node and eventually create a tower of size two on the central node (Rule 4).

After robots in the border towers see a tower of size two, they move toward the tower (Rule 1 and 5) and eventually all robots gather within three nodes. Then border robots join the middle node (Rule 6). From asynchronous execution, it is possible that all robots occupy two nodes. However, in such a case, since two robots neighboring to the central hole create a tower in Rule 3 or 4, the number of robots is different between the two nodes. Therefore, robots in the node with smaller number of robots join the other node (Rule 7) and achieve gathering.

**V. Conclusion**

We have considered in this paper the gathering problem on ring shaped networks by oblivious, asynchronous robots that have a limited visibility i.e., robots cannot sense the nodes beyond a certain distance \(\phi\). This is a preliminary work to get a full characterization under which the problem can be solved. We conjecture that the assumptions used in this paper are necessary in order to achieve the gathering.

**References**


