Gathering Non–Oblivious Mobile Robots

Mark Cieliebak

Institute of Theoretical Computer Science
ETH Zurich
cieliebak@inf.ethz.ch

Abstract. We study the Gathering Problem, where we want to gather a set of n autonomous mobile robots at a point in the plane. This point is not fixed in advance. The robots are very weak, in the sense that they have no common coordinate system, no identities, no central coordination, no means of direct communication, and no synchronization. Each robot can only sense the positions of the other robots, perform a deterministic algorithm, and then move towards a destination point. It is known that these simple robots cannot gather if they have no additional capabilities. In this paper, we show that the Gathering Problem can be solved if the robots are non-oblivious, i.e., if they are equipped with memory.

1 Introduction

We consider a distributed system whose entities are autonomous mobile robots, where the robots can freely move in the two-dimensional plane. The coordination mechanism for these robots is totally *decentralized*, i.e., the robots are completely *autonomous* and no central control is used. The research interest is to establish a minimal set of capabilities the robots need to have to be able to perform a certain task, like forming a pattern. In this paper, we study the problem of gathering the robots at a point. This problem is known as Gathering Problem (or rendezvous, or point-formation problem) and is obviously one of the most primitive tasks that a set of robots might perform. The Gathering Problem has been studied intensively in the literature, in particular in the realm of distributed computing [2, 4, 5, 7, 8], but also in robotics [3] and artificial intelligence [6].

We study the Gathering Problem for a set of weak robots: the robots are anonymous (i.e., identical), they have no common coordinate system, and they have no means of direct communication. All robots operate individually, according to the following cycle: Initially, they are in a waiting state. They wake up independently and asynchronously, observe the other robots' positions, and compute a point in the plane. They start moving towards this points, but may not reach it (e.g. because of limits to the robot's motion energy). Then they become waiting again. Details of the model are given in Section 2. For these robots, the Gathering Problem is defined as follows:

Definition 1. *Given n robots $r_1, \ldots, r_n$, arbitrarily placed in the plane, with no two robots at the same position, make them gather at one point.*

If the robots are asked only to move "very close" to each other, this task is easily solved: each robot computes the center of gravity[1] of all robots, and moves towards it. However, in the GATHERING PROBLEM we ask the robots to meet at *exactly* one point.

If the robots are oblivious, i.e., if they do not remember previous observations and calculations, then the GATHERING PROBLEM is *unsolvable* [7, 8]. On the other hand, the problem can be solved if we change the nature of the robots: If we assume a common coordinate system, gathering is possible even with limited visibility [5]; if the robots are synchronous and movements are instantaneous, then the GATHERING PROBLEM has a simple solution [8] and can be achieved even with limited visibility [2]; finally, the problem can be solved for more than two robots if the robots can detect how many robots are at a certain point (*multiplicity detection*) [4]. Recently, the GATHERING PROBLEM was studied in the presence of faulty robots; assuming a strong model of synchronizity, then the non–faulty robots can gather if at most one third of the robots are faulty [1].

In this paper, we show that the GATHERING PROBLEM is solvable for $n \geq 2$ non–oblivious robots. First, we present in Section 4 an algorithm that gathers $n = 2$ robots. At the beginning, two robots move on a line ℓ, which connects their initial positions, away from each other. As soon as both robots have observed the configuration at least once (hence, they know ℓ), they start moving on lines perpendicular to ℓ until, again, both have seen both perpendicular lines. Finally, they meet on ℓ in the center between the two perpendicular lines.

For more than two robots, we distinguish in Section 5 how many robots are on the smallest enclosing circle *SEC* of the positions of all robots in the initial configuration. If there are more than two robots on *SEC*, then each robot moves on a circle around the center of *SEC* until all robots have seen *SEC*. Hereby, we use the fact that the smallest enclosing circle of the robots positions does not change. Then all robots gather at the center of *SEC*. On the other hand, if there are only two robots on *SEC*, then the robots that are not on *SEC* move perpendicular to the line ℓ connecting the two robots on *SEC*, while the robots on *SEC* move on line ℓ away from each other. The smallest enclosing circle increases, but ℓ remains invariant. As soon as all robots have seen line ℓ and the configuration, they gather at the intersection between ℓ and a line k, which is the median perpendicular line of the robots, if n is odd, or the center between the two median perpendicular lines, if n is even.

2 Autonomous Mobile Robots

A robot is a mobile computational unit provided with sensors, and it is viewed as a point in the plane. Once activated, the sensors return the set of all points in the plane occupied by at least one robot. This forms the current *local view* of the robot. The local view of each robot also includes a unit of length, an origin (which we assume w.l.o.g. to be the position of the robot in its current observation),

[1] For n points $p_1, \ldots, p_n$ in the plane, the center of gravity is $c = \frac{1}{n} \sum_{i=1}^{n} p_i$.

and a coordinate system (e.g. Cartesian). There is no a priori agreement among the robots on the unit of length, the origin, or the coordinate systems.

A robot is initially in a *waiting* state (*Wait*). Asynchronously and independently from the other robots, it *observes* the environment (*Look*) by activating its sensors. The sensors return a snapshot of the world, i.e., the set of all points that are occupied by at least one other robot, with respect to the local coordinate system. The robot then *calculates* its destination point (*Compute*) according to its deterministic algorithm (the same for all robots), based only on its local view of the world. It then *moves* towards the destination point (*Move*); if the destination point is the current location, the robot stays still. A move may stop before the robot reaches its destination. The robot then returns to the waiting state. The sequence *Wait* - *Look* - *Compute* - *Move* forms a *cycle* of a robot.

The robots are *fully asynchronous*, i.e., the amount of time spent in each state of a cycle is finite but otherwise unpredictable. In particular, the robots do not have a common notion of time. As a result, robots can be seen by other robots while moving, and thus computations can be made based on obsolete observations. The robots are *anonymous*, meaning that they are a priori indistinguishable by their appearance, and they do not have any kind of identifiers that can be used during the computation. Finally, the robots have *no means of direct communication*: any communication occurs in a totally implicit manner, by observing the other robots' positions.

There are two limiting assumptions concerning *infinity*: The amount of time required by a robot to complete a cycle is not infinite, nor infinitesimally small; and the distance traveled by a robot in a cycle is not infinite, nor infinitesimally small (unless it brings the robot to the destination point). As no other assumptions on space exist, the distance traveled by a robot in a cycle is unpredictable. All times and distances are under control of the adversary. We assume in our algorithms that the adversary is *fair*, in the sense that he respects the previous assumptions, and that no robot sleeps forever, since otherwise no algorithm can guarantee to gather the robots.

For the remainder of this paper, we assume that the robots are *non–oblivious*, meaning that each robot is equipped with infinite memory, and its computation in each cycle can be based on its observations and computation results from previous cycles.

3 Notation

In general, r indicates any robot in the system; when no ambiguity arises, r is used also to represent the point in the plane occupied by that robot. A *configuration* of the robots at a given time instant t is the set of positions in the plane occupied by the robots at time t.

We say that a point p is *on a circle* if it is on the circumference of the circle, and that p is *inside the circle* if it is strictly inside the circle. Given three distinct points p, q and c, we denote by $\sphericalangle(p, c, q)$ the *convex angle* (i.e., the angle that is

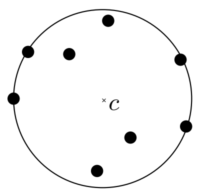

Fig. 1. Smallest enclosing circle *SEC* for 8 points.

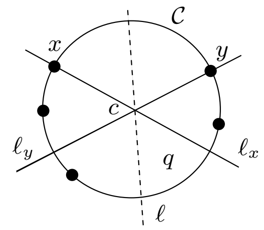

Fig. 2. Proof of Lemma 2. Center of *SEC* cannot be at q.

at most 180°) between p and q, centered in c. The Euclidean distant between p and q is denoted by $dist(p,q)$.

Given a set of n distinct points P in the plane, the *smallest enclosing circle* of the points is the circle with minimum radius such that all points from P are inside or on the circle (see Figure 1). We denote it by $SEC(P)$, or SEC if set P is unambiguous from the context. The smallest enclosing circle of a set of n points is unique and can be computed in polynomial time [9].

The smallest enclosing circle of P remains invariant if we move some of the points from P that are inside SEC such that they remain inside SEC; moreover, the maximum angle between any two adjacent points on SEC w.r.t. the center of SEC is 180°, since otherwise there would be a smaller circle enclosing all points. The following lemma shows that the smallest enclosing circle remains invariant even if we move the points along the rim of SEC, as long as no angle of more than 180° between adjacent points occurs.

Lemma 1. *Let $P = \{p_1, \ldots, p_k\}$ be k points on a circle C with center c. If the maximum angle between any two adjacent points w.r.t c is at most 180°, then C is the smallest enclosing circle of the points.*

Proof (sketch). The idea of the proof is as follows (cf. Figure 2): Assume that the center of $SEC(P)$ would be at some point $q \neq c$. Then there are two adjacent points $x, y \in P$ such that their angle w.r.t. c is minimum (and at most 180°), and such that q is within the sector of C that is beyond c and delimited by the lines ℓ_x and ℓ_y from x and y, respectively, through c (bottom sector in Figure 2). Let ℓ be the perpendicular line that bisects the angle between x and y (dashed line ℓ in Figure 2). If x and q are not on the same side of ℓ, then $dist(x,c) \leq dist(x,q)$; otherwise, y and q are not on the same side of ℓ, and $dist(y,c) \leq dist(y,q)$. In both cases, the radius of C is at most the radius of $SEC(P)$. Thus, since the smallest enclosing circle is unique, we have $C = SEC(P)$. □

4 Gathering Two Robots

In this section, we present an algorithm that solves the Gathering Problem for two robots. The idea of our algorithm, which is similar to the algorithm

Algorithm 1 Gathering two robots

If first observation **Then**
  $x_0 :=$ my position; $y_0 \leftarrow$ other robot's position;
  $\ell \leftarrow$ line through x and y; $d_0 :=$ distance between x and y;
  $state \leftarrow 1$; move on ℓ by distance $\frac{d_0}{100}$ away from y;
If $state = 1$ **Then**
  If other robot is at y_0 **Then** do nothing;
  Else
    $x_{perp} \leftarrow$ my position; $y_1 :=$ other robot's position;
    $d_1 :=$ distance between x_{perp} and y_1;
    If other robot is on ℓ **Then** $state \leftarrow 2$; move perpendicular to ℓ by $\frac{d_1}{100}$;
    Else
      $y_{perp} \leftarrow$ intersection between ℓ and line through other robots position perpendicular to ℓ; $d_{perp} \leftarrow$ distance between x_{perp} and y_{perp};
      $state \leftarrow 3$; move perpendicular to ℓ by distance $\frac{d_{perp}}{100}$;
If $state = 2$ **Then**
  If other robot is on ℓ **Then** do nothing
  Else
    $y_{perp} \leftarrow$ intersection between ℓ and line through other robots position perpendicular to ℓ; $d_{perp} \leftarrow$ distance between x_{perp} and y_{perp};
    $state \leftarrow 3$; do nothing;
If $state = 3$ **Then**
  If other robot is on the line perpendicular to ℓ through y_{perp} and less than d_{perp} away from ℓ **Then** move perpendicular to ℓ to distance d_{perp};
  Else
    $g \leftarrow$ center point between x_{perp} and y_{perp};
    $state \leftarrow 4$; do nothing;
If $state = 4$ **Then**
  If I am not at g **Then** move to g **Else** $state \leftarrow STOP$; do nothing;
End.

presented in [8], is as follows: The two robots move away from each other until both have seen the configuration at least once. Then they know the connecting line ℓ through their initial positions. In a next phase, they both move on lines that are perpendicuar to ℓ, again until both have seen the other robot at least once on its perpendicular line. Then they both know ℓ and its intersection with the two perpendicular lines, hence, they can gather on ℓ in the center between the perpendicular lines.

Lemma 2. *Two robots can gather at a point.*

Proof. Both robots perform Algorithm 1. Here, we use $\leftarrow$ to assign a value to a variable that is stored in the permanent memory of the robot (and is available in subsequent cycles), while we use $:=$ to assign values to variables that are only used in the current cycle.

We now prove that this algorithms gathers the two robots. Let r and s be the two robots. The following proofs are presented from the point of view of one robot r; analogous proofs yield the same propositions for the other robot. We denote

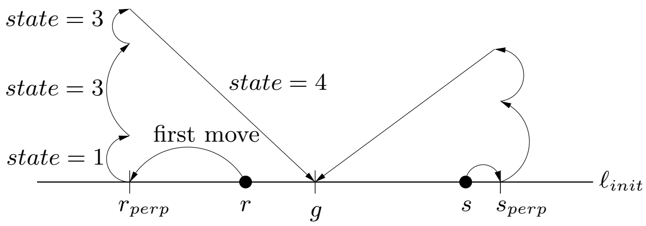

Fig. 3. Illustration of the algorithm for two robots. Distances are not drawn to scale.

the variables of robot r and s with superscript r and s, respectively. Let ℓ_{init} be the line through the initial positions of the robots, before any of the robots made its first movement. A schematic illustration of the robots' movements can be found in Figure 3.

1. If robot r is the first robot that leaves ℓ_{init}, then both robots agree on ℓ, i.e., $\ell^r = \ell^s = \ell_{init}$.

   Proof. If r leaves ℓ_{init} while s is still on the line, then r is in state 1 and $\ell^r = \ell_{init}$. Moreover, r has seen s in two different positions on ℓ_{init}, thus, s has moved on ℓ_{init} *before* r leaves ℓ_{init}. Hence, s has seen ℓ_{init} already, and we have $\ell^s = \ell_{init}$.

2. Both robots eventually leave ℓ_{init}.

   Proof. Assume that robot r wakes up first. Then it moves by $\frac{d_0^r}{100}$ and enters state 1. As soon as s has moved at least once, r moves away from ℓ_{init} by either $\frac{d_1^r}{100}$, if s is still on ℓ_{init}, or by $\frac{d_{perp}^r}{100}$, if s has left ℓ_{init}. Hence, as soon as r has observed the first movement of s, it leaves ℓ_{init}. If robot s has left ℓ_{init} at that time already, we are done. Otherwise, we know from Item 1 that s is in state 1, since it knows already ℓ_{init}, but it is still on ℓ_{init}. Hence, when s wakes up the next time, it observes that r has left ℓ_{init}, and s moves away from ℓ_{init} by $\frac{d_{perp}^s}{100}$.

3. Every subsequent movement of robot r after it left line ℓ_{init} is perpendicular away from ℓ_{init}, until it reaches state 4.

   Proof. When r moves away from ℓ_{init} for the first time, it is in state 1. By construction, this movement is perpendicular away from ℓ_{init}, starting in x_{perp}, by either distance $\frac{d_1^r}{100}$ or $\frac{d_{perp}^r}{100}$. Afterwards, robot r moves only if it is in state 3, and there the movements are by definition perpendicular to ℓ_{init}. It remains to show that r always moves *away* from ℓ_{init}. Since d_{perp}^r never changes, it is sufficient to show that $\frac{d_1^r}{100} < d_{perp}^r$. To see this, let d_{init} be the distance between the initial positions of the robots. When the robots wake up first, each of them makes one movement by at most $\frac{d_0^r}{100}$ and $\frac{d_0^s}{100}$, respectively, on ℓ_{init}, away from the other robot. Afterwards, all movements are perpendicular to ℓ_{init}. Hence, we have $d_{init} \leq d_1^r \leq d_{init} + \frac{d_0^r}{100} + \frac{d_0^s}{100}$.

With $d_0^r \leq d_{init} + \frac{d_0^s}{100}$ and $d_0^s \leq d_{init} + \frac{d_0^r}{100}$, straight–forward analysis shows that $d_1^r \leq \frac{10}{9} d_{init}$. This yields the claim, since d_{perp}^r is obviously greater than d_{init}.

4. Both robots eventually agree on point g, and gather there.

   Proof. Due to Item 2, both robots eventually leave line ℓ_{init}, say at positions r_{perp} and s_{perp}. Let r be the robot that leaves ℓ_{init} first. Then r stores value r_{perp} in x_{perp}^r, moves by $\frac{d_1^r}{100}$ away from ℓ_{init}, and enters state 2, where it remains until s will have left ℓ_{init}. When s wakes up the next time, it observes that r has left ℓ_{init}, and moves perpendicular away from ℓ_{init}, too. Moreover, it stores s_{perp} in x_{perp}^s, and r_{perp} in y_{perp}^s, since robot r has moved only perpendicular to ℓ_{init} due to Item 3. The next time robot r wakes up, it observes that s has left ℓ_{init}, too, and stores $y_{perp}^r = s_{perp}$ (again, since s moved only perpendicular to ℓ_{init}). Hence, both robots agree on the points r_{perp} and s_{perp} where they left ℓ_{init}, on distance d_{perp} between these points, and on the center point g. Moreover, both robots move on their perpendicular line until at least one of them, say s, has reached distance d_{perp} from ℓ_{init} (state 3). When this is observe by the other robot r, it enters state 4 and moves straight towards g, hence, r leaves its perpendicular line. When s wakes up the next time, it observes that r has left its perpendicular line, and s starts moving towards g, too. Eventually, both robots reach g and gather there.

□

5 Gathering $n > 2$ Robots

We now show how to gather more than two robots. We split the algorithm up into two separate cases, depending on the number of robots on the smallest enclosing circle *SEC* in the initial configuration: if there are at least three robots on *SEC*, we make all robots move on circles around the center of *SEC* until all robots know *SEC* (which does not change during the movements); then we gather the robots at the center of *SEC*. This is shown in the following Lemma 3. On the other hand, if there are exactly two robots on *SEC*, then we adapt the algorithm for two robots from Section 4 to gather all robots at the line connecting the two robots on *SEC*. This is shown in Lemma 4.

Lemma 3. *If there are more than* 2 *robots on the smallest enclosing circle in the initial configuration, then the robots can gather at a point.*

Proof. Given a configuration of the robots, we define a *movement angle* γ and a *movement direction moveDir* for each robot r as follows (cf. Figure 4): Let c be the center of the smallest enclosing circle of all robots. Let C be the circle with center c such that r is on C. If there is no other robot on C, then let $\gamma = \frac{1}{360n}$ and *moveDir* be an arbitrary direction on C, say clockwise. If there are exactly two robots on C, then let s be the other robot. [By assumption, C is not the

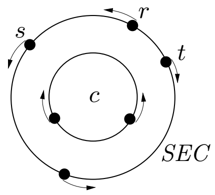

Fig. 4. Idea of Algorithm 2. Arrows indicate movement directions of the robots.

smallest enclosing circle of the robots.] Let α and β be the two angles between r and s w.r.t. c. Assume w.l.o.g. $\alpha \leq \beta$. Let $\gamma = \frac{\alpha}{360n}$, and let $moveDir$ be in the direction of angle β. If there are more than two robots on C, then let s and t be the two robots on C that are adjacent to r. Let α be the angle between r and s w.r.t. c, and β be the angle between r and t w.r.t c. Assume w.l.o.g. that $\alpha \leq \beta$. Then $\alpha < 180°$. If $\alpha \leq 178°$, then let $\gamma = \frac{\alpha}{360n}$ and $moveDir = t$. If $178° < \alpha$, then $\gamma = \frac{180°-\alpha}{360n}$ and $moveDir = t$.

If robot r observes the configation of all robots, it can order the other robots in a unique way, for instance by using the coordinates of the robots positions in the local coordinate system of robot r. We assume w.l.o.g. that robot r has index 1 in this ordering. Recall that different robots may have different coordinate systems, hence, the robots do not agree on this ordering. We will ensure in our algorithm that the basic configuration remains invariant; in particular, robots will stay on the same circle with center c, and no two robots on the same circle will interchange their position. Each robot stores the positions of all robots that it observes in its first cycle in an array $posns$, where $posns_j$ denotes the position of robot r_j. Hence, in later cycles robot r can compare the current position of a robot r_j with the position of r_j observed in its first cycle. This allows r to determine whether r_j has made at least one movement at some time. In addition, robot r maintains a vector $hasMoved$, such that $hasMoved_j$ is set to $true$ if r has observed at least once that robot r_j has moved.

The algorithm that the robots perform is shown in Algorithm 2, and an illustration can be found in Figure 4. We prove that the robots gather at c, the center of the smallest enclosing circle of the robots initial positions, by showing the following items:

1. Every robot makes at most n moves by angle γ in its direction $moveDir$.

   Proof. A robot only moves in direction $moveDir$ in states 2 and 3. If it is in state 2, then it moves once in direction $moveDir$, sets $hasMoved_1 = true$, and changes into state 3. In state 3, it moves in direction $moveDir$ if a value $hasMoved_j$ has changed from $false$ to $true$ (i.e., if another robot has moved). This can happen at most $n-1$ times, once for each other robot.

2. The angles between two adjacent robots on the same circle changes at most by 1°.

   Proof. We have $\gamma \leq \frac{180°}{360n}$ by definition, and each robot moves at most n times by its angle γ. Hence, the movement of a single robot changes the

Algorithm 2 Gathering with more than 2 robots on *SEC*

```
If this is my first observation Then
   n ← number of robots;
   SEC ← smallest enclosing circle of all robots; c ← center of SEC;
   If I am at c Then
      d := minimum distance of any other robot to c;
      state ← 2; move away from c by distance d/2;
   Else
      If some robot is at c Then state ← 1; do nothing;
      Else state ← 2; do nothing;
If state = 1 Then
   If a robot is at c Then do nothing;
   Else state ← 2; do nothing;
If state = 2 Then
   posns ← all robots positions, with posns_1 my own position;
   ∀j : hasMoved_j ← false; hasMoved_1 ← true;
   γ ← my movement angle; moveDir ← my movement direction;
   state ← 3; move by angle γ in direction moveDir;
If state = 3 Then
   If a robot decreased its distance from c Then state ← 4; do nothing;
   Else
      ∀j such that robot r_j changed its position w.r.t posns_j: hasMoved_j ← true;
      If ∀j : hasMoved_j = true Then state ← 4; do nothing;
      If at least one value hasMoved_j changed to true in this step Then
         move by angle γ in direction moveDir;
      Else do nothing;
If state = 4 Then
   If I am not at c Then move to c Else state ← STOP; do nothing;
End.
```

angle between itself and its neighbors by at most $n\gamma \leq \frac{1}{2}^{\circ}$. Thus, even if two adjacent robots move in opposite directions, the angle between them changes by at most 1°.

3. No two robots on the same circle interchange their position.

   Proof. Let v, w, x and y be adjacent robots (in this ordering) on the same circle with center c. Assume by contradiction that w and x interchange their positions. We show that this cannot happen even if w and x move towards each other. The other cases, where either x and w move in the same direction, or they move away from each other, can be shown analogous. If w and x move towards each other, then $\sphericalangle(w,c,x) > \sphericalangle(v,c,w)$ and $\sphericalangle(w,c,x) > \sphericalangle(x,c,y)$. By construction, we have $\gamma_w \leq \frac{\sphericalangle(w,c,x)}{360n}$: if $\sphericalangle(v,c,w) \leq 178^{\circ}$, then $\gamma_w = \frac{\sphericalangle(v,c,w)}{360n}$; on the other hand, if $\sphericalangle(v,c,w) > 178^{\circ}$, then $\gamma_w = \frac{180^{\circ} - \sphericalangle(w,c,x)}{360n} \leq \frac{\sphericalangle(w,c,x)}{360n}$. Analogously, $\gamma_x \leq \frac{\sphericalangle(w,c,x)}{360n}$. Robot w moves at most by angle $n \cdot \gamma_w$ towards x, and robot x moves at most by angle $n \cdot \gamma_x$ towards w (due to Item 1). Hence, the new angle between w and x is at least

$\sphericalangle(w,c,x) - n\gamma_w - n\gamma_x \geq \sphericalangle(w,c,x)(1 - \frac{1}{180}) > \frac{\sphericalangle(w,c,x)}{2} > 0°$, i.e., the two robots do not interchange their position.

4. SEC remains invariant until at least one robot has reached its state 4.

   Proof. Until some robots reach their state 4, all robots move on circles with center c. Hence, the smallest enclosing circle can only change if the maximum angle between the robots on SEC becomes larger than 180° (cf. Lemma 1). By previous Item 2, the angle between adjacent robots changes by at most 1°; thus, if all adjacent robots on SEC in the initial configuration have angle at most 178°, the smallest enclosing circle cannot change. If in the initial configuration there is exactly one angle between adjacent robots on SEC that is greater than 178°, say between robots x and y, then this is for both x and y the maximum adjacent angle. Hence, the moving direction of x is towards y by definition, and the moving direction of y is towards x. Thus, the angle between x and y decreases, and no angle of more that 180° can occur.
   For the case that there are 2 angles of more than 178°, first assume that there is no robot on SEC that has an angle of more than 178° to both neighbors (see Figure 5). Then there are two disjoint pairs of robots x, y and u, v such that the angle between x and y is greater than 178°, and the angle between u and v is greater than 178°. By construction, x moves towards y and y towards x, decreasing the angle between them. Likewise, u and v move towards each other. Hence, no angle greater than 180° occurs.
   Now assume that there is one robot r on SEC such that both angles α and β to its two neighbors s and t, respectively, are greater than 178° (see Figure 6). Assume that $\alpha \leq \beta$. Both s and t move towards r. By definition, the movement angle for robot r is $\gamma = \frac{180° - \alpha}{360n}$, and r moves towards t. Hence, the angle between r and t decreases. On the other hand, even if s does not move at all, and even if r moves by maximum angle $n\gamma$ towards t, then the new angle between s and r is at most $\alpha + n\gamma \leq 180°$. Hence, SEC does not change.

5. If a robot reaches its state 4, then all robots agree on SEC and c.

   Proof. A robot r reaches its state 4 only if $hasMoved_j^r = true$ for all $1 \leq j \leq n$. This yields the claim, since $hasMoved_j^r$ is set to $true$ only if robot r_j has made a move, i.e., if it was awake and had observed the configuration, including SEC and c.

6. At least one robot eventually reaches its state 4.

   Proof. Let r be the first robot to wake up. Then r observes the initial configuration of the robots. If there is a robot at c in the initial configuration, then this robot moves away from c in its first cycle. Afterwards, every robot that wakes up moves on its circle by its movement angle γ. Assuming a fair schedule where no robot sleeps for an infinite time, after some finite time every robot has woken up at least once. If some other robot but r reaches its state 4, then the claim is true. Otherwise, as soon as robot r wakes up

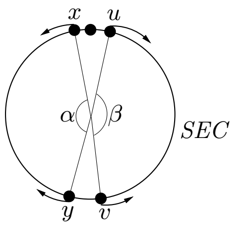

Fig. 5. Proof of Lemma 3, Item 4, for angles $\alpha, \beta > 178°$. Angles are not drawn to scale.

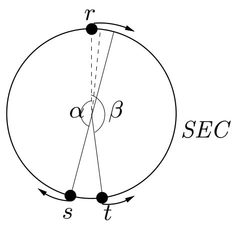

Fig. 6. Proof of Lemma 3, Item 4, for angles $\alpha, \beta > 178°$. Angle between the dashed lines is $n\gamma$. Angles are not drawn to scale.

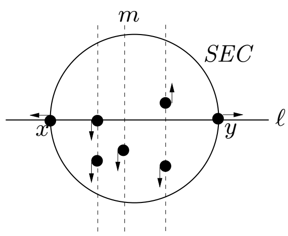

Fig. 7. Idea of algorithm for two robots on SEC. Line m is the median perpendicular line.

the next time, it observes that all other robots have moved since its first observation (i.e., $hasMoved_j^r = true$ for all $1 \leq j \leq n$), and r enters state 4.

7. All robots eventually reach their state 4.

   Proof. Due to Item 6, at least one robot r reaches its state 4. In its next cycle, this robot moves towards c, i.e., it decreases its distance from c. Hence, all other robots that wake up afterwards observe this decrease of the distance, and enter their state 4. Assuming a fair schedule where no robot sleeps forever yields the claim.

8. All robots gather at c and stop there.

   Proof. This is obvious, since all robots agree on c due to Item 5, all robots reach their state 4 due to Item 7, and each robot that is in state 4 moves towards c.

□

We now show how to solve the GATHERING PROBLEM if only two robots are on the smallest enclosing circle in the initial configuration.

Lemma 4. *If $n > 2$, and there are exactly 2 robots on the smallest enclosing circle in the initial configuration, then the robots can gather at a point.*

Proof (sketch). Let x and y be the two robots on smallest enclosing circle, and let ℓ be the line through x and y. Our algorithm works as follows (see Figure 7). First, all robots move "a little bit" until each robot has moved at least once. Here, both x and y move on ℓ away from each other. Every other robots r moves on a line perpendicular to ℓ, without reaching the next robot (if any) on the same line. The movement of x and y changes the smallest enclosing circle (in fact, it increases the radius of the circle), but x and y remain the only robots on this circle. Hence, each of the other robots moves always on the *same* line perpendicular to ℓ. As soon as all robots have made one move, they all know

ℓ and all perpendicular lines. If the number of robots n is odd, then all robots gather at the intersection of ℓ and the median perpendicular line. Otherwise, they gather at the intersection of ℓ and the center line between the two median perpendicular lines.

□

We summarize our result in the following theorem, which follows immediately from Lemmas 2, 3 and 4.

Theorem 1. *The* GATHERING PROBLEM *can be solved for* $n \geq 2$ *non–oblivious robots.*

6 Conclusion

We have presented an algorithm that gathers a set of n non–oblivious mobile robots. Thus, it is sufficient to equip the robots with memory to make the GATHERING PROBLEM become solvable. Moreover, our results indicates that memory is a more powerful capability than multiplicity detection, since we have shown that two robots with memory can gather, while two robots with multiplicity detection cannot [8].

Our algorithm makes generous use of memory, as it stores, among others, the exact positions of all robots. It would be interesting to see whether this could be significantly reduced. What is the minimum amount of memory necessary to solve the GATHERING PROBLEM?

References

1. N. Agmon and D. Peleg. Fault–tolerant gathering algorithms for autonomous mobile robots. In *ACM–SIAM Symposium on Discrete Algorithms (SODA 2004)*, to appear.
2. H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. A distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transaction on Robotics and Automation*, 15(5):818–828, 1999.
3. T. Balch and R. C. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transaction on Robotics and Automation*, 14(6):926–939, 1998.
4. M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. In *Proc. of the* 30^{th} *Intern. Colloquium on Automata, Languages and Programming (ICALP 2003)*, pages 1181–1196, 2003.
5. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of autonomous mobile robots with limited visibility. In *Proc. of the* 18^{th} *International Symposium on Theoretical Aspects of Computer Science (STACS 2001)*, pages 247–258, 2001.
6. M. J. Matarić. Designing emergent behaviors: From local interactions to collective intelligence. In *From Animals to Animats 2: Int. Conf. on Simulation of Adaptive Behavior*, pages 423–441, 1993.
7. G. Prencipe. *Distributed Coordination of a Set of Autonomous Mobile Robots*. PhD thesis, Università di Pisa, 2002.
8. I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *Siam Journal of Computing*, 28(4):1347–1363, 1999.
9. E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, pages 359–370. Springer, 1991.