

# Rendezvous on a Known Dynamic Point on a Finite Unoriented Grid

Pritam Goswami<sup>[0000-0002-0546-3894<sup>\*</sup>]</sup>, Avisek Sharma<sup>[0000-0001-8940-392X]</sup>, Satakshi Ghosh<sup>[0000-0003-1747-4037]</sup>, and Buddhadeb Sau<sup>[0000-0001-7008-6135]</sup>

Jadavpur University, 188, Raja S.C. Mallick Rd, Kolkata 700032, India  
{pritamgoswami.math.rs, aviseks.math.rs, satakshighosh.math.rs, buddhadeb.sau}@jadavpuruniversity.in

**Abstract.** In this paper, we have considered two fully synchronous *OBLOT* robots having no agreement on coordinates entering a finite unoriented grid through a door vertex at a corner, one by one. There is a resource that can move around the grid synchronously with the robots until it gets co-located along with at least one robot. Assuming the robots can see and identify the resource, we consider the problem where the robots must meet at the location of this dynamic resource within finite rounds. We name this problem "Rendezvous on a Known Dynamic Point".

Here, we have provided an algorithm for the two robots to gather at the location of the dynamic resource. We have also provided a lower bound on time for this problem and showed that with certain assumption on the waiting time of the resource on a single vertex, the algorithm provided is time optimal. We have also shown that it is impossible to solve this problem if the scheduler considered is semi-synchronous.

**Keywords:** Rendezvous · Finite Grid · Dynamic Resource.

## 1 Introduction

A swarm of robots is a collection of inexpensive and simple robots that can do a task collaboratively by executing one single distributed algorithm. In recent days swarm robot algorithm has become an exciting topic for research for several different reasons. Firstly, from the economic perspective, it is in general cheaper than using powerful robots. Moreover, a swarm of robots can be easily scaled based on the size of the environment they are deployed. Also, a swarm of robots is more robust against different faults (eg. crash faults and byzantine faults). There are many other positive sides to using a swarm of robots for executing a task. Thus, this topic has become quite relevant in the field of research and application. The application of swarm robots is huge. For example, it can be used for patrolling, different military operations, rescue operations, cleaning large surfaces, disaster management, network maintenance and there are several others.

---

\* The first three authors are full time research scholars in Jadavpur University.

### 1.1 Background and Motivation

There are several tasks a swarm of robots can do like, arbitrary pattern formation ([1]), gathering ([3]), network exploration ([16]), dispersion ([6]) and many more. Here, we are interested in the problem of gathering. Gathering is a very classical problem where a collection of robots deployed in an environment meets at a single point within a finite time. This problem has been solved under different environments and different settings ([3,4,5,10,13,14,18]). Rendezvous is a special case of gathering where the number of robots that need to gather is exactly two ([7,9,11,12,19]).

Since the deployed robots are simple it is hard for them to exchange important information being far apart. So the main motivation for gathering is to meet at a single point where the robots can exchange information for doing some task. Now let the information is stored at a single point or a set of points in the environment. And the robots need to be on those specific points to exchange information. In that case, the robots must gather at one of those specific points for exchanging information.

Now, let there be one single point of resource in the environment and the resource itself is a robot doing some other task in the same environment and thus, can move freely until it meets with another robot. So, the question is can two robots gather at the location of this moving resource? this is the question that has been the motivation behind this paper. Now, it is quite obvious that the environment should be a bounded region otherwise it would be impossible to do. Also for a bounded region in a plane, finite point robots can't meet at the location of the resource as there are infinitely many empty points where the resource can move to avoid the meeting. Thus it is natural to consider this problem for a bounded network. For that reason, we have considered a finite grid as the environment in this work. Note that if two robots with weak multiplicity detection can gather at the location of the resource, in some bounded networks, then any number of robots can gather. This is because after two robots meet with the resource, the resource becomes still and the other robots simply move to the location of the resource. That is why we have considered this problem with two robots only, rather than using any number of robots.

### 1.2 Earlier Works

In this paper, we are focusing on the problem of rendezvous on a known dynamic vertex. Rendezvous is a special case of gathering involving two robots. Gathering has been studied under different environments and different models throughout the span of research on swarm robot algorithms. In [3], authors have shown that gathering on a plane is possible for fully synchronous *OBLLOT* robots but in [18] it has been proved that for semi-synchronous and asynchronous *OBLLOT* robots it is impossible to gather without any axis agreement and multiplicity detection capabilities. So considering multiplicity detection only a solution has been provided in [2] under the asynchronous scheduler. Gathering has been studied under different networks also ([4,5,13,14]). In [14] Klasing et al. first proposed the problem on a ring and proved that gathering on a ring is impossible without the robots having multiplicity detection capabilities. In [4], the authors examined the problem on the grid and trees and they found out that gathering is impossible even with global multiplicity detection if the configuration is periodic or symmetric and the line of symmetry is passing through any of the grid lines. Considering

limitations in the view of robots many works have been done recently in [8,10,15,17]. Among these, the work in [17] and [10] considered infinite rectangular and triangular grids respectively.

Now Rendezvous is a special case of gathering which has been studied extensively in [7,9,11,12,19]. In [19], Suzuki et al. have shown that two *OBLOT* robots can't gather in a semi-synchronous setting if the robots do not have any agreement on their local coordinate system even with multiplicity detection. So in [7,9,11] authors have solved the problem considering robots with  $O(1)$  memory or  $O(1)$  bits of message communication under an asynchronous scheduler.

### 1.3 Our Contribution

Till now all work in gathering considered the meeting point to be not known from earlier. contrary to that, in this work we have assumed, two fully synchronous robots entering a finite unoriented grid through a door at a corner of the grid, know the meeting point (i.e., can see and identify the resource). But the problem is, the meeting point (i.e the location of the resource) can also move to an adjacent vertex along with the robots in a particular round.

Assuming the robots to be *OBLOT* with the capability of global weak multiplicity detection we have provided a distributed algorithm that solves the rendezvous problem on a known dynamic meeting point within  $O(T_f \times (m+n))$  rounds, where  $T_f$  is the upper bound of the number of consecutive rounds the meeting point i.e., the resource can stay at a single vertex alone and  $m \times n$  is the dimension of the grid. We have also shown that for solving rendezvous on a known dynamic point on a finite grid of dimension  $m \times n$  at least  $\Omega(m+n)$  epochs is necessary. Hence, if we assume that the maximum number of consecutive rounds, the location of the resource can stay the same is  $O(1)$  then, the algorithm provided in this paper is time optimal. We have also proved that solving rendezvous on a known dynamic point on a finite grid is impossible if the scheduler considered is semi-synchronous. This justifies why a fully synchronous scheduler has been considered in this work.

### 1.4 Organization of the Paper

In section 2, we have defined the problem formally and discussed the models of the robot, resource, and scheduler in detail. We also have some definitions and notations in this section which will be needed for the contents in Section 4. In Section 3, we have discussed the lower bound of time required to solve this problem and also proved an impossibility result about solving this problem under semi synchronous scheduler. In Section 4, we have described each phase of the algorithm with the correctness results mentioned in different theorems and lemmas. Finally, in Section 5, we conclude the paper with some future possibilities and pathways for this research to continue.

## 2 Problem Definition and Model

### 2.1 Problem Definition

Let  $G$  be a finite rectangular grid of dimension  $m \times n$ . Suppose there is a doorway in a corner of the grid through which two synchronous robots  $r_1$  and  $r_2$  can enter the grid. Consider a movable resource that is placed arbitrarily on a vertex of  $G$ . Both robots can see the resource. The resource will become fixed if at least one of  $r_1$  or  $r_2$  is on the same vertex with the resource. Now the problem is to design a distributed algorithm such that after finite execution of which both the robots gather at the vertex of the resource.

### 2.2 Model

Let  $G = (V, E)$  be a finite unoriented grid. A corner vertex is a vertex of degree two.  $G$  has exactly one corner vertex that has a door called door vertex. Robots can enter the grid by entering through that door. There is a movable resource, initially placed arbitrarily at a vertex  $g_0$  ( $g_0$  is not the door) of  $G$ .

**Robot Model:** The robots are considered to be

- **Autonomous:** There is no centralized control.
- **Anonymous:** The robots do not have any unique identifiers for distinction.
- **Homogeneous:** All robots run the same distributed algorithm.
- **identical:** The robots are physically indistinguishable.

Also, the robots are considered to be point *OBLLOT* robots (i.e., robots with no persistent memory). The robots can enter through the door one by one. A robot can distinguish if a vertex is on the boundary or a corner of the grid. Also, a robot can identify the door from any other vertex. A robot can distinguish the resource from other robots. Each robot has its local coordinate system but they do not agree on any global coordinate system.

The robots operate in a *LOOK-COMPUTE-MOVE* (LCM) cycle. In each of the cycles, a robot that was previously idle wakes and does the following phases,

**LOOK:** In *LOOK* phase a robot takes a snapshot of its surroundings and gets the location of other robots and the resource according to its local coordinate system.

**COMPUTE:** In this phase a robot performs an algorithm with the locations of resource and other robots as input and as an output of that algorithm it gets the location of a neighboring vertex called the destination point.

**MOVE:** In *MOVE* phase a robot moves to the destination point through the edge of  $G$  joining its current location and destination vertex. It is assumed that no two robots can cross each other through one edge without collision.

After completion of *MOVE* phase, the robot becomes idle until it is activated again.

The activation of the robots is controlled by an entity called a scheduler. In the literature, there are mainly three types of schedulers. In the following, we discuss all the scheduler models and the scheduler we have chosen among them for solving this problem.

**Scheduler Model:** There are mainly three types of schedulers that have been considered throughout the literature of swarm robotics. The models are as follows:

**Fully Synchronous Scheduler (FSYNC)**

- ◆ Time is divided into rounds of equal lengths
- ◆ At the beginning of each round all robots are activated.
- ◆ In a particular round all activated robots perform the *LOOK*, *COMPUTE* and *MOVE* phases together.

**Semi Synchronous Scheduler (SSYNC)**

- ◆ Time is divided into rounds of equal lengths
- ◆ At the beginning of each round a subset of robots are activated.
- ◆ In a particular round all activated robots perform the *LOOK*, *COMPUTE* and *MOVE* phases together.

**Asynchronous Scheduler (ASYNC)**

- ◆ There is no sense of rounds.
- ◆ A robot can either be idle or in any of the *LOOK*, *COMPUTE*, or *MOVE* phases while some other robots are activated.

In this work, we have shown that it is impossible to solve the problem of rendezvous on a known dynamic point if the scheduler is semi-synchronous. Hence considering a fully synchronous scheduler we have provided an algorithm *DYNAMIC RENDEZVOUS* that solves the problem within finite rounds.

**Resource Model:** The resource *res* is a movable entity, initially which is placed arbitrarily on a vertex (except the door) of  $G$ . The resource moves synchronously along with the robots. let the position of *res* at round  $i$  is denoted as  $g_i$  ( $g_0$  is the initial location). for some round  $i$ ,  $g_i$  and  $g_{i+1}$  are at most 1-hop away. The movement of the resource *res* is controlled by an adversary. So  $g_{i+1}$  can be any neighbor of  $g_i$ . We assume that resource will stay fixed if it meets with at least a robot among  $r_1$  and  $r_2$ . Otherwise, it can not stay fixed on a vertex forever. Let  $T_f$  be the upper bound of the number of rounds that *res* can stay fixed alone on a vertex of  $G$ . Also, it is assumed that the resource can not cross a robot on an edge without collision.

### 2.3 Notation and Definitions

For a robot  $r$  we denote the resource as *res* and the other robot as  $r'$ . Now we have the following definitions.

**Definition 1 (Door boundary of a robot).** *If a robot  $r$  is located on a boundary of the grid on which the door vertex is also located then that boundary is called the door boundary of the robot  $r$  and is denoted as  $BD(r)$ .*

**Definition 2 (Perpendicular Line of robot  $r$ ).** *For a robot  $r$  on a boundary, the straight line perpendicular to  $BD(r)$  passing through  $r$  is called the perpendicular line of robot  $r$ . It is denoted as  $PD(r)$ .*

**Definition 3 (Distance from resource along  $BD(r)$ ).** Distance of the resource  $res$  along boundary  $BD(r)$  is defined as the hop distance of robot  $r$  from the vertex  $v$  on  $BD(r)$  such that the line joining  $v$  and  $res$  is perpendicular to  $BD(r)$ . We denote this distance as  $dist(r)$  for a robot  $r$  on  $BD(r)$ .

**Definition 4 (InitGather Configuration).** A configuration  $C$  is called a INITGATHER CONFIGURATION if:

1. there is a robot  $r$  such that  $r$  and the resource  $res$  are on a grid line (say  $L$ ).
2. the perpendicular distance of the other robot  $r'$  to the line passing through  $res$  and perpendicular to  $L$  is at most one.

In the following Fig. 1 and Fig. 2 we have mentioned the entities we have defined above.

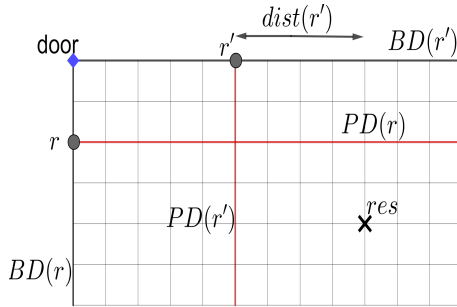


Fig. 1: Diagram of a configuration mentioning  $BD(r)$ ,  $BD(r')$ ,  $PD(r)$ ,  $PD(r')$  and  $dist(r')$ .

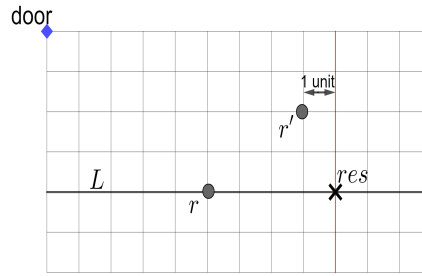


Fig. 2: Diagram of an INITGATHER CONFIGURATION

### 3 Lower Bound of Time and Impossibility

In this section, we will discuss the lower bound of time required to solve the problem of rendezvous on a known dynamic point on a finite grid of dimension  $m \times n$ . Also, we will prove an impossibility result which will justify our assumption of considering a fully synchronous scheduler to solve this problem. But first, let us define “epoch”. An epoch is a time interval within which each robot in the system has been activated at least once. In the case of a fully synchronous scheduler, an epoch is equivalent to a round but for other schedulers, an epoch interval is finite but unpredictable. Now in the following theorem, we will discuss the time lower bound of solving rendezvous at a known dynamic point on a finite grid.

**Theorem 1.** Any algorithm that solves rendezvous at a known dynamic point on a finite grid of dimension  $m \times n$  takes  $\Omega(m + n)$  epochs in the worst case.

*Proof.* Let us consider the scheduler to be a fully synchronous scheduler. Thus an epoch is equivalent to a round. Consider the following diagram (Fig. 3) where after each  $T_f$  consecutive rounds, the resource changes its location from either  $P$  to  $Q$  or  $Q$  to  $P$ . This implies after entering from the door vertex the robots must meet the resource either in vertex  $P$  or in vertex  $Q$ . Now from the door vertex, the shortest path to  $P$  or  $Q$  is of length  $m + n - 1$ . So to meet at either  $P$  or  $Q$  with the resource, each robot must travel through a path of length at least  $m + n - 1$ . Now since in a round a robot can only move a path of length one, to travel a path of length  $m + n - 1$  at least  $m + n - 1$  round i.e epoch is necessary to solve this problem. Hence the result.  $\square$

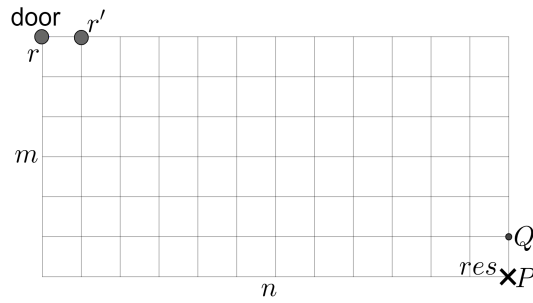


Fig. 3: from the door vertex to reach  $P$  or  $Q$  the robot  $r$  needs to travel at least a path of length  $m + n - 1$ .

Now we will discuss the impossibility result in the next theorem.

**Theorem 2.** *No algorithm can solve the problem of rendezvous on a known dynamic point on a finite grid of dimension  $m \times n$  if the scheduler is semi-synchronous.*

*Proof.* Let there is an algorithm  $\mathcal{A}$  such that after finite execution of which two robots on a finite grid of dimension  $m \times n$  meet at the location of the dynamic resource. Let  $m, n > 2$ . Also, let  $t$  be the round such that after completion of which at least one robot reaches the location of the resource and terminates.

Let no robot is adjacent to the resource at the beginning of round  $t$ . This implies at the beginning of round  $t$ , the resource has at least two empty neighbor vertices. Now let the adversary activates only one robot during this round. Thus, even if the activated robot moves to one of the resource's empty adjacent vertex, another empty vertex remains empty. So even if the resource has to move during round  $t$  it can always find an empty vertex to move that remains empty after the completion of the round. Hence after completion of round  $t$ , no robot can move to the location of the resource. Thus we reach a contradiction. Now, let exactly one robot is adjacent to the resource  $res$  at the beginning of round  $t$ . Then, at least  $res$  has one empty vertex which is not reachable by the adjacent robot in one round. So, if the adversary activates only the adjacent robot, say  $r$ , and  $res$  moves to the empty vertex not reachable by  $r$  then again we reach a contradiction. Hence both the robots must be adjacent to the resource at the beginning

of round  $t$ . Now if the resource is not at the corner and both the robots are adjacent to the resource at the beginning of round  $t$  then, the resource must have an empty adjacent vertex that is not reachable by the robots in one round. Thus if the resource moves to that vertex during round  $t$ , we again reach a contradiction.

Now if we can prove that the configuration (say,  $C_{corner}$ ) where the resource is at a corner and both the robots are adjacent to it, is never formed then we are done. Let the adversary always activates only one robot in a particular round. Now, if possible let at the beginning of round  $t$ , the configuration is  $C_{corner}$ . This implies the configuration, say,  $C_{corner-1}$ , that was formed just before  $C_{corner}$  must be one of  $C_1, C_2, C_3$  or  $C_4$  (Fig.4). Since adversary is compelled to activate only one robot in a particular round, so in configuration  $C_{corner-1}$  one the robot must be adjacent to the corner vertex. Without loss of generality let  $r'$  be that robot. Note that, in  $C_1$  and  $C_2$   $res$  did not move to form  $C_{corner}$  and, in  $C_3$  and  $C_4$   $res$  had to move to form  $C_{corner}$ . Note that in all of these configurations, there is only one robot that is adjacent to the resource. If the adversary activates the adjacent robot then, in all of these configurations the resource can find an empty adjacent vertex that is not a corner and remains empty even after the move of the resource. Thus from any of the four configurations  $C_1, C_2, C_3$  and  $C_4$ ,  $C_{corner}$  is not formed. Hence we arrive at a contradiction. Thus  $C_{corner}$  will never be formed and hence the result.  $\square$

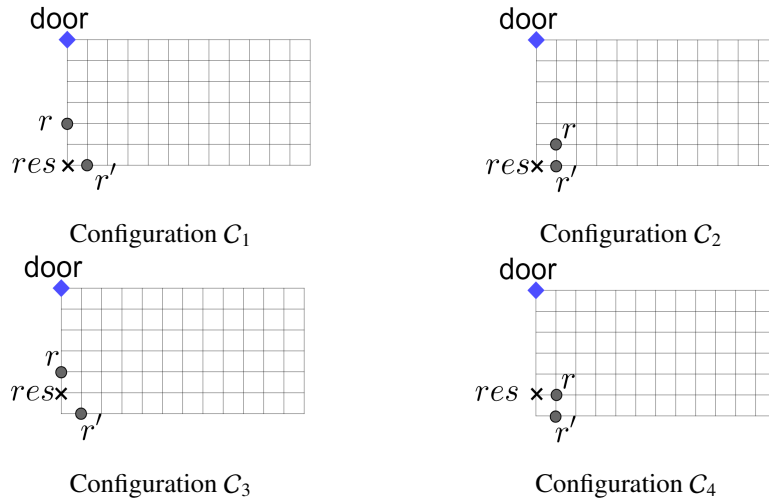


Fig. 4: Some examples of configuration  $C_{corner-1}$ .

This justifies the necessity of a fully synchronous scheduler to solve this problem. In the next section assuming a fully synchronous scheduler, we have provided an algorithm that solves this problem of rendezvous on a known dynamic point on a finite grid.



## 4 Algorithm

It is quite obvious to observe that without the help of the other robot, a robot can not independently reach the location of the resource if the resource is controlled by an adversary. So to solve this problem the two robots must work together collaboratively and push the resource toward a corner. Now since there is no agreement on the coordinates of the robots and the robots are oblivious, the main challenge here is to agree on the direction for the robots to move.

The rendezvous algorithm `DYNAMIC RENDEZVOUS`, proposed in this section is executed in three phases. `ENTRY PHASE`, `BOUNDARY PHASE` and `GATHER PHASE`. In the first two phases, the agreement on the direction of movement for the robots is constructed from the fact that the robots know the location of the resource, door vertex, and the boundaries of the grid and they always remain on the same boundary during these two phases. In the `GATHER PHASE` though, the robots move inside the grid leaving its boundary. In this situation as the robots are not on boundaries, they can not decide on a specific boundary for agreement. In this scenario, the agreement on the direction comes from the fact that according to the algorithm, at least one robot must be on a line along with the resource during each round of this phase. The algorithm `DYNAMIC RENDEZVOUS` is as follows.

---

### Algorithm 1: Dynamic Rendezvous

---

```

1 if a robot is on the door vertex then
2   | begin ENTRY PHASE;
3 else if The configuration is not an INITGATHER CONFIGURATION then
4   | begin BOUNDARY PHASE;
5 else
6   | begin GATHER PHASE;
```

---

The three phases are described in more detail in the following subsections.

#### 4.1 ENTRY PHASE

The first phase is called the `ENTRY PHASE`. During the `ENTRY PHASE`, both the robots enter through the door vertex one by one into the grid  $G$ . A robot on the door vertex first checks if it can see another robot already on the grid. If it does not find any other robot on the grid, it moves through any of the two edges that are incident on the door vertex. On the other hand, if there is already a robot on an adjacent vertex of the door vertex, the robot on the door vertex moves through the other edge which is not incident on the adjacent vertex where it saw another robot. A robot on the adjacent vertex of the door

vertex does not move until it sees another robot on the other adjacent vertex of the door vertex.

---

**Algorithm 2:** Entry Phase for robot  $r$

---

```

1 if  $r$  is on the door vertex then
2   if no other robot on boundary then
3     | move through any edge on the boundary;
4   else
5     | move through the edge on the boundary where there is no other robot;

```

---

The ENTRY PHASE ends when both robots are at the two distinct adjacent vertices of the door vertex. After the ENTRY PHASE the robots will check if the configuration is a INITGATHER CONFIGUATION or not. If the configuration is not an INITGATHER CONFIGUATION then the robots execute the BOUNDARY PHASE otherwise they execute the GATHER PHASE

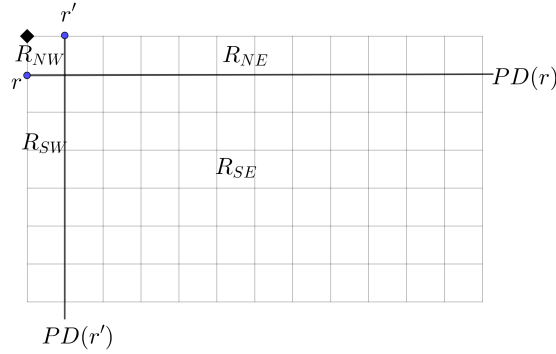
## 4.2 BOUNDARY PHASE

The BOUNDARY PHASE starts after the end of the ENTRY PHASE. In this configuration, both the robots are at the boundary and on the two distinct adjacent vertices of the door vertex initially. If a robot  $r$  sees that, the distance from the resource  $res$  for the robots  $r$  and  $r'$  along  $BD(r)$  and  $BD(r')$  respectively both are non zero then  $r$  finds out the vertex  $v$  among its two adjacent vertices for which the distance of  $res$  along  $BD(r)$  decreases for the current view. If  $v$  is not the door,  $r$  moves to that vertex  $v$ .

On the other hand if  $r$  sees that, the distance of the resource for the other robot  $r'$  along  $BD(r')$  is zero and its distance from  $res$  along  $BD(r)$  is strictly greater than one then it moves along  $BD(r)$  towards the resource  $res$ . Note that for this case a robot never moves to the door vertex as if  $r$  moves to the door vertex by executing this case during some round  $t$  then,  $res$  must be on  $BD(r')$  at the beginning of round  $t$ . Also at the beginning of round  $t$ ,  $r$  must be one hop away from the door vertex. This implies  $dist(r) = 1$ . This leads to contradiction as this case is only executed by  $r$  when  $dist(r) > 1$ .

**Definition 5 (Quadrants).** The grid  $G$  is divided into four quadrants by the two lines  $PD(r)$  and  $PD(r')$ . The quadrants on the northeast, northwest, southeast, and southwest are denoted as  $R_{NE}$ ,  $R_{NW}$ ,  $R_{SE}$  and  $R_{SW}$  respectively (Fig. 5).

At the beginning of the BOUNDARY PHASE, quadrant  $R_{NW}$  is a  $2 \times 2$  grid,  $R_{SW}$  is a  $(m-1) \times 2$  grid,  $R_{NE}$  is a  $2 \times (n-1)$  grid and  $R_{SE}$  is a  $(m-1) \times (n-1)$  grid. At the beginning of BOUNDARY PHASE, the resource  $res$  must be either inside or on one of  $R_{NE}$ ,  $R_{SW}$  and  $R_{SE}$ .

Fig. 5: Four quadrants divided by  $PD(r)$  and  $PD(r')$ **Algorithm 3:** Boundary Phase for robot  $r$ 


---

```

1 if on the same vertex with  $res$  then
2   | terminate;
3 else
4   | if  $r'$  is on the same vertex with  $res$  then
5   |   | move to  $res$  along any shortest path avoiding door;
6   | else
7   |   | if  $dist(r) \neq 0$  and  $dist(r') \neq 0$  then
8   |   |   |  $v \leftarrow$  adjacent vertex on  $BD(r)$  which is near  $res$  along the
9   |   |   |   | boundary.;
10  |   |   | if  $v$  is not the door vertex then
11  |   |   |   | | move to  $v$ ;
12  |   |   | else
13  |   |   |   | | if  $dist(r') = 0$  and  $dist(r) > 1$  then
14  |   |   |   |   | Move along boundary towards  $res$ ;

```

---

**Theorem 3.** For a grid of dimension  $m \times n$ , the BOUNDARY PHASE terminates within  $O(\max\{m-1, n-1\})$  rounds.

*Proof.* If possible let us assume that BOUNDARY PHASE never terminates. This implies, no robot ever reaches the resource  $res$  and INITGATHER CONFIGURATION is never achieved. Now we claim the following:

*Claim (1).* Within  $O(\max\{m-1, n-1\})$  rounds, the resource must cross at least once or move on to any one of  $PD(r)$  or  $PD(r')$ .

Let us first assume that if possible our claim is false, i.e., the resource  $res$  never moves onto and never crosses  $PD(r)$  and  $PD(r')$ . So  $res$  can never be on  $PD(r)$  or on  $PD(r')$  at the beginning of BOUNDARY PHASE. Now there are three cases depending on the location of  $res$  at the beginning of the BOUNDARY PHASE.

**Case 1:** Let us assume that  $res$  is inside  $R_{SE}$ . Also, observe that  $res$  can never get out of  $R_{SE}$  as otherwise, it has to move onto or cross any one of  $PD(r)$  or  $PD(r')$ . so  $dist(r), dist(r') \geq 1$  in every rounds. So according to the algorithm both  $r$  and  $r'$  move along  $BD(r)$  and  $BD(r')$  respectively towards the direction of  $res$ . So, in each round the height and width of the quadrant  $R_{SE}$  decreases by one unit. Now since initially the  $R_{SE}$  was of dimension  $(m - 1 \times (n - 1))$  so within  $O(\max\{m - 1, n - 1\})$ , the resource must move onto or crosses either  $PD(r)$  or  $PD(r')$ . A contradiction.

**Case 2:** Let us assumes  $res$  is inside  $R_{NE}$  at the beginning of BOUNDARY PHASE, i.e., According to Fig 5,  $res$  is on the line segment of  $BD(r')$  but on the right side of  $r'$ . Now observe that in this case though for both the robots  $dist(r)$  and  $dist(r') \geq 1$ ,  $r$  never moves as otherwise, it has to move to the door vertex. So  $res$  can never move below  $BD(r')$  otherwise it would move onto  $PD(r)$  and we will reach a contradiction. So, even if the  $res$  moves it must move on the line segment of  $BD(r')$  which is on the right side of  $r'$ . Also since  $r'$  never reaches the resource  $res$ ,  $dist(r') \geq 1$ , always in each round. So  $r'$  also moves in each round along  $BD(r')$  towards  $r'$ . Hence the length of the line segment of  $BD(r')$  which is on the right of  $r'$  decreases in each round. Initially, at the beginning of BOUNDARY PHASE, the length of that line segment was  $n - 1$  unit. So within  $n - 1$  rounds  $res$  either move onto  $PD(r)$  or move into  $r'$ . Again we reach a contradiction.

**Case 3:** Let the resource be inside  $R_{SW}$  at the beginning of BOUNDARY PHASE. This case is similar to case 2. and we will again reach a contradiction. So our assumption was wrong. Hence Claim (1).

So, if  $res$  does not move into any one of  $r$  or  $r'$  and the configuration does not becomes an INITGATHER CONFIGURATION then  $res$  must move onto or crosses any one of  $PD(r)$  and  $PD(r')$  within  $\max\{m - 1, n - 1\}$  rounds in the worst case. Now we again claim the following:

*Claim (2).* If the  $res$  has moved onto or crosses  $PD(R)$  ( $R \in \{r, r'\}$ ) at some round (say  $t$ ), then  $dist(R) \leq 1$  from round  $t$  on wards.

Without loss of generality let  $res$  have crossed or moved onto  $PD(r)$  at round  $t$ . So at the beginning of round  $t + 1$ ,  $dist(r)$ , must be less or equal to one.

**Case 1:** Let  $res$  be on  $PD(r)$  at the beginning of round  $t + 1$ . Then  $dist(r) = 0$ . Now during round  $t + 1$ ,  $res$  either moves parallel to  $PD(r)$  (horizontally in Fig. 5) or, Perpendicular to  $PD(r)$  (vertically in Fig. 5) or does not move at all. Now if  $res$  moves parallel to  $PD(r)$  or does not move at all, then  $dist(r)$  remains the same after completion of round  $t + 1$  according to the algorithm of BOUNDARY PHASE. On the other hand, If  $res$  moves Perpendicular to  $PD(r)$  during round  $t + 1$ , then  $dist(r)$  becomes one after the completion of round  $t + 1$ .

**Case 2:** Let  $res$  crosses  $PD(r)$  at round  $t$ . Then at the beginning of the round  $t + 1$ ,  $dist(r) = 1$ . Now if  $res$  moves parallel to  $PD(r)$  or does not move at all during round  $t + 1$  then after the completion of the round  $dist(r)$  either stays one or decreases to zero. Now let  $res$  moves Perpendicular to  $PD(r)$  during round  $t + 1$ , then if  $res$  moves towards  $PD(r)$  then  $dist(r)$  either remains same as one (as  $r$  also moves towards  $res$  along  $BD(r)$ ) or becomes zero in case  $r$  does not move along  $BD(r)$ . On the other hand, if  $res$  moves away from  $PD(r)$  during round  $t + 1$ , then  $dist(r)$  remains one after completion of round  $t + 1$  as  $r$  also moves during round  $t + 1$  towards the direction of  $res$  along  $BD(r)$ .

So after completion of round  $t + 1$ ,  $dist(r)$  is still less or equal to 1. Now with similar arguments, it is easy to see that if after completion of round  $t + i$ ,  $dist(r) \leq 1$ , then  $dist(r) \leq 1$  after completion of round  $t + i + 1$  for some natural number  $i$ . Hence We can conclude Claim (2). Now we claim another statement below.

*Claim (3).* If  $res$  has moved onto or crossed  $PD(R)$  where  $R \in \{r, r'\}$  at some round  $t$ , then  $res$  never crosses  $PD(R')$  at round  $t$  on wards (Here  $R' = r$  if  $R = r'$  and  $R' = r'$  if  $R = r$ ).

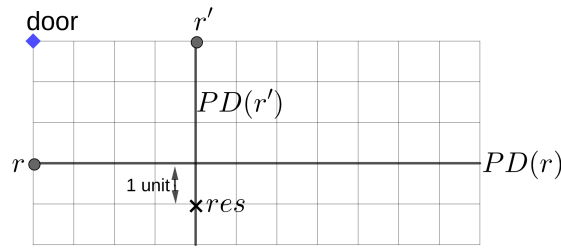


Fig. 6: When  $res$  moves onto  $PD(r')$  even after crossing or moving onto  $PD(r)$  in an earlier round.

Without loss of generality let  $res$  move onto or crosses  $PD(r)$  at some round  $t$  and  $BD(r')$  is the boundary of  $G$  at the north (Fig. 5). We have to show that from round  $t$  onwards  $res$  never crosses or moves onto  $PD(r')$ . If possible let  $res$  move onto or crosses  $PD(r')$  at some round  $t' > t$ . If  $res$  moves onto  $PD(r')$  at round  $t'$  then after completion of round  $t'$ ,  $r'$  and  $res$  are on a line  $L = PD(r')$  and  $dist(r) \leq 1$  (from claim (2)) (Fig. 6). i.e., Perpendicular distance of  $r$  to the line passing through  $res$  and perpendicular to  $BD(r)$  is at most one. Now, since  $BD(r)$  and  $PD(r') = L$  are parallel after completion of round  $t'$ , the configuration becomes an INITGATHER CONFIGURATION which is a contradiction. So, let us assume that at round  $t' > t$ ,  $res$  has crossed  $PD(r')$ . So at the beginning of the round  $t'$ ,  $res$  must be on  $R_{NE} \cup R_{SE}$ .

**Case 1:** Let  $res$  was on  $PD(r)$  at the beginning of round  $t'$  and  $res$  crosses  $PD(r')$  during the round  $t'$ . This implies,  $dist(r') \leq 1$  and  $res$  are on  $PD(r)$  at the beginning of round  $t'$ . Let  $PD(r) = L$  (Fig. 7). Also, the line passing through  $res$  and perpendicular to  $L$  is parallel to  $PD(r')$ . So  $dist(r') =$  perpendicular distance of  $r'$  to the line passing through  $res$  and perpendicular to  $L \leq 1$ . Hence, at the beginning of round  $t'$  the configuration is an INITGATHER CONFIGURATION. Hence a contradiction.

**Case 2:** Let  $res$  was in  $R_{SE}$  and not on  $PD(r)$  at the beginning of round  $t'$  and it crosses  $PD(r')$  during this round. Note that in this case  $res$  moves parallel to  $PD(r)$  (according to Fig. 8, horizontally). Also,  $r$  moves along  $BD(r)$  opposite of the door. Since initially at the beginning of round  $t'$ ,  $dist(r) = 1$  (by claim (2) and  $res$  is not on  $PD(r)$ ), after completion of the round  $dist(r)$  becomes zero i.e.,  $res$  moves on to  $PD(r)$  (Fig. 8). Now since during this round  $res$  also crosses  $PD(r')$ ,  $dist(r')$  becomes

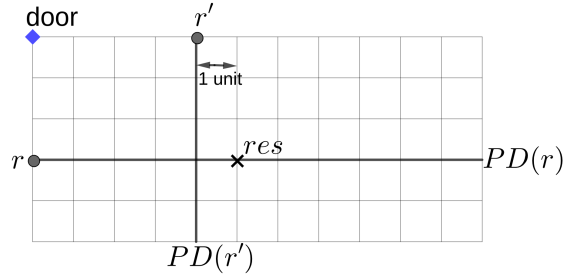


Fig. 7: When  $res$  is on  $PD(r)$  at the beginning of round  $t'$  and at a distance 1 unit from  $PD(r')$  then configuration is an INITGATHER CONFIGURATION.

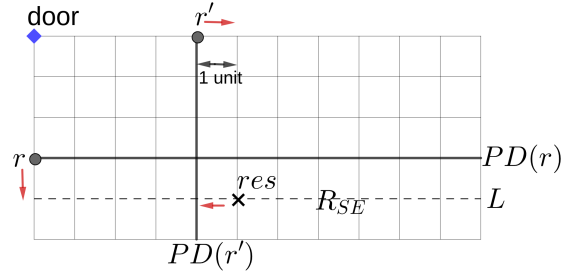


Fig. 8:  $res$  is on  $R_{SE}$  and not on  $PD(r)$  or  $PD(r')$  at the beginning of round  $t'$ .  $res$  crosses  $PD(r')$  during round  $t'$ . the red arrows denote the moves of the robots and the resource during round  $t'$ .

one. Now considering  $PD(r) = L$  it is easy to see that after completion of round  $t$ , the configuration becomes an INITGATHER CONFIGURATION. Which is again a contradiction.

**Case 3:** Let  $res$  is in  $R_{NE}$  and not on  $PD(r)$  at the beginning of round  $t'$  and it crosses  $PD(r')$  during this round. Let at the beginning of round  $t'$  dimension of  $R_{NE}$  is  $m' \times n'$  where  $n' \geq 2$  (as a robot never moves to the door vertex during the BOUNDARY PHASE). If  $n' > 2$  (i.e., destination vertex of  $r$  is not the door vertex) then observe that during the round  $t'$ ,  $res$  moves parallel to  $PD(r)$  (horizontally according to Fig. 9) and  $r$  moves along  $BD(r)$  towards  $res$ . Now as  $dist(r) = 1$  (by claim (2)) and the fact that  $res$  is not on  $PD(r)$  at the beginning of round  $t'$ , after completion of the round,  $res$  moves onto  $PD(r)$ . Also,  $dist(r')$  becomes one after completion of round  $t'$  as  $res$  just crosses  $PD(r')$  during this round (Fig. 9). So considering  $PD(r) = L$  it is easy to see that after completion of round  $t'$ , the configuration becomes an INITGATHER CONFIGURATION, a contradiction. So let us consider  $n' = 2$ . In this case,  $res$  must be on  $BD(r')$  and  $dist(r') = 1$  at the beginning of round  $t'$  i.e.,  $res$  is on the vertex which is on  $BD(r')$  and adjacent to the vertex of  $r'$  at the beginning of round  $t'$  (Fig. 10). Now since we are assuming  $res$  crosses  $PD(r')$  during the round  $t'$ , that means  $res$  must have crossed  $r'$  during round  $t'$  along  $BD(r')$  but that is a contradiction as it would cause a collision of  $res$  and  $r'$ .

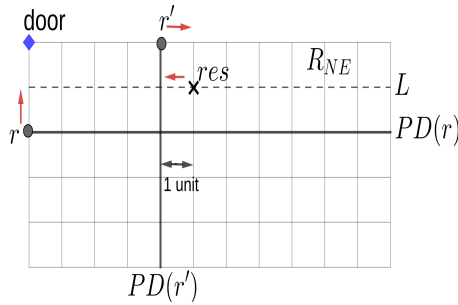


Fig. 9:  $res$  is on  $R_{NE}$  where height of  $R_{NE}$  is more than 2, but not on  $PD(r)$  or,  $PD(r')$ , at the beginning of round  $t'$ . The red arrows denote the moves of the robots and the resource during round  $t'$ .

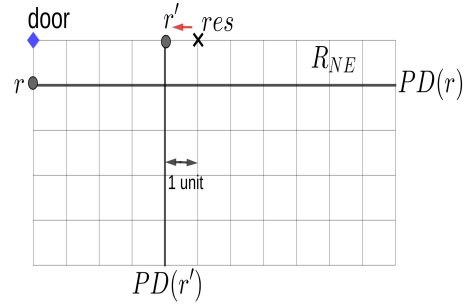


Fig. 10: When  $R_{NE}$  has height 2 at the beginning of round  $t'$ . Note that  $res$  must be on  $BD(r')$  and can not cross  $PD(r')$  without colliding with  $r'$ . The red arrows denote the moves during round  $t'$ .

In all of these above cases, we arrive at contradictions. Hence our assumption was wrong. So if  $res$  crosses or moves onto  $PD(R)$  for some  $R \in \{r, r'\}$  at some round  $t$  then it never crosses or moves onto  $PD(R')$  in the rounds onwards where  $R' = r$  if  $R = r'$  and  $R' = r'$  if  $R = r$ .

From the above three claims we conclude that  $res$  must cross or moves onto  $PD(R)$  where  $R \in \{r, r'\}$  at some round  $t$  within  $O(\max\{m-1, n-1\})$  rounds and  $dist(R) \leq 1$  from round  $t$  onwards. Also we have shown that  $res$  never crosses  $PD(R')$  again where  $R' = r$  if  $R = r'$  and  $R' = r'$  if  $R = r$  from round  $t$  onwards.

Now without loss of generality let,  $res$  has moved onto or crossed  $PD(r)$  at some round  $t$  where  $BD(r')$  is the boundary on the north of the grid (Fig. 5). Then from round  $t$  onwards  $res$  must lie inside  $(R_{NE} \cup R_{SE}) \setminus PD(r')$  i.e.,  $dist(r') \geq 1$  from round  $t$  onwards. So,  $r'$  must move away from the door along  $BD(r')$  in each round from round  $t$  onwards. Let after completion of round  $t$  the dimension of  $(R_{NE} \cup R_{SE})$  is  $m' \times n'$ , where the length  $n' < n$ . Note that after  $n'$  rounds  $(R_{NE} \cup R_{SE}) \setminus PD(r') = \phi$  as length of the rectangle  $(R_{NE} \cup R_{SE})$  decreases in each round due to the move of  $r'$ . So,  $res$  can not stay inside  $(R_{NE} \cup R_{SE}) \setminus PD(r')$  for all rounds after the round  $t$ . We arrive at this contradictory conclusion because our primary assumption about the termination of BOUNDARY PHASE was wrong. So BOUNDARY PHASE must terminate within  $O(\max\{m-1, n-1\})$  rounds, and hence the Theorem  $\square$

### 4.3 GATHER PHASE

GATHER PHASE starts if none of the two robots reaches the location of  $res$  after the termination of the BOUNDARY PHASE. Throughout the execution of this phase, the configuration will remain an INITGATHER CONFIGURATION (Lemma 1). So, in each round, a robot will lie on the same line (say  $L$ ) along with  $res$ , and the perpendicular distance of the other robot to the line passing through  $res$  and perpendicular to  $L$  must be at most one. During

this phase, if a robot is in the same location with  $res$ , it terminates and the other robot moves to the location of  $res$  along any shortest path. On the other hand when none of the robots are on the same vertex with  $res$ , the robot on  $L$  checks if  $res$  is adjacent to it. If  $res$  is not on its adjacent vertex, it moves towards  $res$  along  $L$ . Otherwise, if  $res$  is on its adjacent vertex then it only moves towards  $res$  along  $L$  if it sees  $res$  is on a corner and the other robot is also on another adjacent vertex of  $res$ . Now if the robot is not on any line along with the resource  $res$ , that implies its perpendicular distance to the line through  $res$  and perpendicular to  $L$  is one. In this case, the robot will move parallel to  $L$  towards  $res$ .

---

**Algorithm 4:** Gather Phase for robot  $r$

---

```

1 if  $r$  is on same vertex with  $res$  then
2   | terminate;
3 else
4   if  $r'$  is on the same vertex with  $res$  then
5     | move to  $res$  along any shortest path avoiding door vertex;
6   else
7     if  $r$  is on a line  $L$  with  $res$  then
8       | if  $res$  is not adjacent to  $r$  then
9         | move towards  $res$  along  $L$ ;
10      | else
11        | if  $res$  is at a corner and  $r'$  is adjacent to  $res$  then
12          | | move towards  $res$  along  $L$ ;
13      | else
14        | | move parallel to  $L$  towards  $res$ ;

```

---

Now we prove the following lemma that proves our claim in the description of this phase that during the Execution of GATHER PHASE the configuration at the beginning of any round is an INITGATHER CONFIGURATION.

**Lemma 1.** *Let at the beginning of some round  $t$  the configuration is an INITGATHER CONFIGURATION, then at the beginning of round  $t + 1$  the configuration will again be an INITGATHER CONFIGURATION*

*Proof.* Let at the beginning of some round  $t$ , the configuration is an INITGATHER CONFIGURATION. This implies there is a robot say  $r$ , which is on line with the resource  $res$  on the grid  $G$ . Let us call this line  $L$ . Also the perpendicular distance of the other robots  $r'$  is at most one to the line passing through  $res$  and perpendicular to the line  $L$ . Observe that during the round  $t$ ,  $res$  can either move parallel to  $L$  or perpendicular to  $L$  or does not move at all.

**Case 1:** Let us consider the case where  $res$  is moving Parallel to  $L$ . In this case since  $r$  moves along  $L$  or if does not move at all, it would still be on the line along with  $res$  after the completion of the round. Note that since  $res$  moves along  $L$  the perpendicular to  $L$  and passing through  $res$  (say,  $L'$ ) shifts along  $L$ . Now if the other robot (say  $r'$ ) is one unit apart from  $L'$  at the beginning of round  $t$  then,  $r'$  moves parallel to  $L$  towards  $res$  i.e., towards  $L'$  (Fig. 11), the perpendicular distance of  $r'$  to  $L'$  still remains one



after completion of the round. So at the beginning of the round  $t + 1$ , the configuration again is an INITGATHER CONFIGURATION. Now if at the beginning of round  $t$ ,  $r'$  is on  $L'$  then,  $r'$  moves along  $L'$  towards  $res$  (Fig. 12). Now even if  $res$  moves along  $L$ , after completion of round  $t$ ,  $r'$  is at most one unit apart from  $L'$ . So the configuration is still an INITGATHER CONFIGURATION at the beginning of round  $t + 1$ .

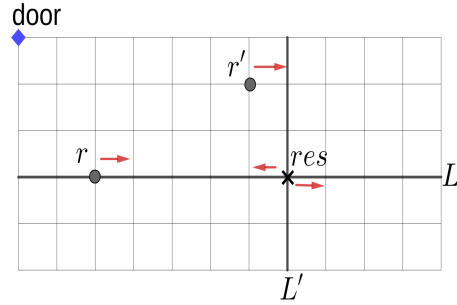


Fig. 11:  $r'$  is one unit apart from  $L'$  and  $res$  moves along  $L$  at the beginning of round  $t$ . The red arrows are the direction of movement of the robots and the resource.

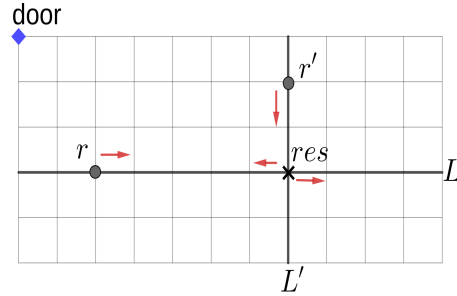


Fig. 12:  $r'$  is on  $L'$  and  $res$  moves along  $L$  at the beginning of round  $t$ . The red arrows are the direction of movement of the robots and the resource.

**Case 2:** Let us consider the case where  $res$  moves perpendicular to the line  $L$  during the round  $t$ . Observe that here  $res$  moves along the line  $L'$ , the line perpendicular to  $L$  and passing through  $res$ . So, the line does not shift during round  $t$ . Now, if  $r'$  is one unit apart from  $L'$  then, it moves parallel to  $L$  towards  $res$  (i.e., towards  $L'$ ). Thus, the distance of  $r'$  to  $L'$  decreases to zero after the completion of the round (Fig. 13). Also, if at the beginning of round  $t$ ,  $r'$  is on  $L'$  then even if  $r'$  moves it remains on  $L'$  as  $res$  also moves perpendicular to  $L$  i.e., along  $L'$  during round  $t$  (Fig. 14). So after completion of round  $t$ ,  $r'$  and  $res$  will be on same line i.e.,  $L'$  for both the cases of  $r'$  being on  $L'$  or not. Also note that during round  $t$ ,  $r$  remains on the line  $L$  irrespective of whether it moved or not. Now, the line perpendicular to  $L'$  and passing through  $res$  (say  $L''$ ) is parallel to  $L$  and one unit apart from  $L$ . Thus after completion of round  $t$ , the perpendicular distance of  $r$  to the line  $L''$  is one. Thus after completion of round  $t$  and hence at the beginning of round  $t + 1$ , the configuration is still an INITGATHER CONFIGURATION.

**Case 3:** Now, let us consider the case where  $res$  does not move at all. In this case, the robot  $r$  on the line  $L$  along with  $res$  stays on the same line  $L$  along with  $res$ , even if it moves. This is because  $r$  moves along  $L$  according to the algorithm for GATHER PHASE. Note that, the line  $L'$  passing through  $res$  and perpendicular to  $L$  does not shift as  $res$  is not moving. Also observe that the other robot,  $r'$  moves parallel to  $L$  towards  $res$  i.e., towards  $L'$  if at the beginning of round  $t$ ,  $r'$  is one unit apart from  $L'$  (Fig. 15) and moves along  $L'$  if it was already on  $L'$  at the beginning of round  $t$  (Fig. 16). So, after completion of the round  $t$ , the distance of  $r'$  to  $L'$  must be zero. Thus after completion of

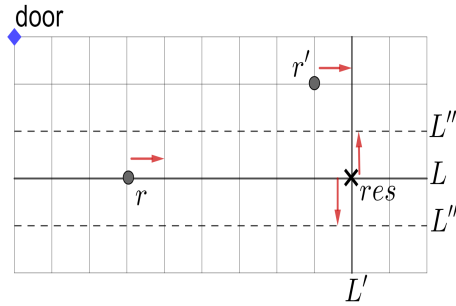


Fig. 13: Diagram of an INITGATHER CONFIGURATION where  $res$  moves perpendicular to  $L$  and  $r'$  is not on  $L'$  at the beginning of the round  $t$ .  $L''$  is the line parallel to  $L$  on which  $res$  moves during round  $t$ .

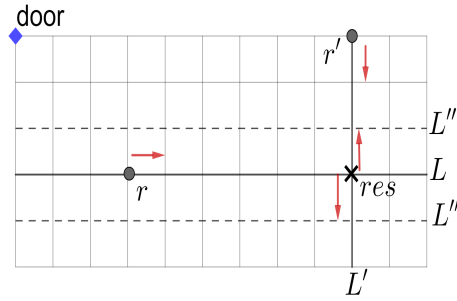


Fig. 14: Diagram of an INITGATHER CONFIGURATION where  $res$  moves perpendicular to  $L$  and  $r'$  is on  $L'$  at the beginning of the round  $t$ .  $L''$  is the line parallel to  $L$  on which  $res$  moves during round  $t$ .

round  $t$ , and so at the beginning of round  $t + 1$ , the configuration is again an INITGATHER CONFIGURATION. Hence the proof.  $\square$

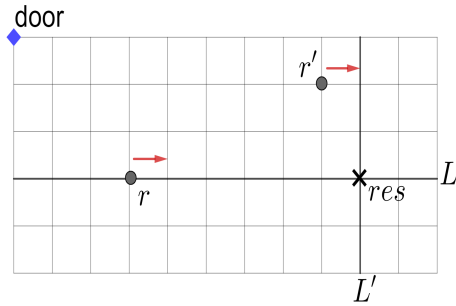


Fig. 15: Diagram of an INITGATHER CONFIGURATION where  $res$  does not move and  $r'$  is not on  $L'$  at the beginning of the round  $t$ .

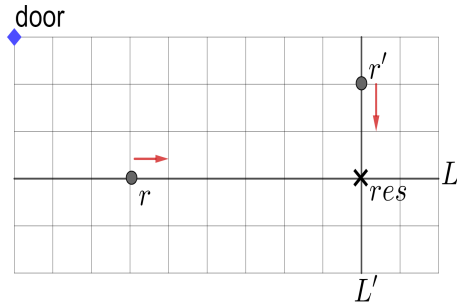


Fig. 16: Diagram of an INITGATHER CONFIGURATION where  $res$  does not move and  $r'$  is on  $L'$  at the beginning of the round  $t$ .

**Lemma 2.** *If none of the two robots have terminated and both of them are not adjacent to  $res$  at a corner then, during the execution of GATHER PHASE, two robots never are on the same line.*

*Proof.* Let after completion of round  $t$ , the execution of GATHER PHASE has been started. Note that at the beginning of round  $t + 1$ , the robots were in different boundaries of the grid and hence they were not on the same line. If possible let  $t' > t + 1$  be the round

where two robots move to be on the same line for the first time. Observe that at the beginning of round  $t'$ , there is a robot (say,  $r$ ) that must be on a line (say,  $L$ ) along with  $res$ . And according to the algorithm,  $r$  must stay on  $L$  after the completion of round  $t'$ . Let  $I$  be the line perpendicular to  $L$  and passing through  $r$  after the completion of the round.

*Claim (4).* We claim that the other robot  $r'$  never moves to  $I$  during the round  $t'$ .

Let,  $r'$  is not on  $L'$ , the line passing through  $res$  and perpendicular to  $L$ . Thus  $r'$  must move parallel to  $L$  during round  $t'$ . Since  $r'$  moves parallel to  $L$  and hence perpendicular to  $I$  towards the direction of  $res$ , it moves to  $I$  only if  $res$  and  $r$  are on the same direction of  $r'$  along  $L$ . Now, observe that  $L'$  must be between  $r'$  and  $I$  at the beginning of round  $t'$  (Fig. 17). This implies that  $r'$  moves onto  $I$  during round  $t'$  only if both  $r'$  and  $r$  move onto  $L'$  during round  $t'$  (i.e.,  $L' = I$  after completion of the round). But observe that  $r$  moves to  $L'$  implies,  $r$  must be adjacent to  $res$  on line  $L$  at the beginning of round  $t'$ . This implies  $r$  does not move (as  $res$  is not at the corner with  $r'$  adjacent to  $res$ ). Hence we reach a contradiction and thus we can assure that to be on the same line  $r'$  must move onto  $L$  during round  $t'$ .

Now observe that if the distance of  $r'$  to  $L'$  is one then  $r'$  must move parallel to  $L$  and thus, never reaches  $L$ . So let the distance of  $r'$  to  $L'$  is zero at the beginning of round  $t'$ . This implies  $r'$  is also on the same line  $L'$  along with  $res$ . Now,  $r'$  moves on to  $L$  during round  $t'$ . Thus at the beginning of round  $t'$ ,  $r'$  must be adjacent to  $res$  (Fig. 18). This implies during round  $t'$ ,  $r'$  does not move at all (as  $res$  is not at a corner with both the robots on two adjacent vertices). We arrive at a contradiction assuming the existence of a round where both the robots will move on to the same line. Our assumption was wrong and hence the lemma.  $\square$

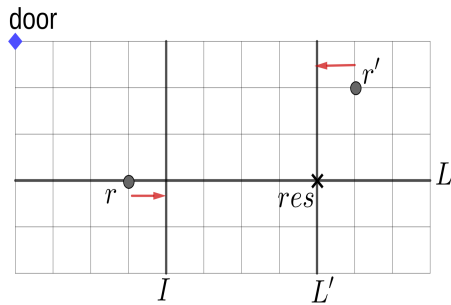


Fig. 17:  $r'$  is not on  $L'$  then  $r'$  never moves onto  $I$  during the round  $t'$ .

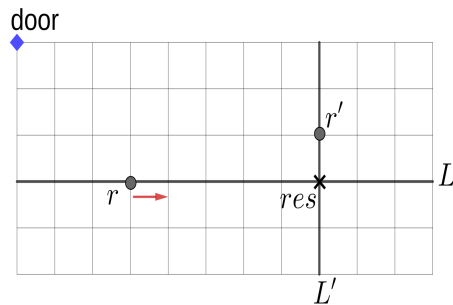


Fig. 18:  $r'$  is on  $L'$  then  $r'$  never moves onto  $L$  during the round  $t'$ .

Let at the beginning of a particular round during the GATHER PHASE take any robot (say  $r$ ) which is on the same line  $L$  along with the resource,  $res$ . Let us define two lines, firstly,  $L_1$  passing through  $r$  and perpendicular to  $L$ , and secondly  $L_2$ , passing through the vertex of the other robot  $r'$  and parallel to  $L$ . Note that the lines  $L_1$  and  $L_2$  divides

the entire grid into one or more rectangles. The rectangle inside of which the resource  $res$  is located is called the "Containing Rectangle" and it is denoted as  $R_{Con}$  (Fig. 19). Observe that, at the beginning of the first round of GATHER PHASE,  $L_1$  is  $BD(r)$  and  $L_2$  is  $BD(r')$  and  $R_{Con} = G$ .

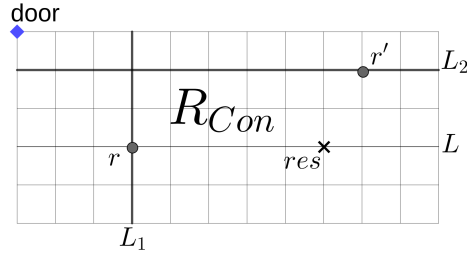
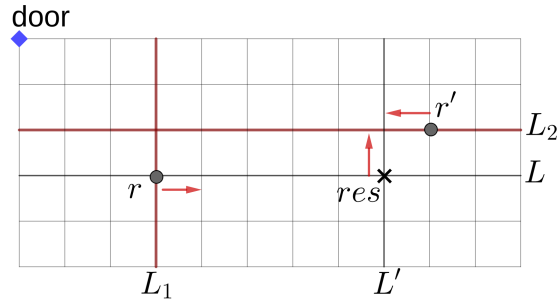


Fig. 19: Diagram of  $R_{Con}$

**Lemma 3.** *The resource  $res$  never moves onto  $L_1$  or  $L_2$  during the GATHER PHASE without arriving at the same location with a robot.*

*Proof.* Initially at the beginning of GATHER PHASE, resource  $res$  must not be on  $BD(r)$  or  $BD(r')$ . Otherwise,  $res$  must have crossed or moved onto both  $PD(r)$  and  $PD(r')$  during the BOUNDARY PHASE, which is not possible due to claim (3). If possible let during a particular round  $t$ , the resource  $res$  first moves on to any one of  $L_1$  or  $L_2$  for the first time without arriving at the same location with a robot. Let  $res$  moves onto  $L_1$  after completion of round  $t$ . So, the perpendicular distance of  $res$  to  $L_1$  is one at the beginning of round  $t$ . Now,  $r$  is on the line  $L$  along with  $res$  at the beginning of round  $t$ . So,  $r$  is adjacent to  $res$  at the beginning of round  $t$ . For moving onto  $L_1$ ,  $res$  must move along  $L$  towards  $r$ , so after completion of the round  $t$ ,  $res$  reaches the same vertex as  $r$ . Thus after completion of round  $t$ , the location of  $r$  and  $res$  becomes the Same. So  $res$  never moves onto  $L_1$ . Now let  $res$  move onto  $L_2$  during round  $t$ . Let  $L'$  be the line perpendicular to  $L$  and passing through  $res$ . If the perpendicular distance of  $r'$  to  $L'$  is zero, i.e.,  $r'$  is also on line  $L'$  along with  $res$  at the beginning of round  $t$  then, as the same logic above we can show that  $res$  can not move onto  $L_2$  during round  $t$ . So let us assume the perpendicular distance of  $r'$  to the line  $L'$  is one at the beginning of round  $t$ . In this case,  $r'$  moves along  $L_2$  towards the direction of  $res$ , and  $res$  moves along  $L'$  towards  $L_2$  during round  $t$ . Now since we have assumed  $res$  moves onto  $L_2$  during round  $t$ ,  $L_2$  must be one distance away from  $res$  along  $L'$ . Also  $L'$  is one distance away from  $r'$  along  $L_2$  and  $r'$  moves along  $L_2$  towards  $L'$  (Fig. 20). So after completion of round  $t$ , both  $r'$  and  $res$  meet at the same vertex, which is a contradiction. Hence  $res$  never moves onto  $L_1$  or  $L_2$  during the GATHER PHASE without arriving at the same location with a robot.  $\square$

**Corollary 1.** *The resource  $res$  never moves outside  $R_{Con}$  during the GATHER PHASE without arriving at the same location along with a robot.*


 Fig. 20: Diagram of  $R_{Con}$ 

*Proof.* If  $res$  moves out of  $R_{Con}$  then it must cross either  $L_1$  or  $L_2$  without moving onto them.

Now the resource,  $res$  never crosses  $L_1$  without reaching the same vertex along with  $r$ . Also, for the same reason,  $res$  never crosses  $L_2$  while  $r'$  is also on the same line along with  $res$ . So let us assume  $r'$  is not on the same line along with  $res$  at the beginning of some round  $t$  during which  $res$  crosses  $L_2$ . Now if  $r'$  is not on a line along with  $res$  then it must move along  $L_2$  during round  $t$ . So, the line  $L_2$  does not shift after the completion of round  $t$ . This implies  $res$  must move onto  $L_2$  to cross it which is not possible due to Lemma 3. Hence  $res$  never moves out of  $R_{Con}$ .  $\square$

Let at the beginning of the first round of GATHER PHASE,  $R_{Con}$  be a  $m_1 \times n_1$  grid. Let the height and width of  $R_{Con}$  be  $m_1$  and  $n_1$  where both  $m_1 > 2$  and  $n_1 > 2$ . We will prove that within  $T_f + 1$  rounds either both  $m_1$  and  $n_1$  decrease or one of  $m_1$  and  $n_1$  decreases and the other one stays the same.

**Lemma 4.** *If both height and width of the  $R_{Con}$  be more than two during the GATHER PHASE then, within  $T_f + 1$  rounds, if no robots are terminated either both height and width of  $R_{Con}$  decreases or one of height or width decreases and the other remains same.*

*Proof.* Let at the beginning of some round  $t$  during the GATHER PHASE,  $r$  be a robot on the line  $L$  along with the resource,  $res$ . The lines  $L_1$  (line passing through  $r$  and perpendicular to  $L$ ) and  $L_2$  (line passing through the vertex of the other robot,  $r'$  and parallel to  $L$ ) and the boundaries that do not contain the door vertex forms a rectangle  $R_{Con}$ . We have proved that  $res$  is contained within  $R_{Con}$  and never moves out of it. Let the dimension of  $R_{Con}$  at the beginning of round  $t$  be  $m_1 \times n_1$  where both  $m_1$  and  $n_1$  are greater than two. Thus if  $res$  is at the corner at the beginning of round  $t$ , both  $r$  and  $r'$  are not adjacent to  $res$ . Thus during round  $t$ , no robot moves to the location of  $res$ .

**Case 1:** Now let at the beginning of round  $t$ ,  $r$  is not adjacent to  $res$ . Also, let without loss of generality number of vertices on side of  $R_{Con}$  which is parallel to  $L$  at the beginning of round  $t$  is the width of  $R_{Con}$ . Now according to the algorithm,  $r$  moves along  $L$  towards  $res$  i.e towards the direction of the interior of  $R_{Con}$ . Hence  $L_1$  shifts towards the interior of  $R_{Con}$ . So the width of  $R_{Con}$  decreases during round  $t$ . Now, if  $r'$  is not on any line with  $res$  or adjacent to  $res$  on the line  $L'$  (line perpendicular to  $L$  and

passing through  $res$ ) at the beginning of round  $t$  then,  $r'$  moves along  $L_2$  (Fig. 21) or does not move at all. In both of these cases, the height of  $R_{Con}$  remains the same after the completion of the round. on the other hand if at the beginning of round  $t$ ,  $r'$  is on  $L'$  along with  $res$  and not adjacent to  $r'$  then,  $r'$  moves along  $L'$  towards the direction of  $res$  (Fig. 22). Note that in this case  $L_2$  also shifts towards the interior of  $R_{Con}$  and decreases the height of  $R_{Con}$  after completion of round  $t$ .

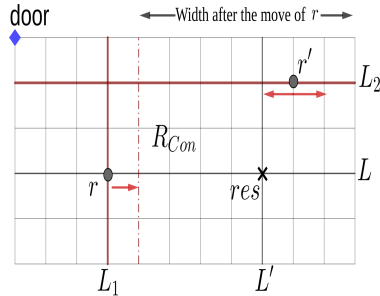


Fig. 21: Only width of  $R_{Con}$  decreases and height remains same.

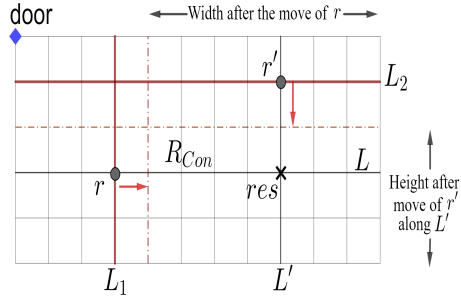


Fig. 22: both height and width of  $R_{Con}$  decreases.

So we have shown that if  $r$ , a robot on a line  $L$  with  $res$  is not adjacent to  $res$  then either both height and width decrease or only width decreases in one round.

**Case 2:** Let, if  $res$  is adjacent to  $r$  on  $L$  at the beginning of round  $t$  then,  $r$  will not move along  $L$ . It is assumed that  $res$  will not stay at the same location for more than  $T_f$  consecutive rounds. Note that since both height and width are more than two,  $res$  gets an empty vertex to move. Now in the worst case during the round  $t + T_f$ ,  $res$  must have moved either along  $L$  or along  $L'$  (line perpendicular to  $L$  and passing through  $res$ ). Note that  $res$  can not move towards  $r$  along  $L$  as it would end up at the same location as  $r$ .

**Case 2(a):** Let during the round  $t + T_f$ ,  $res$  moves along  $L$  opposite to  $r$  then, at the beginning of round  $t + T_f + 1$ ,  $r$  and  $res$  are not adjacent along  $L$ . Hence during this round, either only the width of  $R_{Con}$  decreases and height remains the same, or both height and width of  $R_{Con}$  decrease by a similar argument as case 1.

**Case 2(b):** Now let us consider the case where  $r$  is adjacent to  $res$  on  $L$  at the beginning of round  $t + T_f$  but  $res$  moves perpendicular to  $L$  i.e., along  $L'$  during the round  $t + T_f$ .

If at the beginning of round  $t + T_f$ ,  $r'$  was on  $L'$  then, by the same argument as in Case 1 and Case 2(a) we can conclude that in the worst case, after completion of round  $t + T_f + 1$ , the height of  $R_{Con}$  must decrease while width either remains same or also decreases.

Let us now consider  $r'$  is not on  $L'$  at the beginning of round  $t + T_f$ . In this case, after completion of round  $t + T_f$ ,  $r'$  and  $res$  must be on the line  $L'$  and hence according to the same argument as above cases in the worst during round  $t + 2T_f + 1$  either height and width of  $R_{Con}$  both decrease or height decreases while the width remains same.  $\square$

**Corollary 2.** *no robot moves to door vertex during GATHER PHASE.*

*Proof.* Let at round  $t$ , GATHER PHASE is started. Note that at the beginning of round  $t$ , no robot is on the door vertex as no robot moves to the door vertex during BOUNDARY PHASE. Now, after completion of round  $t$ , at least one robot must leave the boundary and moves inside  $R_{Con}$ . So from round  $t + 1$  onward, door vertex must remain outside of  $R_{Con}$ . This is because by lemma 4 height or width of  $R_{Con}$  never increase to include the door vertex inside it. Hence the result.  $\square$

By Corollary 2 and Lemma 1 we can conclude that, after one execution of the GATHER PHASE, if no robot is terminated then in the next round, GATHER PHASE will be executed again. Also, from the above Lemma 4 it is evident that if the dimension of the  $R_{Con}$  is  $m_1 \times n_1$  where both  $m_1 > 2$  and  $n_1 > 2$  then, in the worst case in every  $2T_f + 1$  round, the height or the width of the configuration decreases and none of them ever increase. So within  $O(T_f \times (m + n))$  there will be a round (say  $t_0$ ) when either the height or the width of  $R_{Con}$  becomes two. Without loss of generality let the dimension of  $R_{Con}$  be  $m_1 \times 2$ , at the beginning of round  $t_0$ , where  $m_1 > 2$ . Now we claim the following lemma.

**Lemma 5.** *If at the beginning of some round  $t_0$ , the dimension of  $R_{Con}$  is  $m_1 \times 2$  (resp.  $2 \times n_1$ ) then, within  $(T_f + 1)(m_1 - 2)$  (resp.  $(T_f + 1)(n_1 - 2)$ ) rounds if no robots are terminated, dimension of  $R_{Con}$  becomes  $2 \times 2$ .*

*Proof.* Without loss of generality let the dimension of  $R_{Con}$  be  $m_1 \times 2$  at the beginning of round  $t_0$  where  $m_1 > 2$ . This implies exactly one of the height or width of  $R_{Con}$  is two. Without loss of generality let the width is two and height be  $m_1 > 2$ . Thus  $R_{Con}$  consists of exactly two lines perpendicular to the width. One of these two lines (say,  $L$ ) is a boundary of  $G$  which does not contains the door vertex and the other one is the line parallel and adjacent to it (Say  $L_2$ ). Now by Lemma 2 each of these two lines contains exactly one robot. Let without loss of generality  $r$  be on the line  $L$  and  $r'$  is on  $L_2$ . Now by Lemma 3,  $res$  must be on  $L$ . Note that, the configuration at the beginning of round  $t_0$  is an INITGATHER CONFIGURATION as during round  $t_0$  the robots are executing GATHER PHASE. So, the distance of  $r'$  to the line perpendicular to  $L$  and passing through  $res$  (say  $L'$ ) is at most one. So if during round  $t_0$ ,  $res$  moves perpendicular to  $L$  it must move into the location of  $r'$  and  $r'$  terminates. So let us consider  $res$  either move along  $L$  or does not move at all during round  $t_0$  (Fig. 23). Now since  $res$  can not stay at the same location alone for more than  $T_f$  rounds we can assume  $res$  moves along  $L$ . Now, by a similar argument as in Lemma 4 in the worst case within  $T_f + 1$  rounds the height of  $R_{Con}$  must decrease. Since  $m_1 > 2$ , at the beginning of round  $t_0$ , if  $res$  is at a corner,  $r$  is not adjacent to  $res$  and hence  $r'$  can only move parallel to  $L_2$ . So unless  $m_1 = 2$ , width of  $R_{Con}$  remains two. Now since in the worst case in each  $T_f + 1$  round the height of  $R_{Con}$  decreases by a unit, so in  $(T_f + 1)(m_1 - 2)$  rounds, the height of  $R_{Con}$  becomes two while the width still remains two. Hence the lemma.  $\square$

Now we have proved that within  $O(T_f \times (m + n))$  rounds there is a round  $t_1$  such that at the beginning of it,  $R_{Con}$  is a  $2 \times 2$  rectangle on the bottom right corner of the grid  $G$  (Fig. 24). At the beginning of round  $t_1$ ,  $res$  must be at the corner of the Grid

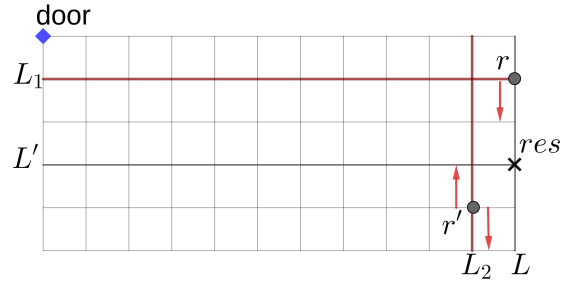


Fig. 23: Height of  $R_{Con}$  is greater than 2 so width remains the same but height decreases.

opposite to the corner of the door vertex (by Lemma 3) Here two robots  $r$  and  $r'$  must be on two different adjacent vertices of  $res$  (by Lemma 2). Hence by the algorithm of GATHER PHASE  $r$  and  $r'$  both move to the vertex of  $res$  during the round  $t_1$ , while  $res$  has no other edges to move out as it is on the corner. So, both robot reaches the location of  $res$  and terminates. From this discussion, we can conclude the following Theorem.

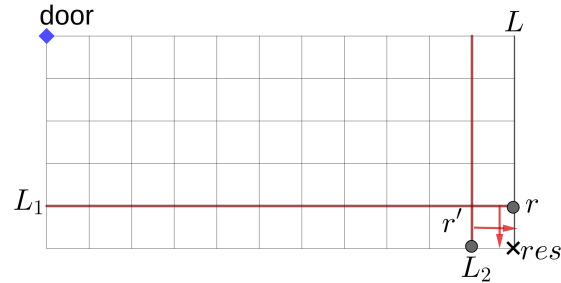


Fig. 24:  $R_{Con}$  has dimension  $2 \times 2$ .

**Theorem 4.** For a grid of dimension  $m \times n$ , the GATHER PHASE terminates within  $O(T_f \times (m + n))$  rounds.

Now since the termination of GATHER PHASE implies termination of the whole algorithm we can conclude with the following theorem.

**Theorem 5.** Algorithm GATHER-DYNAMIC terminates within  $O(T_f \times (m + n))$  rounds.

## 5 Conclusion

Gathering is a classical problem in the field of swarm robotics. Rendezvous is a special case of gathering where two robots gather at a single point in the environment. All the previous works on gathering considered the meeting point to be not known by the robots



but here we have considered the robots to know the meeting point but the meeting point can move in the environment until a robot reaches it. To the best of our knowledge, it is the first work that considers a dynamic meeting point. In this work, we have shown that it is impossible for two robots to gather at a known dynamic meeting point on a finite grid if the scheduler is semi-synchronous. Then considering a fully synchronous scheduler we have provided a distributed algorithm DYNAMIC RENDEZVOUS which gathers the two robots on the known dynamic meeting point called the resource, within  $O(T_f \times (m + n))$  rounds where  $m \times n$  is the dimension of the grid and  $T_f$  is the upper bound of the number of rounds the resource can stay at a single vertex alone. We have also provided a lower bound of time i.e.,  $\Omega(m + n)$  to solve this problem considering a  $m \times n$  grid. So, if  $T_f \leq k$  for some constant  $k$  then our algorithm is time optimal.

For future courses of research, one can think of solving this problem on other different networks such as tree, ring, etc. In ring networks, solving this problem with limited visibility can be really interesting. Also, One can think of finding out the minimum number of robots needed to gather at a known dynamic meeting point for different schedulers in different networks.

## References

1. Bose, K., Adhikary, R., Kundu, M.K., Sau, B.: Arbitrary pattern formation on infinite grid by asynchronous oblivious robots. In: Das, G.K., Mandal, P.S., Mukhopadhyaya, K., Nakano, S. (eds.) WALCOM: Algorithms and Computation - 13th International Conference, WALCOM 2019, Guwahati, India, February 27 - March 2, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11355, pp. 354–366. Springer (2019). [https://doi.org/10.1007/978-3-030-10564-8\\_28](https://doi.org/10.1007/978-3-030-10564-8_28), [https://doi.org/10.1007/978-3-030-10564-8\\_28](https://doi.org/10.1007/978-3-030-10564-8_28)
2. 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>, <https://doi.org/10.1137/100796534>
3. 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>, <https://doi.org/10.1137/S0097539704446475>
4. 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>
5. D’Angelo, G., Stefano, G.D., Navarra, A.: Gathering six oblivious robots on anonymous symmetric rings. J. Discrete Algorithms **26**, 16–27 (2014). <https://doi.org/10.1016/j.jda.2013.09.006>
6. Das, A., Bose, K., Sau, B.: Memory optimal dispersion by anonymous mobile robots. In: Mudgal, A., Subramanian, C.R. (eds.) Algorithms and Discrete Applied Mathematics - 7th International Conference, CALDAM 2021, Rupnagar, India, February 11-13, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12601, pp. 426–439. Springer (2021). [https://doi.org/10.1007/978-3-030-67899-9\\_34](https://doi.org/10.1007/978-3-030-67899-9_34), [https://doi.org/10.1007/978-3-030-67899-9\\_34](https://doi.org/10.1007/978-3-030-67899-9_34)
7. Das, S., Flocchini, P., Prencipe, G., Santoro, N., Yamashita, M.: Autonomous mobile robots with lights. Theor. Comput. Sci. **609**, 171–184 (2016). <https://doi.org/10.1016/j.tcs.2015.09.018>, <https://doi.org/10.1016/j.tcs.2015.09.018>

8. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of asynchronous robots with limited visibility. *Theoretical Computer Science* **337**(1), 147–168 (2005). <https://doi.org/https://doi.org/10.1016/j.tcs.2005.01.001>
9. Flocchini, P., Santoro, N., Viglietta, G., Yamashita, M.: Rendezvous with constant memory. *Theor. Comput. Sci.* **621**, 57–72 (2016). <https://doi.org/10.1016/j.tcs.2016.01.025>, <https://doi.org/10.1016/j.tcs.2016.01.025>
10. Goswami, P., Sharma, A., Ghosh, S., Sau, B.: Time optimal gathering of myopic robots on an infinite triangular grid. In: Devismes, S., Petit, F., Altisen, K., Luna, G.A.D., Anta, A.F. (eds.) *Stabilization, Safety, and Security of Distributed Systems - 24th International Symposium, SSS 2022, Clermont-Ferrand, France, November 15-17, 2022, Proceedings. Lecture Notes in Computer Science*, vol. 13751, pp. 270–284. Springer (2022). [https://doi.org/10.1007/978-3-031-21017-4\\_18](https://doi.org/10.1007/978-3-031-21017-4_18), [https://doi.org/10.1007/978-3-031-21017-4\\_18](https://doi.org/10.1007/978-3-031-21017-4_18)
11. Heriban, A., Défago, X., Tixeuil, S.: Optimally gathering two robots. In: Bellavista, P., Garg, V.K. (eds.) *Proceedings of the 19th International Conference on Distributed Computing and Networking, ICDCN 2018, Varanasi, India, January 4-7, 2018*. pp. 3:1–3:10. ACM (2018). <https://doi.org/10.1145/3154273.3154323>, <https://doi.org/10.1145/3154273.3154323>
12. Izumi, T., Souissi, S., Katayama, Y., Inuzuka, N., Défago, X., Wada, K., Yamashita, M.: The gathering problem for two oblivious robots with unreliable compasses. *SIAM J. Comput.* **41**(1), 26–46 (2012). <https://doi.org/10.1137/100797916>, <https://doi.org/10.1137/100797916>
13. 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>
14. 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>
15. Luna, G.A.D., Uehara, R., Viglietta, G., Yamauchi, Y.: Gathering on a circle with limited visibility by anonymous oblivious robots. In: Attiya, H. (ed.) *34th International Symposium on Distributed Computing, DISC 2020, October 12-16, 2020, Virtual Conference. LIPIcs*, vol. 179, pp. 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). <https://doi.org/10.4230/LIPIcs.DISC.2020.12>
16. Ooshita, F., Tixeuil, S.: Ring exploration with myopic luminous robots. *CoRR abs/1805.03965* (2018), <http://arxiv.org/abs/1805.03965>
17. Poudel, P., Sharma, G.: Time-optimal gathering under limited visibility with one-axis agreement. *Inf.* **12**(11), 448 (2021). <https://doi.org/10.3390/info12110448>
18. 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>, <https://doi.org/10.1016/j.tcs.2007.04.023>
19. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. Comput.* **28**(4), 1347–1363 (mar 1999). <https://doi.org/10.1137/S009753979628292X>, <https://doi.org/10.1137/S009753979628292X>