# Self-stabilizing gathering with strong multiplicity detection 

Yoann Dieudonné ${ }^{\text {a,* }}$, Franck Petit ${ }^{\text {b }}$<br>${ }^{a}$ MIS, Université de Picardie Jules Verne, France<br>${ }^{\mathrm{b}}$ LIP6, UPMC Sorbonne Universités, France

## ARTICLE INFO

## Article history:

Received 5 January 2011
Received in revised form 6 December 2011
Accepted 9 December 2011
Communicated by D. Peleg

## Keywords:

Distributed coordination
Gathering
Mobile robot networks
Self-stabilization


#### Abstract

In this paper, we investigate the possibility to deterministically solve the gathering problem starting from an arbitrary configuration with weak robots, i.e., anonymous, autonomous, disoriented, oblivious, and devoid of means of communication. By starting from an arbitrary configuration, we mean that robots are not required to be located at distinct positions in the initial configuration. We introduce strong multiplicity detection as the ability for the robots to detect the exact number of robots located at a given position. We show that with strong multiplicity detection, there exists a deterministic algorithm solving the gathering problem starting from an arbitrary configuration for $n$ robots if, and only if, $n$ is odd.


© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

The distributed systems considered in this paper are teams (or swarms) of mobile robots (sensors or agents). Such systems supply the ability to collect (to sense) environmental data such as temperature, sound, vibration, pressure, motion, etc. Each robot interprets the information gathered from its sensors and acts in a given (sometimes dangerous) physical environment. Numerous potential applications exist for such multi-robot systems, e.g., environmental monitoring, largescale construction, risky area surrounding, exploration of an unknown area. All these applications involve basic cooperative tasks such as pattern formation, gathering, scatter, leader election, flocking, etc.

Among the above fundamental coordination tasks, we address the gathering (or Rendez-Vous) problem. This problem can be stated as follows: robots, initially located at various positions, gather at the same position in finite time and remain at this position thereafter. The difficulty to solve this problem greatly depends on the system settings, e.g., whether the robots can remember past events or not, their means of communication, their ability to share a global property like observable IDs, sense of direction, global coordinate, etc. For instance, assuming that the robots share a common global coordinate system or have (observable) IDs allowing to differentiate any of them, it is easy to come up with a deterministic distributed algorithm for that problem. Gathering turns out to be very difficult to solve with weak robots, i.e., devoid of (1) any (observable) IDs allowing to differentiate any of them (anonymous), (2) any central coordination mechanism or scheduler (autonomous), (3) any common coordinate mechanism or common sense of direction (disoriented), (4) means of communication allowing them to communicate directly, e.g., by radio frequency, and (5) any way to remember any previous observation nor computation performed in any previous step (oblivious).

With the same algorithm for all robots, each robot computes a destination point according to the snapshot of the positions of all other robots. Assuming that robots are points evolving on the plane, no deterministic algorithm exists for the gathering problem if the system contains two robots only [21]. The gathering problem has been extensively studied in the

[^0]literature assuming various settings. For instance, the robots move either in a discrete environment (i.e., among the nodes of a graph) [12,15], or in the plane [1,2,5,13,16,17,21], their visibility can be limited (visibility sensors are supposed to be accurate within a constant range, and sense nothing beyond this range) [13,19], robots are prone to faults [1,7], the robots gather with some probability (i.e., they gather in an expected finite time) by using randomization [6,7,14], etc.

A deterministic system is self-stabilizing if, regardless of the initial states of the computing units, it is guaranteed to converge to the intended behavior in a finite number of steps [11]. Self-stabilization is a very desirable property for modern distributed systems in order to design distributed algorithms that withstand transient faults. In addition to fault-tolerance, self-stabilizing systems encompass an other advantage that is the absence of initialization phase.

The gathering problem (as many other basic tasks related to robot coordination) is quite close to be a self-stabilizing task in its statement. However, the gathering problem was always tackled in a deterministic way assuming that in the initial configuration, no two robots are located at the same position. (A probabilistic solution with no restriction on the initial configuration is proposed in [9].) As a matter of fact, none of the existing deterministic solutions for the gathering problem works if some robots share the same positions in the initial configuration. So, effectively, as already noticed in [8,9], this implies that none of them is "truly" self-stabilizing because some initial configurations (where robots are located at the same positions) are avoided. Notice that surprisingly, such a restriction prevents to initiate the system in a configuration where the problem is already solved-i.e., initially all the robots occupy the same position.

In this paper, we investigate the self-stabilizing gathering problem that is, gathering the robots deterministically with no kind of restriction on the initial configuration. In particular, robots are allowed to share same positions in the initial configuration. In [17], Prencipe shows that the gathering problem cannot be deterministically solved if the robots are devoid of an extra capability, called multiplicity detection, i.e., each robot is able to know whether zero, one, or more than one robots are located at position on the plane. As a matter of fact, it is easy to show that the multiplicity detection as above stated-in the sequel referred to as weak multiplicity detection-is not a sufficient capability allowing weak robots to deterministically solve self-stabilizing gathering. Indeed, since the initial configuration is arbitrary, no matter the number of robots, all the robots can be initially located at exactly two positions. As we consider deterministic solutions only, the system may behave as if it contains exactly two robots, leading to the impossibility result in [21].

Let us introduce strong multiplicity detection as the ability for the robots to count the exact number of robots located at a given position. ${ }^{1}$ Again, it is easy to show that even with such capability, the problem cannot be solved deterministically, if the number of robots is even. The proof is similar as above: if initially the robots occupy exactly two positions, then there is no way to maintain a particular position as an invariant. Again, the impossibility result in [21] holds. By contrast, we show that with an odd number of robots, the problem becomes solvable. Our proof is constructive, as we present and prove a deterministic self-stabilizing algorithm for the gathering problem.

In the next section (Section 2), we describe the distributed system, the problem considered in this paper and some basic geometric requirements. Our main result with its proof is given in Section 3. We conclude this paper in Section 4.

## 2. Preliminaries

In this section, we define the distributed system followed by the problem considered in this paper. Basic geometric requirements are also provided.

### 2.1. Distributed model

We adopt the semi-synchronous model introduced in [20], below referred to as SSM. The distributed system considered in this paper consists of $n$ robots $r_{1}, r_{2}, \ldots, r_{n}$-the subscripts $1, \ldots, n$ are used for notational purpose only. Each robot $r_{i}$, viewed as a point in the Euclidean plane, moves on this two-dimensional space unbounded and devoid of any landmark. It is assumed that two or more robots may simultaneously occupy the same physical location.

Any robot can observe, compute and move with infinite decimal precision. The robots are equipped with sensors enabling to detect the instantaneous position of the other robots in the plane. In particular, we assume that the robots are able to sense the number of robots located at a given position. More formally:
Definition 1 (Strong Multiplicity Detection). The robots have strong multiplicity detection if, for every point $p$, their sensors can detect the number of robots at $p$.

Each robot has its own local coordinate system and unit measure. The robots do not agree on the orientation of the axes of their local coordinate system, nor on the unit measure. They are uniform and anonymous, i.e., they all have the same program using no local parameter such that an (observable) identity allowing to differentiate any of them. They communicate only by observing the position of the others and they are oblivious, i.e., none of them can remember any previous observation nor computation performed in any previous step. In the sequel, such uniform, anonymous, disoriented, and oblivious robots are said to be weak.

[^1]Time is represented as an infinite sequence of time instants $0,1, \ldots, j, \ldots$ Let $\mathcal{P}(t)$ be the set of the positions in the plane occupied by the $n$ robots at time $t$. For every $t, \mathcal{P}(t)$ is called the configuration of the distributed system in $t$. Given any point $p,|p|$ denotes the number of robots located at $p$. At each time instant $t$, each robot $r_{i}$ is either active or inactive. The former means that, during the computation step $(t, t+1)$, using a given algorithm, $r_{i}$ computes in its local coordinate system a position $p_{i}(t+1)$ depending only on the system configuration at $t$, and moves toward $p_{i}(t+1)-p_{i}(t+1)$ can be equal to $p_{i}(t)$, making the location of $r_{i}$ unchanged. In the latter case, $r_{i}$ does not perform any local computation and remains at the same position. In every single activation, the distance traveled by any robot $r$ is bounded by $\sigma_{r}$. So, if the destination point computed by $r$ is farther than $\sigma_{r}$, then $r$ moves toward a point of at most $\sigma_{r}$. This distance may be different between two robots. Given a cohort of $n$ robots $\left\{r_{1}, r_{2}, \ldots, r_{n}\right\}$, we denote by $\epsilon$ the distance such that $\epsilon=\min \left(\sigma_{r_{1}}, \sigma_{r_{2}}, \ldots, \sigma_{r_{n}}\right)$.

The concurrent activation of robots is modeled by the interleaving model in which the robot activations are driven by a fair scheduler. At each time instant $t$, the scheduler arbitrarily activates a (nonempty) set of robots. Fairness means that every robot is infinitely often activated by the scheduler.

### 2.2. Specification

In the literature, the gathering problem is tackled assuming that initially, no two robots are located at the same position (or node, if the environment is a graph). In this paper, we address the problem without any restriction on the initial configuration. In other words, we aim to provide an algorithm able to work starting from any arbitrary configuration, i.e., even if initially some robots share the same locations.
Definition 2 (Self-Stabilizing Gathering Problem). The Self-Stabilizing Gathering Problem is to design a (deterministic) distributed protocol (or, algorithm) $P$ for $n$ mobile robots so that, in every execution, the following properties are true:

- Convergence: regardless of the initial positions of the robots on the plane, all the robots gather at one point in finite time.
- Closure: starting from a configuration where all the robots are located at the same position, all the robots remain at the same position thereafter.

Note that a probabilistic solution could not fit Definition 2. Indeed, convergence and closure together impose that the protocol terminates in finite time. Any probabilistic solution only ensures that the protocol terminates in an expected finite time.

### 2.3. Notations, basic definitions and properties

Given a configuration $\mathcal{P}, \operatorname{Max} \mathscr{P}$ indicates the set of all the points $p$ such that $|p|$ is maximal. In other terms, $\forall p_{i} \in \operatorname{Max} \mathcal{P}$ and $\forall p_{j} \in \mathcal{P}$, we have $\left|p_{i}\right| \geq\left|p_{j}\right| .|\operatorname{Max} \mathcal{P}|$ will be the cardinality of Max $\mathcal{P}$.

Remark 1. Since the robots have the strong multiplicity detection, then they are able to compute $|p|$ for every point $p \in \mathcal{P}$. In particular, all the robots can determine $\operatorname{Max} \mathcal{P}(t)$ at each time instant $t$.

Given three distinct points $r, r^{\prime}$ and $c$ in the plane, we say that the two half-lines $[c, r)$ and $\left[c, r^{\prime}\right]$ divide the plane into two sectors if and only if:

- either $r, r^{\prime}$ and $c$ are not collinear,
- or $r, r^{\prime}$ and $c$ are collinear and $c$ is between $r$ and $r^{\prime}$ on the segment $\left[r, r^{\prime}\right]$.

If it exists then this pair of sectors is denoted by $\left\{\Delta\left(r c r^{\prime}\right), \nabla\left(r c r^{\prime}\right)\right\}$ and we assume that the two half-lines [ $c, r$ ) and [ $c, r^{\prime}$ ) do not belong to any sector in $\left\{\Delta\left(r c r^{\prime}\right), \nabla\left(r c r^{\prime}\right)\right\}$. Note that, if the three points $r, r^{\prime}$, and $c$ are not collinear then one of two sectors is convex (angle centered at $c$ between $r$ and $r^{\prime} \leq 180^{\circ}$ ) and the other one is concave (angle centered at $c$ between $r$ and $r^{\prime}>180^{\circ}$ ). Otherwise, the three points $r, r^{\prime}$, and $c$ are collinear and the two sectors are convex. More precisely, they are straight (both conjugate angles centered at $c$ between $r$ and $r^{\prime}$ are equal to $180^{\circ}$ ).

Definition 3 (Smallest Enclosing Circle). [8] Given a set $\mathcal{P}$ of $n \geq 2$ points $p_{1}, p_{2}, \ldots, p_{n}$ on the plane, the smallest enclosing circle of $\mathcal{P}$, called $\operatorname{SEC}(\mathcal{P})$, is the smallest circle enclosing all the positions in $\mathcal{P}$. It passes either through two of the positions that are on the same diameter (opposite positions), or through at least three of the positions in $\mathcal{P}$.

When no ambiguity arises, $\operatorname{SEC}(\mathcal{P})$ will be shortly denoted by $\operatorname{SEC}$ and $\operatorname{SEC}(\mathcal{P}) \cap \mathcal{P}$ will indicate the set of all the points both on $\operatorname{SEC}(\mathscr{P})$ and $\mathscr{P}$. Besides, we will say that a robot $r$ is inside SEC if, and only if, $r$ is not located on the circumference of SEC. In any configuration $\mathscr{P}, S E C$ is unique and can be computed in linear time [3].

Given a set $\mathcal{P}$ of $n \geq 2$ points $p_{1}, p_{2}, \ldots, p_{n}$ on the plane and $\operatorname{SEC}(\mathcal{P})$ its smallest enclosing circle, $\mathscr{R} \operatorname{ad}(\operatorname{SEC}(\mathcal{P}))$ will indicate the length of the radius of $\operatorname{SEC}(\mathcal{P})$.

The next lemma states a simple fact:
Lemma 1. Let $\mathcal{P}_{1}$ be an arbitrary configuration of $n$ points. Let $\mathcal{P}_{2}$ be a configuration obtained by pushing inside $\operatorname{SEC}\left(\mathcal{P}_{1}\right)$ all the points which are in $\mathcal{P}_{1} \cap \operatorname{SEC}\left(\mathscr{P}_{1}\right)$. We have $\operatorname{Rad}\left(\operatorname{SEC}\left(\mathcal{P}_{2}\right)\right)<\operatorname{Rad}\left(\operatorname{SEC}\left(\mathcal{P}_{1}\right)\right)$.


Fig. 1. $C_{2}$ is an enclosing circle for the three points $r_{i}, r_{j}$ and $r_{k}$. However, there is no point in the intersection between $C_{2}$ and the concave sector formed by $r_{i}, r_{j}$ and the center $c$ of $C_{2}$. So, $C_{2}$ can be replaced by a smaller enclosing circle, here $C_{1}$, even if all the points are on the circumference of $C_{2}$.

Let $s$ and $C$ be a sector in $\left\{\Delta\left(p c p^{\prime}\right), \nabla\left(p c p^{\prime}\right)\right\}$ and a circle centered at $c$, respectively. We denote by $\operatorname{arc}(C, \delta)$ the arc of the circle $C$ inside $s$. Given a set $\mathcal{P}$ of $n \geq 2$ points $p_{1}, p_{2}, \ldots, p_{n}$ on the plane and $\operatorname{SEC}(\mathscr{P})$ its smallest enclosing circle centered at $c$, we say that $p$ and $p^{\prime}$ are adjacent on $\operatorname{SEC}(\mathcal{P})$ if, and only if, $p$ and $p^{\prime}$ are in $\mathcal{P}$ and there exists one sector $s \in\left\{\Delta\left(p c p^{\prime}\right), \nabla\left(p c p^{\prime}\right)\right\}$ such that there is no point in $\operatorname{arc}(\operatorname{SEC}(\mathcal{P}), \delta) \cap \mathcal{P}$.

The following property is fundamental about smallest enclosing circles:
Property 1. [4] Let $\mathcal{P}$ and $c$ be a set of $n \geq 2$ points $p_{1}, p_{2}, \ldots, p_{n}$ in the plane and the center of $\operatorname{SEC}(\mathcal{P})$, respectively. If $p$ and $p^{\prime}$ are adjacent on $\operatorname{SEC}(\mathcal{P})$ then, there does not exist a concave sector $s$ in $\left\{\Delta\left(p c p^{\prime}\right), \nabla\left(p c p^{\prime}\right)\right\}$ such that there is no point in $\operatorname{arc}(S E C(\mathcal{P}), \ngtr) \cap \mathcal{P}$.

Property 2 is more general than Property 1 :
Property 2. Let $\mathcal{P}$ and $c$ be a set of $n \geq 2$ points $p_{1}, p_{2}, \ldots, p_{n}$ in the plane and the center of $\operatorname{SEC}(\mathcal{P})$, respectively. If $p$ and $p^{\prime}$ are in $\mathcal{P}$ then, there does not exist a concave sector $s$ in $\left\{\Delta\left(p c p^{\prime}\right), \nabla\left(p c p^{\prime}\right)\right\}$ such that there is no point in $\mathcal{S} \cap \mathcal{P}$.
Proof. Assume by contradiction that $p$ and $p^{\prime}$ are in $\mathcal{P}$ and, there exists a concave sector $\&$ in $\left\{\Delta\left(p c p^{\prime}\right), \nabla\left(p c p^{\prime}\right)\right\}$ such that there is no point in $\delta \cap \mathcal{P}$. So, there is no point in $\operatorname{arc}(\operatorname{SEC}(\mathcal{P}), \delta) \cap \mathscr{P}$. We deduce that there exists a concave sector $s^{\prime}$ in $\left\{\Delta\left(q c q^{\prime}\right), \nabla\left(q c q^{\prime}\right)\right\}$ such that $q$ and $q^{\prime}$ are adjacent on $\operatorname{SEC}(\mathcal{P})$ and there is no point $\operatorname{arc}\left(\operatorname{SEC}(\mathcal{P}), s^{\prime}\right) \cap \mathcal{P}$. This contradicts Property 1.

Fig. 1 illustrates Property 2.
Observation 1. Given three collinear points, $c, r$, and $r^{\prime}$. If $c$ is on the segment $\left[r, r^{\prime}\right]$, then $c$ cannot be on the circumference of $a$ circle enclosing $r$ and $r^{\prime}$.
Definition 4 (Convex Hull). [18] Given a set $\mathcal{P}$ of $n \geq 2$ points $p_{1}, p_{2}, \ldots, p_{n}$ on the plane, the convex hull of $\mathcal{P}$, denoted $H(\mathcal{P})$, is the smallest polygon such that every point in $\mathscr{P}$ is either on an edge of $H(\mathcal{P})$ or inside it.

Informally, convex hull can be seen as the shape of a rubber-band stretched around $p_{1}, p_{2}, \ldots, p_{n}$. The convex hull is unique and can be computed with time complexity $O(n \log n)$ [18]. When no ambiguity arises, $H(\mathscr{P})$ will be shortly denoted by $H$ and $H(\mathcal{P}) \cap \mathscr{P}$ will indicate the set of the positions both on $H(\mathcal{P})$ and $\mathscr{P}$.

From Definition 4, we deduce the following property:
Property 3. Let $\mathcal{P}$ be a set of $n \geq 2$ points that are not on the same line and let $H(\mathcal{P})$ be a convex hull, respectively. The two following properties are equivalent:

1. Any point $c$, not necessarily in $\mathcal{P}$, is located on $H$ (either on a vertex or an edge);
2. there is a concave or a straight sector $\mathcal{s}$ in $\left\{\Delta\left(r c r^{\prime}\right), \nabla\left(r c r^{\prime}\right)\right\}$ such that $r$ and $r^{\prime}$ are in $\mathscr{P}$ and there exists no point $\in \mathscr{P} \cap s$.

## 3. Self-stabilizing gathering

In this section, we prove the following theorem:
Theorem 1. With strong multiplicity detection, there exists a deterministic algorithm solving the self-stabilizing gathering problem in SSM for $n$ weak robots if, and only if, $n$ is odd.

As mentioned in the introduction, even with strong multiplicity detection, no deterministic algorithm exists solving the self-stabilizing gathering problem for an even number of robots. So, to prove Theorem 1, we give a deterministic algorithm that solves the self-stabilizing gathering problem for an odd number of robots having the strong multiplicity detection. The algorithm is followed by its correctness proof.

### 3.1. A deterministic self-stabilizing algorithm for an odd number of robots

The main idea of our algorithm is as follows: it consists in transforming an arbitrary configuration $\mathcal{P}$ into one where there is exactly one point $p_{\max } \in \operatorname{Max} \mathcal{P}$. When such a configuration is reached, all the robots which are not located at $p_{\max }$ move toward $p_{\max }$ avoiding to create another point $q$ than $p_{\max }$ such that $|q| \geq p_{\max }$. When $|\operatorname{Max} \mathcal{P}| \neq 1$, we will distinguish two cases: $|\operatorname{Max} \mathcal{P}|=2$ and $|\operatorname{Max} \mathcal{P}| \geq 3$.

If $\operatorname{Max} \mathcal{P}=\left\{p_{\max 1} ; p_{\max 2}\right\}$, then each robot which is not neither at $p_{\max 1}$ nor at $p_{\max 2}$ moves toward its closest position $\in \operatorname{Max} \mathcal{P}$ by avoiding to create an adding maximal point. Since the number of robots is odd, we have eventually either $\left|p_{\max 1}\right|>\left|p_{\max 2}\right|$ or $\left|p_{\max 2}\right|>\left|p_{\max 1}\right|$ and then, $|\operatorname{Max} \mathcal{P}|=1$.

For the case $|\operatorname{Max} \mathcal{P}| \geq 3$, our strategy consists in trying to create a unique maximal point inside $S E C$. To reach such a configuration, we distinguish three cases:

1. If there is no robot inside $S E C$, then all the robots are allowed to move toward the center of SEC.
2. If all the robots inside SEC are located at the center of $S E C$, then only the robots in $S E C \cap$ Max $\mathcal{P}$ are allowed to move toward the center of SEC.
3. If some robots inside SEC are not located at the center of SEC, then only the robots inside SEC are allowed to move toward the center of SEC.

The algorithm is shown in Algorithm 1. It uses two subroutines:

- move_to_carefully $(p)$ allows a robot $r$, located at $q$, to move toward $p$ only if there is no robot on the segment $[q, p]$, except the robots located at $p$ or at $q$.
- choose_closest_position $\left(p_{1}, p_{2}\right)$ returns the closest position to $r$ among $\left\{p_{1}, p_{2}\right\}$. If the distance between $r$ and $p_{1}$ is equal to the distance between $r$ and $p_{2}$, then the function returns $p_{1}$.

```
Algorithm 1 Gathering for an odd number of robots, executed by each robot.
    \(\mathcal{P}:=\) the set of all the positions;
    \(\operatorname{Max} \mathcal{P}:=\) the set of all the points \(p \in \mathcal{P}\) such that \(|p|\) is maximal;
    if \(|\operatorname{Max} \mathcal{P}|=1\)
    then \(p_{\text {max }}:=\) the unique point in \(\operatorname{Max} \mathcal{P}\);
        if I am not on \(p_{\text {max }}\);
        then move_to_carefully \(\left(p_{\max }\right)\)
        endif
    endif
    if \(|\operatorname{Max} \mathcal{P}|=2\)
    then \(p_{\max 1}:=\) the first point in \(\operatorname{Max} \mathcal{P}\);
        \(p_{\max 2}:=\) the second point in \(\operatorname{Max} \mathcal{P}\);
        if I am not neither on \(p_{\max 1}\) nor \(p_{\text {max2 }}\)
        then \(q:=\) choose_closest_position \(\left(p_{\max 1}, p_{\max 2}\right)\);
            move_to_carefully \((q)\);
        endif
endif
if \(|\operatorname{Max} \mathcal{P}| \geq 3\)
then SEC := the smallest circle enclosing all the points in \(\mathcal{P}\);
        \(c:=\) the center of \(S E C\)
        Boundary : \(=S E C \cap \mathcal{P}\);
        Inside \(:=\mathcal{P} \backslash\) Boundary;
        if Inside \(\neq \emptyset\)
        then if All the robots \(\in\) Inside are located at \(c\)
            then if I am in (Boundary \(\cap \operatorname{Max} \mathcal{P}\) )
                then move_to(c);
                endif
            else if I am in Inside
                then move_to(c);
                endif
            endif
        else move_to(c)
        endif
    endif
```


### 3.2. Proof of closure

Lemma 2 (Closure). According to Algorithm 1, if all the robots are located at the same position p, then all the robots remain at the same position thereafter.

Proof. If all the robots are located at the same position $p$, then $|\operatorname{Max} \mathcal{P}|=1$ and all the robots are at the unique position $p \in \operatorname{Max} \mathcal{P}$. According to Algorithm 1, in the case $|\operatorname{Max} \mathcal{P}|=1$ the robots located on $p$ remains idle. So, all the robots remain at the same position forever.

### 3.3. Proof of convergence

Cases $|\operatorname{Max} \mathcal{P}|=1$ and $|\operatorname{Max} \mathcal{P}|=2$.
Lemma 3. Let $\mathcal{P}$ be an arbitrary configuration for an odd number of $n$ robots. According to Algorithm 1, if $|\operatorname{Max} \mathcal{P}|=1$ then all the robots gather at the same position in finite time.

Proof. Let $p_{\max }$ be the unique point in $\operatorname{Max} \mathcal{P}(t)$. According to Algorithm 1, the robots located at $p_{\max }$ during step $(t, t+1)$ remains idle. Moreover, according to Algorithm 1 and Function move_to_carefully(), if two robots $r_{i}$ and $r_{j}$ are not at the same point at time $t\left(p_{i}(t) \neq p_{j}(t)\right)$, then $p_{i}(t+1) \neq p_{j}(t+1)$ at time $t+1$ unless they have reached $p_{\text {max }}$. Hence, $p_{\text {max }}$ remains the unique point in $\operatorname{Max} \mathcal{P}\left(t_{k}\right)$, for all $t_{k} \geq t$. So, according to Algorithm 1 and by fairness, we deduce that $\left|p_{\max }\right|=n$ in finite time.

Lemma 4. Let $\mathcal{P}$ be an arbitrary configuration for an odd number of $n$ robots. According to Algorithm 1, if $|\operatorname{Max} \mathcal{P}|=2$ then all the robots gather at the same position in finite time.

Proof. The proof is organized as follows: first, we prove that there exists $t_{k} \geq t$ such that $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right| \neq 2$. Then, we prove that there does not exist any time $t_{k} \geq t$ such that $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right| \geq 3$. Finally, we deduce that Lemma 3 holds.

1. Assume by contradiction that no time $t_{k} \geq t$ exists such that $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right| \neq 2$. Consequently, for every $t_{k} \geq t$, $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right|=2$. Let $p_{\max 1}$ and $p_{\max 2}$ be the two points in $\operatorname{Max} \mathcal{P}(t)$ at time $t$. According to Algorithm 1, every robot located either on $p_{\max 1}$ or on $p_{\max 2}$ during step $(t, t+1)$ remains idle. Moreover, according to Algorithm 1 and Function move_to_carefully () , if two robots $r_{i}$ and $r_{j}$ are not at the same point at time $t$, i.e., $p_{i}(t) \neq p_{j}(t)$ then $p_{i}(t+1) \neq p_{j}(t+1)$ at time $t+1$ unless either $r_{i}$ and $r_{j}$ have reached $p_{\max 1}$ or $r_{i}$ and $r_{j}$ have reached $p_{\max 2}$. So, by induction we deduce that $p_{\max 1}$ and $p_{\max 2}$ remain the only positions in $\operatorname{Max} \mathcal{P}\left(t_{k}\right)$ for every $t_{k} \geq t$. By fairness, we deduce that, all the robots are either at $p_{\max 1}$ or at $p_{\max 2}$ in finite time. However, since the number of robots is odd then, we have either $\left|p_{\max 1}\right|>\left|p_{\max 2}\right|$ or $\left|p_{\max 1}\right|<\left|p_{\max 2}\right|$. Hence, $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right|=1$. A contradiction.
2. Assume by contradiction that there exists $t_{k} \geq t$ such that $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right| \geq 3$. Without lost of generality, we assume that $t_{k}$ is the first time for which $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right| \geq 3$. Clearly, there exists no time $t_{l}$ such that $t<t_{l}<t_{k}$ and $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right|=1$ : Indeed from Lemma 2 and the proof of Lemma 3, once there exists a unique point $p_{\text {max }}$ then, it remains the unique point in $\operatorname{Max} \mathcal{P}$ forever and that would be a contradiction.

Hence, $\left|\operatorname{Max} \mathcal{P}\left(t_{k}-1\right)\right|=2$.
Let $p_{\max 1}$ and $p_{\max 2}$ be the two points in $\operatorname{Max} \mathcal{P}\left(t_{k}-1\right)$ at time $t_{k}-1$. According to Algorithm 1 , the robots located either on $p_{\max 1}$ or on $p_{\max 2}$ during step $(t, t+1)$ remains idle. Besides, according to Algorithm 1 and Function move_to_carefully(), if two robots $r_{i}$ and $r_{j}$ are not at the same point at time $t_{k}-1\left(p_{i}\left(t_{k}-1\right) \neq p_{j}\left(t_{k}-1\right)\right)$, then $p_{i}(k) \neq p_{j}(k)$ at time $t_{k}$ unless either $r_{i}$ and $r_{j}$ have reached $p_{\max 1}$ or $r_{i}$ and $r_{j}$ have reached $p_{\max 2}$. So, $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right| \leq 2$ at time $t_{k}$. A contradiction.
From above, we deduce that if $|\operatorname{Max} \mathcal{P}(t)|=2$ at time $t$ then, according to Algorithm 1 there exists $t_{k}, t_{k}>t$ such that $\left|\operatorname{Max} \mathcal{P}\left(t_{k}\right)\right|=1$. So, from Lemma 3, we know that all the robots will gather at the same position in finite time.

Case $|\operatorname{Max} \mathcal{P}| \geq 3$. From now on, we prove that starting from a configuration where $|\operatorname{Max} \mathscr{P}| \geq 3$, all the robots gather at the same position in finite time.

In order to prove Lemma 8, we use Lemmas 5-7. In particular, Lemma 5 shows that, under specific conditions, the center of $\operatorname{SEC}(\mathcal{P}(t))$ is inside $\operatorname{SEC}(\mathcal{P}(t+1))$ even if $\operatorname{SEC}(\mathcal{P}(t)) \neq \operatorname{SEC}(\mathcal{P}(t+1))$ or the center of $\operatorname{SEC}(\mathcal{P}(t))$ is not the center of $\operatorname{SEC}(\mathcal{P}(t+1))$.

Lemma 5. Let $\mathcal{P}(t)$ be a configuration such that $|\operatorname{Max} \mathcal{P}(t)| \geq 3$ and there exists at least one robot inside $\operatorname{SEC}(\mathcal{P}(t))$. Let $c$ be the center of $\operatorname{SEC}(\mathcal{P}(t))$.

According to Algorithm 1, at least one of these properties is true:

1. $|\operatorname{Max} \mathcal{P}(t+1)| \leq 2$.
2. The center $c$ of $\operatorname{SEC}(\mathcal{P}(t))$ is inside $\operatorname{SEC}(\mathcal{P}(t+1))$ at time $t+1$ (even if $c$ is not the center of $\operatorname{SEC}(\mathscr{P}(t+1))$ ).

Proof. Let us consider the two following cases:

1. There exists at least one robot inside $\operatorname{SEC}(\mathcal{P}(t))$ that is not located at $c$ at time $t$. According to Algorithm 1 , only the robots inside $\operatorname{SEC}(\mathscr{P}(t))$ are allowed to move at time $t$. So, the robots on the circumference of $S E C$ remains so at time $t$. Moreover, according to Algorithm 1, no robot inside $\operatorname{SEC}(\mathcal{P}(t))$ can move outside of $\operatorname{SEC}(\mathcal{P}(t))$ at time $t$. So, $\operatorname{SEC}(\mathcal{P}(t))=\operatorname{SEC}(\mathcal{P}(t+1))$ and thus, Lemma 5 holds for this case. Note that, due to the motion of the robots inside SEC, it may also be possible that $|\operatorname{Max} \mathcal{P}(t+1)| \leq 2$.


Fig. 2. The numbers between parenthesis indicate the multiplicity. Figure $a$ describes a configuration $\mathcal{P}(t)$, where the center $c$ of $\operatorname{SEC}(\mathcal{P}(t))$ is inside the convex hull. Figure $b$ describes $\mathcal{P}(t+1)$, where some robots have moved toward $c$, and where $c$ is inside the new convex hull.
2. All the robots inside $S E C(\mathcal{P}(t))$ are located at $c$ at time $t$. According to Algorithm 1, only the robots on a maximal point on $\operatorname{SEC}$, i.e., in $\operatorname{Max} \mathcal{P}(t) \cap \operatorname{SEC}(\mathcal{P}(t))$, are allowed to move. Clearly, if none of these robots move then Lemma 5 holds. That is why, in the rest of this proof, we assume that at least one robot in $\operatorname{Max} \mathcal{P}(t) \cap \operatorname{SEC}(\mathcal{P}(t))$ moves. The, we need to consider the two following sub-cases:
(a) $\exists p \in \operatorname{Max} \mathcal{P}(t) \cap \operatorname{SEC}(\mathcal{P}(t))$ such that all the robots in $p$ reach $c$ at time $t+1$. Since there is at least one robot at $c$ at time $t$, the number of robots located at $c$ at time $t+1$, denoted by $|c(t+1)|$ is strictly greater than the number of robots located on the maximal point $p$ at time $t$, denoted by $|p(t)|$. So, we have $|c(t+1)|>|p(t)|$. Furthermore, since all the robot inside SEC at time $t$ are located at $c$ and since the robots move in straight line toward $c$ at time $t$, two robots coming from two distinct maximal positions on SEC cannot occupy the same point at time $t+1$ except at $c$. So, it is impossible to create another point $p^{\prime}$ at time $t+1$ such that $\left|p^{\prime}(t+1)\right|>|p(t)|$. We deduce that $c$ is the unique maximal point at time $t+1$, i.e., $|\operatorname{Max} \mathcal{P}(t+1)|=1$ and thus, Lemma 5 is true.
(b) $\forall p \in \operatorname{Max} \mathcal{P}(t) \cap \operatorname{SEC}(\mathcal{P}(t))$ there is at least one robot on $p$ that does not reach $c$ at time $t+1$. Depending on whether $c$ is on the convex hull $H(\mathcal{P}(t))$ or not, at time $t$, we consider two cases again:
i. $c$ is on $H(\mathscr{P}(t))$ at time $t$. From Property 3 , there exists a concave or a straight sector $s$ in $\{\Delta(x c y), \nabla(x c y)\}$ such that $x$ and $y$ are in $\mathscr{P}(t)$ and there is no point $\in \mathscr{P}(t) \cap s$. However, from Property 2 , we know that there exists no pair of points $x$ and $y$ in $\mathcal{P}(t)$ such that there exists a concave sector $\&$ in $\{\triangle(x c y), \nabla(x c y)\}$ and $\mathcal{P}(t) \cap s=\emptyset$. So, there exists only a straight sector $s$ in $\{\Delta(x c y), \nabla(x c y)\}$ such that $x$ and $y$ are in $\mathcal{P}(t)$ and there is no point $\in \mathcal{P}(t) \cap f$. Consequently, $c$ is on the segment $[x, y]$ at time $t$. Since the robots move in straight line toward $c$ and since some robots are located at $x$ and some other at $y$ which do not reach $c$ at time $t+1$ then, $c$ is on the segment $[r, s]$ at time $t+1$ with $r$ and $s \in \mathcal{P}(t+1)$. From Observation 1 , we deduce that $c$ is inside $\operatorname{SEC}(\mathcal{P}(t+1))$ at time $t+1$.
ii. $c$ is not on $H(\mathscr{P}(t))$ at time $t$. In this case, all the points in $\mathcal{P}(t)$ are not on the same line otherwise $c$ would have been on $H(\mathcal{P}(t))$. So, from Property 3 we know that there does not exist a concave or a straight sector $s$ in $\{\Delta(x c y), \nabla(x c y)\}$ such that $x$ and $y$ are in $\mathcal{P}(t)$ and there is no point $\in \mathscr{P}(t) \cap s$. Since the robots move in straight line toward $c$ and since for each point $p \in \mathcal{P}(t)$ there exists at least one robot located at $p$ which does not reach $c$ at time $t+1$ then, we deduce that there does not exist a concave or a straight sector $s$ in $\{\triangle(r c s), \nabla(r c s)\}$ such that $r$ and $s$ are in $\mathcal{P}(t+1)$ and there is no point $\in \mathcal{P}(t+1) \cap s$-refer to Fig. 2. So, from Property $3, c$ is inside $H(\mathcal{P}(t+1))$ at time $t+1$, which directly implies that $c$ is inside $\operatorname{SEC}(\mathcal{P}(t+1))$.

Lemma 6. Let $\mathcal{P}(t)$ be a configuration such that $|\operatorname{Max} \mathcal{P}| \geq 3$. If any robot $r$ is inside $\operatorname{SEC}(\mathcal{P}(t))$ and $r$ is on the boundary of $\operatorname{SEC}(\mathcal{P}(t+1))$ then $|\operatorname{Max} \mathcal{P}(t+1)| \leq 2$.

Proof. We consider two cases.

- No robot is inside SEC at time $t$. Since there is no robot inside $\operatorname{SEC}(\mathcal{P}(t))$ then there exists no robot inside $\operatorname{SEC}(\mathcal{P})$ at time $t$ which is on the boundary of $\operatorname{SEC}(\mathcal{P}(t+1))$ at $t+1$. So, in this case, Lemma 6 always holds.
- There exists at least one robot inside $\operatorname{SEC}(\mathcal{P}(t))$. By contradiction, assume that $r$ is inside $\operatorname{SEC}(\mathcal{P}(t))$ and $r$ is located on the boundary of $\operatorname{SEC}(\mathcal{P}(t+1))$ and $|\operatorname{Max} \mathcal{P}(t+1)|>2$. Let $c$ be the center of $\operatorname{SEC}(\mathcal{P}(t))$ at time $t$. From assumption, some robots on the boundary of $\operatorname{SEC}(\mathcal{P}(t))$ have moved toward the center of $\operatorname{SEC}(\mathcal{P}(t))$. According to Algorithm 1, that implies that all the robots inside $\operatorname{SEC}(\mathcal{P}(t))$, namely $r$, are located at the center of $\operatorname{SEC}(\mathscr{P}(t))$ at time $t$. So, $c$ is on the boundary of $\operatorname{SEC}(\mathcal{P}(t+1))$. From Lemma 5 , we deduce that there exists a point $p \in \mathcal{P}(t) \cap \operatorname{SEC}(\mathcal{P}(t))$ such that all the robots in $p$ have reached $c$ at time $t+1$. However, according to Algorithm 1 only the robots located in $\operatorname{Max} \mathcal{P}(t) \cap \operatorname{SEC}(\mathcal{P}(t))$ are allowed to move at time $t$. Therefore, for every point $p \neq c$ we have $|c|>|p|$ at time $t+1$. Hence, $\operatorname{Max} \mathcal{P}(t+1)=\{c\}$, i.e., $|\operatorname{Max} \mathcal{P}(t+1)|=1$. A contradiction.

Before proving Lemma 7, we need some extra definitions, observations and properties.


Fig. 3. Illustration of Observation 2.
Definition 5 (Chord). Given a circle $C$, a chord is a line segment whose two endpoints lie on the circle $C$.
Definition 6 (Circular Sector). A circular sector of a circle $C$ is a region bounded by a chord and an arc of the circle $C$ lying between the chord's endpoints.
Definition 7 (Sagitta). Given a circular sector, its sagitta is the line segment perpendicular to the chord between the midpoint of that chord and the arc of circle.

The relationship between chord and sagitta is given by the following property:
Property 4. Given a circular sector, its sagitta $x(x>0)$ and its chord $y$, the length of the radius $r$ of the unique circle which will fit around the two lines is given by:

$$
r=\frac{y^{2}}{8 x}+\frac{x}{2}
$$

Proof. Whether $r$ is larger than $x$ or not, from the Pythagorean Theorem, we know that:

$$
r^{2}=(r-x)^{2}+\left(\frac{y}{2}\right)^{2}
$$

Hence, we have:

$$
r^{2}=r^{2}-2 r x+x^{2}+\left(\frac{y}{2}\right)^{2}
$$

Then, we get:

$$
2 r x=x^{2}+\left(\frac{y}{2}\right)^{2}
$$

From above, we deduce that

$$
r=\frac{x^{2}}{2 x}+\frac{\left(\frac{y}{2}\right)^{2}}{2 x}=\frac{y^{2}}{8 x}+\frac{x}{2}
$$

Observation 2. Given a circle $C$ and its radius $r$ and given a circular sector $\mathcal{K}$, its chord ch, its sagitta $s$ and o the intersection of ch and s, we have

1. If $r<s$ then for every point $p$ on the arc of $\mathcal{K}, \overline{o p}<s$.
2. If $r=s$ then for every point $p$ on the $\operatorname{arc}$ of $\mathcal{K}, \overline{o p}=s$.
3. If $r>s$ then for every point $p$ on the arc of $\mathcal{K}, \overline{o p}>s$.

Observation 2 is depicted in Fig. 3.
Observation 3. Let $C$ and $r$ be a circle centered at $c$ and its radius, respectively. Given a circular sector of $C$, its two chord endpoints $A$ and $B$, and its sagitta s, we have:

If $r>s$ then the sector $\&$ in $\{\triangle(A c B), \nabla(A c B)\}$ which does not contain $s$ is a concave sector.


Fig. 4. An example illustrating the extra assumptions used in Case $b$ of the proof of Lemma 7.
In the following lemma, assuming at least one robot $\in \mathcal{P}(t) \cap \operatorname{SEC}(\mathcal{P}(t))$ moves in straight line toward the center $c$ of $\operatorname{SEC}(t)$, we show that if, during the computation step $(t, t+1)$, we do not obtain at most two maximal points and the set of robots located on the boundary of $S E C$ remains unchanged then the radius of $S E C$ is reduced by at least $\min \left(\frac{\epsilon}{2}, \frac{\epsilon^{2}}{4 \mathcal{R a d}(\operatorname{SEC}(\mathcal{P}(t)))}\right)$ (recall that $\epsilon$ is a constant defined at the end of Section 2.1).
Lemma 7. Let $\mathcal{P}(t)$ be a configuration such that $|\operatorname{Max} \mathcal{P}| \geq 3$. According to Algorithm 1, if at least one robot $\in \mathcal{P}(t) \cap S E C(\mathcal{P}(t))$ moves in straight line toward the center $c$ of $\operatorname{SEC}(t)$, then:

$$
\begin{aligned}
& |\operatorname{Max} \mathcal{P}(t+1)| \geq 3 \wedge \operatorname{SEC}(\mathcal{P}(t+1)) \cap \mathcal{P}(t+1)=\operatorname{SEC}(\mathcal{P}(t)) \cap \mathcal{P}(t) \\
& \Longrightarrow \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t+1))) \leq \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))-\min \left(\frac{\epsilon}{2}, \frac{\epsilon^{2}}{4 \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))}\right) .
\end{aligned}
$$

Proof. According to Algorithm 1, when $|\operatorname{Max} \mathcal{P}(t+1)| \geq 3$, the robots that are allowed to move, can only move toward the center of SEC. Thus, the fact that all the robots on the circumference of $\operatorname{SEC}(\mathcal{P}(t))$ remains on the circumference of $\operatorname{SEC}(\mathcal{P}(t+1))$ is possible only if the radius of $\operatorname{SEC}(\mathcal{P}(t+1))$ is closer than the radius of $\operatorname{SEC}(\mathcal{P}(t))$. So, in the rest of this proof, we show that the radius is reduced by at least $\min \left(\frac{\epsilon}{2}, \frac{\epsilon^{2}}{4 \mathcal{R a d}(\operatorname{SEC}(\mathcal{P}(t)))}\right)$ during the computation step $(t, t+1)$.

From assumption, at least one robot $\in \mathscr{P}(t) \cap \operatorname{SEC}(\mathcal{P}(t))$ moves in straight line toward the center $c$ of $\operatorname{SEC}(t)$. This implies that every robot inside $\operatorname{SEC}(t)$, if any, is located at the center of SEC and thus, remains so during the computation step $(t, t+1)$. So, it is enough to focus on the robots located on the boundary of SEC only. In this way, we have only two cases to consider:

1. All the robots allowed to move at time $t$ are at the same position $p_{1}$ on $\operatorname{SEC}(\mathcal{P}(t))$. From hypothesis, at least one robot $r \in \mathcal{P}(t) \cap \operatorname{SEC}(\mathscr{P}(t))$ moves in straight line toward the center $c$ of $\operatorname{SEC}(t)$. So, $r$ is located at $p_{1}$ at time $t$. Besides, all the robots located at $p_{1}$ have moved toward the center of SEC at time $t$. Indeed, if there exists one robot on $p_{1}$ which remains so during the computation step $(t, t+1)$ that would imply that $\operatorname{SEC}(\mathcal{P}(t))$ is not smaller than $\operatorname{SEC}(\mathcal{P}(t+1))$ and thus, $\operatorname{SEC}(\mathcal{P}(t+1)) \cap \mathcal{P}(t+1) \neq \operatorname{SEC}(\mathcal{P}(t)) \cap \mathcal{P}(t)$ because $r$ has moved inside SEC. That would be a contradiction. So, all the robots located at $p_{1}$ at time $t$ have moved inside SEC toward the center of SEC. Denote by $p_{2}$ the new position occupied at time $t+1$ by the slowest robot among those who were located at $p_{1}$. Again, we need to consider two cases:
(a) There are only two distinct positions that are occupied on SEC at time $t$. In this case, these two positions are on the same diameter (opposite positions), and $p_{1}$ is one of them. So, we know that the diameter of SEC is reduced by at least $\epsilon$ during the computation step $(t, t+1)$. Therefore, we deduce that $\operatorname{Rad}(\operatorname{SEC}(\mathscr{P}(t+1))) \leq \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))-\frac{\epsilon}{2}$.
(b) There are at least three distinct positions that are occupied on SEC at time $t$. For the sake of this case, we need to have some extra notations-refer to Fig. 4.

* the center of $\operatorname{SEC}(\mathcal{P}(t))$ and $\operatorname{SEC}(\mathcal{P}(t+1))$ are $c_{1}$ and $c_{2}$, respectively.
* $\mathcal{K}_{1}$ denotes the circular sector such that the arc of circle passes through $p_{1}$ and the two chords endpoints $A$ and $B$ are the two distinct occupied positions on $\operatorname{SEC}(\mathcal{P}(t))$ such that $A$ is adjacent to $p_{1}$ on $\operatorname{SEC}(\mathcal{P}(t))$ in counterclockwise and $B$ is adjacent to $A$ on $\operatorname{SEC}(\mathcal{P}(t))$ in counterclockwise. $\mathcal{K}_{2}$ denotes the circular sector whose the arc of circle passes through $p_{2}$ and such that the two chords endpoints are $A$ and $B$, i.e., the sames than $\mathcal{K}_{1}$.
* Sagittas $s_{1}$ and $s_{2}$ denote the sagitta of $\mathcal{K}_{1}$ and $\mathcal{K}_{2}$, respectively.
* The midpoint of chord $[A ; B]$ is called $o . k$ denotes the endpoint of $s_{2}$ on $\operatorname{SEC}(\mathscr{P}(t+1))$ which is opposite to $o$.
* The radii $r_{1}$ and $r_{2}$ stand for $\operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))$ and $\operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t+1)))$, respectively.

Clearly, $s_{1} \geq \mathscr{R a d}(\operatorname{SEC}(\mathcal{P}(t)))$. Indeed, if this would not be true, that would imply from Observation 3 that there is a concave sector $s$ in $\left\{\triangle\left(A c_{1} B\right), \nabla\left(A c_{1} B\right)\right\}$ such that there is no point in $\operatorname{arc}(\operatorname{SEC}(\mathcal{P}), \delta) \cap \mathscr{P}$ (because $A$ and $B$ are adjacent on $S E C$ ), which would contradict Property 1 . So, we can deduce $c_{1}$ is located at $s_{1}$. In the same way, we know that $s_{2} \geq \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t+1)))$. Thus $c_{2}$ is on $s_{2}$. So, in the following, we first prove that $s_{1}-s_{2} \geq \epsilon$. Then, we prove that $\operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t+1))) \leq \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))-\frac{\epsilon^{2}}{4 \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))}$.

1. $\left(s_{1}-s_{2} \geq \epsilon\right)$. Since $r_{1}>r_{2}$, we deduce that:

$$
\overline{O c_{1}}>\overline{O c_{2}}
$$

Now, consider circular sector $\mathcal{K}_{3}$ whose the arc of circle passes through $p_{2}$ and such that the chord is the line segment perpendicular to $s_{1}$ and passing through $c_{1}$ and bounded by $\operatorname{SEC}(\mathscr{P}(t+1))$. Seeing that $s_{3}=\overline{c_{1} k}<r_{2}$, from Observation 2:

$$
\overline{c_{1} p_{2}}>\overline{c_{1} k}
$$

And thus:

$$
\overline{c_{1} p_{2}}-\overline{c_{1} k}>0
$$

Besides:

$$
r_{1}=\overline{c_{1} p_{2}}+\overline{p_{2} p_{1}}
$$

Since $\overline{p_{2} p_{1}} \geq \epsilon$, we have:

$$
r_{1} \geq \overline{c_{1} p_{2}}+\epsilon
$$

However, we also have:

$$
r_{1}=\overline{c_{1} k}+\left(s_{1}-s_{2}\right)
$$

So, $\overline{c_{1} k}+\left(s_{1}-s_{2}\right) \geq \overline{c_{1} p_{2}}+\epsilon$
Then,

$$
s_{1}-s_{2} \geq \overline{c_{1} p_{2}}-\overline{c_{1} k}+\epsilon
$$

As mentioned previously, $\overline{c_{1} p_{2}}-\overline{c_{1} k}>0$. Thus:

$$
s_{1}-s_{2} \geq \epsilon
$$

2. $\left(\operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t+1))) \leq \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))-\frac{\epsilon^{2}}{4 \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))}\right)$. From Property 4: $r_{1}=\frac{\overline{A B}^{2}}{s_{1}}+\frac{s_{1}}{2}$ and $r_{2}=\frac{\overline{A B}^{2}}{s_{2}}+\frac{s_{2}}{2}$. Hence:

$$
r_{1}-r_{2}=\frac{\overline{A B}^{2}}{s_{1}}+\frac{s_{1}}{2}-\frac{\overline{A B}^{2}}{s_{2}}-\frac{s_{2}}{2}
$$

$$
\begin{aligned}
& \text { Since } s_{1}-s_{2} \geq \epsilon>0 \text {, we can state } s_{1}-s_{2}=\sigma>0 \text {. Thus: } \\
& \qquad r_{1}-r_{2}=\frac{\overline{A B}^{2}}{s_{1}}+\frac{s_{1}}{2}-\frac{\overline{A B}^{2}}{s_{1}-\sigma}-\frac{s_{1}-\sigma}{2}=\frac{\overline{A B}^{2}}{s_{1}}-\frac{\overline{A B}_{s_{1}-\sigma}^{2}}{2}+\frac{\sigma}{2}
\end{aligned}
$$

Which expands to:

$$
r_{1}-r_{2}=\frac{\overline{A B}^{2} 8\left(s_{1}-\sigma\right)-\overline{A B}^{2} 8 s_{1}}{8 s_{1} 8\left(s_{1}-\sigma\right)}+\frac{\sigma}{2}
$$

By simplifying, we get:

$$
r_{1}-r_{2}=\frac{\sigma}{2}-\frac{\sigma \overline{A B}^{2}}{8 s_{1}\left(s_{1}-\sigma\right)}
$$

Since $s_{2} \geq r_{2}$, we can say that $s_{1}-\sigma \geq r_{2}$. Moreover, chord [ $A ; B$ ] cannot be greater than the diameter of $\operatorname{SEC}\left(\mathcal{P}(t+1)\right.$ ), i.e., $\overline{A B} \leq 2 r_{2}$. So, $\overline{A B} \leq 2 r_{2} \leq 2\left(s_{1}-\sigma\right)$ and we obtain:

$$
r_{1}-r_{2} \geq \frac{\sigma}{2}-\frac{\sigma\left(2\left(s_{1}-\sigma\right)\right)^{2}}{8 s_{1}\left(s_{1}-\sigma\right)}=\frac{\sigma}{2}-\frac{4 \sigma\left(s_{1}-\sigma\right)^{2}}{8 s_{1}\left(s_{1}-\sigma\right)}
$$

Therefore:

$$
r_{1}-r_{2} \geq \frac{\sigma}{2}-\frac{\sigma\left(s_{1}-\sigma\right)}{2 s_{1}}=\frac{\sigma}{2} *\left(1-\frac{s_{1}-\sigma}{s_{1}}\right)
$$

So:

$$
r_{1}-r_{2} \geq \frac{\sigma}{2} *\left(\frac{\sigma}{s_{1}}\right)=\frac{\sigma^{2}}{2 s_{1}}
$$

However $\sigma=s_{1}-s_{2} \geq \epsilon$. Besides, $s_{1}$ is smaller than or equal to the diameter, i.e., $s_{1} \leq 2 r_{1}=2 \mathscr{R a d}(\operatorname{SEC}(\mathcal{P}(t)))$. Consequently:

$$
\begin{aligned}
& \quad \frac{\sigma^{2}}{2 s_{1}} \geq \frac{\epsilon^{2}}{4 \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))} \\
& \text { Finally, we can state: }
\end{aligned}
$$

$$
r_{1}-r_{2} \geq \frac{\epsilon^{2}}{4 \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))}
$$

2. All the robots allowed to move at time $t$ are on at least two distinct positions on $\operatorname{SEC}(\mathscr{P}(t))$. Clearly, in the worst case the radius of $S E C$ is reduced as much as the case where all the robots allowed to move at time $t$ are located at the same position $p_{1}$ on $\operatorname{SEC}(\mathcal{P}(t))$. So,

$$
\operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t+1))) \leq \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))-\min \left(\frac{\epsilon}{2}, \frac{\epsilon^{2}}{4 \operatorname{Rad}(\operatorname{SEC}(\mathcal{P}(t)))}\right)
$$

Lemma 8. Let $\mathcal{P}$ be a configuration such that $|\operatorname{Max} \mathcal{P}| \geq 3$. According to Algorithm 1, in every execution starting from $\mathcal{P}$, all the robots gather at the same position in finite time.

Proof. Assume by contradiction that there exists a time $t$ such that for all $t^{\prime} \geq t\left|\operatorname{Max} \mathcal{P}\left(t^{\prime}\right)\right| \geq 3$. From Lemma 6 we know that for all $t^{\prime} \geq t \operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}+1\right)\right) \cap \mathcal{P}\left(t^{\prime}+1\right)=\operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}\right)\right) \cap \mathcal{P}\left(t^{\prime}\right)$. So, for every $i$, if at least one robot $\in \operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}+i\right)\right) \cap \mathcal{P}\left(t^{\prime}+i\right)$ moves in straight line toward the center of $\operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}+i\right)\right)$ then from Lemma 7 we deduce:

$$
\operatorname{Rad}\left(\operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}+i+1\right)\right)\right) \leq \operatorname{Rad}\left(\operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}+i\right)\right)\right)-\min \left(\frac{\epsilon}{2}, \frac{\epsilon^{2}}{4 \operatorname{Rad}\left(\operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}+i\right)\right)\right)}\right)
$$

However, according to Algorithm 1, when $|\operatorname{Max} \mathcal{P}| \geq 3$, the robots that are allowed to move, can only move toward the center of SEC. So, for every $i$,

$$
\operatorname{Rad}\left(\operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}\right)\right)\right) \geq \mathscr{R a d}\left(\operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}+i\right)\right)\right)
$$

Consequently, for every $i$

$$
\min \left(\frac{\epsilon}{2}, \frac{\epsilon^{2}}{4 \operatorname{Rad}\left(\operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}\right)\right)\right)}\right) \leq \min \left(\frac{\epsilon}{2}, \frac{\epsilon^{2}}{4 \operatorname{Rad}\left(\operatorname{SEC}\left(\mathcal{P}\left(t^{\prime}+i\right)\right)\right)}\right)
$$

So, from $t^{\prime}$ on, each time a robot located on the boundary of SEC moves toward the center of SEC, the radius of SEC is reduced by an increasingly large value.

So, by fairness, we deduce that the robots gather at the same point in finite time: this is a contradiction with the fact that for all $t^{\prime} \geq t\left|\operatorname{Max} \mathcal{P}\left(t^{\prime}\right)\right| \geq 3$. In subsequence, there exists time $t^{\prime \prime}$ such that $\left|\operatorname{Max} \mathcal{P}\left(t^{\prime \prime}\right)\right| \leq 2$ and from Lemmas 3 and 4 , we deduce that Lemma 8 holds.

## 4. Conclusion

Assuming strong multiplicity detection, we shown it is possible to solve the self-stabilizing gathering problem with $n$ weak robots in SSM if, and only if, $n$ is odd. Our positive result is constructive, as we present an algorithm working with an odd number of robots.

Note that our results do not state that strong multiplicity detection is a necessary condition to solve the self-stabilizing gathering problem. As a matter of fact, we guess that our algorithm could work with other forms of multiplicity detection, for instance a function $D$ such that, for every pair of positions $p_{1}, p_{2}, D$ returns $-1,0$, or 1 depending on the number of robots located at $p_{1}$ is either strictly smaller than, equal to, or greater than $p_{2}$. In future works, we would like to investigate this problem in the fully asynchronous model (Corda).

## References

[1] Noa Agmon, David Peleg, Fault-tolerant gathering algorithms for autonomous mobile robots, SIAM J. Comput. 36 (1) (2006) 56-82.
[2] H. Ando, Y. Oasa, I. Suzuki, M. Yamashita, A distributed memoryless point convergence algorithm for mobile robots with limited visibility, IEEE Trans. Robot. Autom. 15 (5) (1999) 818-828.
[3] P Chrystal, On the problem to construct the minimum circle enclosing $n$ given points in a plane, in: The Edinburgh Mathematical Society, Third Meeting, 1885, p. 30.
[4] Mark Cieliebak, Gathering non-oblivious mobile robots, in: LATIN, 2004, pp. 577-588.
[5] Mark Cieliebak, Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, Solving the robots gathering problem, in: Proceedings of the 30th International Colloquium on Automata, Languages and Programming, ICALP 2003, 2003, pp. 1181-1196.
[6] Julien Clément, Xavier Défago, Maria Gradinariu, Potop-Butucaru Taisuke Izumi, Stéphane Messika, The cost of probabilistic agreement in oblivious robot networks, Inf. Process. Lett. 110 (11) (2010) 431-438.
[7] Xavier Défago, Maria Gradinariu, Stéphane Messika, Philippe Raipin Parvédy, Fault-tolerant and self-stabilizing mobile robots gathering, in: DISC, 2006, pp. 46-60.
[8] Xavier Défago, Akihiko Konagaya, Circle formation for oblivious anonymous mobile robots with no common sense of orientation, in: 2nd ACM International Annual Workshop on Principles of Mobile Computing, POMC 2002, 2002, pp. 97-104.
[9] Yoann Dieudonné, Petit Franck, Scatter of robots, Parallel Process. Lett. 19 (1) (2009) 175-184.
[10] Yoann Dieudonné, Franck Petit, Self-stabilizing deterministic gathering, in: Algorithmic Aspects of Wireless Sensor Networks, 5th International Workshop, ALGOSENSORS 2009, in: Lecture Notes in Computer Science, vol. 5804, Springer, 2009, pp. 230-241.
[11] EW Dijkstra, Self stabilizing systems in spite of distributed control, Commun. Assoc. Comput. Mach. 17 (1974) 643-644.
[12] Paola Flocchini, Evangelos Kranakis, Danny Krizanc, Nicola Santoro, Cindy Sawchuk, Multiple mobile agent rendezvous in a ring, in: LATIN 2004: Theoretical Informatics, 6th Latin American Symposium, in: Lecture Notes in Computer Science, vol. 2976, Springer, 2004, pp. 599-608.
[13] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, Peter Widmayer, Gathering of asynchronous robots with limited visibility, Theor. Comput. Sci. 337 (1-3) (2005) 147-168.
[14] Taisuke Izumi, Tomoko Izumi, Sayaka Kamei, Fukuhito Ooshita, Randomized gathering of mobile robots with local multiplicity, in: Eleventh International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS'09, in: Lecture Notes in Computer Science, vol. 5873, Springer, 2009, pp. 384-398.
[15] Ralf Klasing, Euripides Markou, Andrzej Pelc, Gathering asynchronous oblivious mobile robots in a ring, Theor. Comput. Sci. 390 (1) (2008) 27-39.
[16] Giuseppe Prencipe, Corda: Distributed coordination of a set of autonomous mobile robots, in: Proceedings of the Fourth European Research Seminar on Advances in Distributed Systems, ERSADS 2001, 2001, pp. 185-190.
[17] Giuseppe Prencipe, Impossibility of gathering by a set of autonomous mobile robots, Theor. Comput. Sci. 384 (2-3) (2007) $222-231$.
[18] Franco P. Preparata, S.J. Hong, Convex hulls of finite sets of poin ts in two and three dimensions, Commun. ACM 20 (2) (1977) 87-93.
[19] Samia Souissi, Xavier Défago, Masafumi Yamashita, Using eventually consistent compasses to gather memory-less mobile robots with limited visibility, ACM Trans. Auton. Adapt. Syst. 4 (1) (2009) 1-27.
[20] Ichiro Suzuki, Masafumi Yamashita, Agreement on a common $x-y$ coordinate system by a group of mobile robots, Intell. Robots: Sens. Model. and Planning (1996) 305-321.
[21] Ichiro Suzuki, Masafumi Yamashita, Distributed anonymous mobile robots - formation of geometric patterns, SIAM J. Comput. 28 (4) (1999) 1347-1363.


[^0]:    * A preliminary version of this work appears in [10].
    * Corresponding author. Tel.: +33 344792489.

    E-mail address: yoann.dieudonne@u-picardie.fr (Y. Dieudonné).

[^1]:    ${ }^{1}$ In [14], the authors provide randomized solutions for the gathering problem based on two variants (weak and strong) of multiplicity detection that considers the current position of the robot only.

