Self-Deployment Algorithms for Mobile Sensors on a Ring

Paola Flocchini * Giuseppe Prencipe† Nicola Santoro‡

Abstract

We consider the *self-deployment* problem in a ring for a network of identical sensors: starting from some initial random placement in the ring, the sensors in the network must move, in a purely decentralized and distributed fashion, so to reach in finite time a state of static equilibrium in which they evenly cover the ring. A self-deployment algorithm, the same for all sensors, will specify which sequence of operations (communication/sensing, computing a destination, moving towards there) a sensor must perform whenever it is active. The self-deployment algorithm is *exact* if within finite time the sensors reach a static *uniform* configuration: the distance between any two consecutive sensors along the ring is the same, d; the self-deployment algorithm is ϵ-*approximate* if the distance between two consecutive sensors is between $d - \epsilon$ and $d + \epsilon$.

We examine the algorithmic limitations and issues involved in this problem. We first of all prove a strong negative result. In fact, we prove that *exact* self-deployment is *impossible* if the sensors do not share a common orientation of the ring. This impossibility result holds even if the sensors have unlimited memory and unbounded computational power, and even if all their actions, when active, are instantaneous and their visibility/communication radius is unlimited.

We then consider the problem in an oriented ring. We prove that if the sensors know the desired final distance d, then *exact* self-deployment is possible: we present a simple protocol and prove that it allows the sensors to deploy themselves uniformly along the ring in finite time. If the desired final distance d is unknown, we present another protocol based on a very simple strategy and prove that it is ϵ-approximate for any chosen $\epsilon > 0$. These positive results hold even for the weakest sensors: oblivious (i.e., each sensor has no memory of past actions and computations), asynchronous (i.e., each sensor becomes active at unpredictable times and the duration of its actions is unpredictable), and without a common coordinate system; they work correctly even when every sensor can "locate" only its two neighbors or when the sensors have only a fixed sensing radius.

Our results show that a *shared orientation* of the ring is an important computational and complexity factor for a network of mobile sensors operating in a ring.

*School of Information Technology and Engineering, University of Ottawa, Canada. (flocchin@site.uottawa.ca)

†Dipartimento di Informatica, Università di Pisa, Italy. (prencipe@di.unipi.it)

‡School of Computer Science, Carleton University, Ottawa, Canada. (santoro@scs.carleton.ca)

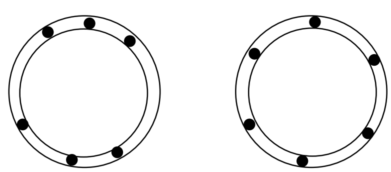
1 Introduction

1.1 The Framework

A mobile sensors network is composed of a distributed collection of sensors that in addition to the traditional sensing, computation, and communication capabilities of static sensors, have also locomotion capabilities. Mobility facilitates a number of useful network capabilities; for example, they can patrol a wide area, they can be re-positioned for better surveillance, etc.; moreover, they are especially useful in environments that may be both hostile and dynamic. There have been some research efforts on the deploying of mobile sensors, most of them based on centralized approaches; e.g., [21] assumes that a powerful cluster head is available to collect the sensor location and determine the target location of the mobile sensors.

Locomotion however allows the sensors to *self-deploy*; that is, starting from some initial random configuration, the sensors in the network can spread out in a purely decentralized and distributed fashion, and cover the area satisfying some optimization criteria (e.g., evenly, maximizing coverage, etc.) [10, 11, 12, 14, 20]. In contrast to [11], where the sensors are deployed one at the time, we consider the case when the sensors are deployed at the same time and they organize themselves in an adaptive manner. Unlike [14], we do not require prespecified destinations for the sensors, and unlike [11] we do not assume the sensors know where they are, since for small sensors localization is very hard. An essential requirement is that the network will reach a state of static equilibrium within finite time.

The *self-deployment* problem is quite similar to the *scattering* or *coverage* problem considered in cooperative mobile sensorics (e.g., [1]), and related to the *formation* problem (e.g. [9, 17, 18]); a key difference in these investigations is that usually there is no requirement that the network reaches a state of static equilibrium.



a. b.

Figure 1: Starting from an initial arbitrary placement (a), the sensors must move to a uniform cover of the ring (b).

1.2 The Problem

In this paper, we are interested in the self-deployment of a mobile sensor network in a *ring* (e.g., a circular rim; see Figure 1): starting from an initial random placement on the ring, the sensors must within finite time position themselves along the ring at (approximately) equal distance. A self-deployment algorithm, the same for all sensors, will specify which sequence of operations (communication/sensing, computing a destination, moving towards a point) a sensor must perform whenever it is active. We say that a self-deployment algorithm is *exact* if within finite time the sensors reach a *uniform* configuration: the distance between any two consecutive sensors along the ring is the same, d. We say that a self-deployment algorithm is ϵ-*approximate* if the distance between two consecutive sensors is between $d - \epsilon$ and $d + \epsilon$.

A self-deployment algorithm has recently been developed for the *line* [5] (e.g., a rectilinear corridor), and for the *ring*, as part of a larger protocol for uniform circle formation [2, 6, 13, 16, 19]. Both protocols yield only *approximate* solutions. However, they operate even with very weak sensors: anonymous (i.e., the sensors are indistinguishable), oblivious (i.e., each sensor has no memory of past actions and computations), asynchronous (i.e., each sensor becomes active at unpredictable times and the duration of its actions is unpredictable), and without a common coordinate system (e.g., no access to GPS). To date, no *exact* solution exists for these types of sensors.

1.3 Our Results

We first prove a strong negative result. In fact, we prove that *exact* self-deployment is actually *impossible* if the sensors do not share a *common orientation* of the ring; notice that this is much less a requirement than having global coordinates or sharing a common coordinate system. This impossibility result holds even if the sensors have unlimited memory and unbounded computational power, and even if all their actions, when active, are instantaneous and their visibility/communication radius is unlimited.

Faced with this strong negative result, the interesting question becomes under what restriction the self-deployment problem can be solved with an exact algorithm. Since the impossibility result holds in absence of common orientation of the ring, we consider the problem in *oriented* rings.

We prove that, in an oriented ring, if the sensors know the desired final distance d, then *exact* self-deployment is possible. In fact we present a simple protocol and prove that it allows the sensors to deploy themselves uniformly along the ring in finite time. This positive result holds even for the weakest sensors: anonymous, oblivious, asynchronous, with no common coordinate system; it works correctly even when every sensor can "locate" only its two neighbors or when the sensors have only a fixed sensing radius $v > d$.

Finally we turn to the case of an oriented ring when the desired final distance d is unknown. We present another protocol based on a very simple strategy and prove that it is ϵ-approximate for any fixed $\epsilon > 0$. As in [3, 4, 5], the difficulty is not in the protocol but in the proof of its correctness. Also in this case, the protocol works even for the weakest sensors: anonymous, oblivious, asynchronous, with no common coordinate system. The algorithm works correctly even when every sensor can "locate" only its two neighbors or when the sensors have only a fixed sensing radius $v \geq 2d$.

In the last protocol, the strategy we use is *go-to-half*. Interestingly was shown by Dijkstra [7] that in the *unoriented* ring *go-to-half* does *not* converge, and hence can not be used for self-deployment[1]. In other words, as already shown by our impossibility result, our result stresses that a *shared ring orientation* is an important computational and complexity factor for a network of mobile sensors operating in a ring.

1.4 Related work

The self-deployment problem has been investigated with the goal to cover the area so to satisfy some optimization criteria (e.g., evenly, maximizing coverage, etc.) [10, 11, 12, 14, 20]. For example, in [20] the problem is to maximize the sensor coverage of the target area minimizing the time needed to cover the area. Typically, distributed self-deployment protocols first discover the existence of coverage holes (the area not covered by any sensor) in the target area based on the sensing service required by the application. After discovering a coverage hole, the protocols calculate the target positions of these sensors, that is the positions where they should move. Loo et al. [14] considered a system consisting of a number of cooperating mobile nodes that move toward a set of prioritized destinations under sensing and communication constraints; unlike them, we do not require prespecified destinations for the sensors. Howard et al. [11] address the problem of incremental deployment, where sensors are deployed one-at-a-time into an unknown environment, and each sensor uses information gathered by previously deployed sensors to determine its deployment location. They assume every sensor is equipped with an ideal localization sensor. We do not assume the sensors know where they are, since for small sensors localization is very hard. The goal is to maximize network coverage under the constraint that nodes maintain line-of-sight with each other.

The self-deployment problem is related to a well studied problem in the field of autonomous mobile sensors: that of the *pattern formation* [8, 9, 17, 18]; in particular to the one of *uniform circle formation* [2, 6, 16]. In this problem, very simple sensors are required to uniformly place themselves on the circumference of a circle not determined in advance (i.e., the sensors do not know the location of the circle to form). The main difference between these robotics investigations and our self-deployment problem in the ring is that in those problems, the sensors can freely move on a two dimensional plane; in contrast, our sensors can move only on the ring.

The strategy *go-to-half*, that we employ in one of our protocols was first analyzed by Dijkstra [7]; he showed that in an *unoriented* ring *go-to-half* does *not* converge (and hence can not be used for self-deployment). Recently, *go-to-half* has been shown by Peleg [5] (with a very involved proof) to converge in a *line*. Convergence in the unoriented ring has been announced for the *go-to-half-half* strategy by Défago and Konagaya [6, 16].

2 Terminology and Model

We consider a sensors network in a ring (i.e., a circular line). Let $s_1, \ldots, s_n$ be the n sensors initially randomly placed on the ring (see Figure 1). Let $d_i(t)$ be the distance between sensor

[1]It does however converge in a *line* as recently shown by Peleg [5] with a very involved proof.

s_i and sensor s_{i+1} at time t. When no ambiguity arises, we will omit the time and simply indicate the distance as d_i.

We will use a very general definition of a sensor as a computational unit capable of sensing (e.g., by communication) the positions of other sensors in its surrounding (within a fixed radius), performing local computations on the located/communicated data, and moving towards the computed destination. The local computation is done according to a deterministic algorithm that takes in input the located/communicated data, and returns a destination point towards which the sensor moves. All the sensors execute the same algorithm.

Each sensor repeatedly cycles through four *states*: when active, a sensor determines the positions of the other sensors in its radius – *Locate*; it computes the next destination point by executing the algorithm – *Compute*; and it moves towards the computed point – *Move*; after such a move the sensor may become inactive – *Wait*. The sequence: *Wait* - *Locate* - *Compute* - *Move* form a *computation cycle* (or briefly *cycle*) of a sensor. In the following, the "view of the world" of a sensor is defined as a snapshot of the positions of the other sensors in its own coordinate system (obtained in the *Locate* state).

The sensors are completely *autonomous*: no central control is needed. Furthermore they are *anonymous*, meaning that they are a priori indistinguishable by their appearance, and they do not (need to) have any kind of identifiers that can be used during the computation. They are *oblivious*: each sensor has no memory of past actions and computations; in other words, the computation is based solely on what located in the current cycle.

In general, no assumptions on the cycle time of each sensor and on the time each sensor takes to execute each state of a given cycle are made. It is only assumed that each cycle is completed in finite time, and that the distance traveled in a cycle is finite. Moreover, the sensors do not need to have a common notion of time, and each sensor can execute its actions at unpredictable time instants: this scenario is called *asynchronous* (Async).

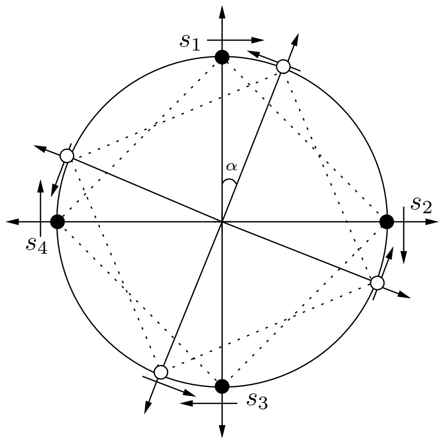
We also consider (in our impossibility result) a different scenario, where there is a global clock tick reaching all sensors simultaneously, and a sensor's cycle is an instantaneous event that starts at a clock tick and ends by the next. This scenario is called *semi-synchronous* (SSync). The only unpredictability is given by the fact that at each clock tick, every sensor is either *active* or *inactive*, and only active sensors perform their cycle. The unpredictability is restricted by the fact that at least one sensor is active at every time instant, and every sensor becomes active at infinitely many unpredictable time instants.

Let us denote by $\mathbb{AS}$ and $\mathbb{SS}$ the class of problems that are solvable in the asynchronous and in the semi-synchronous setting, respectively. Then,

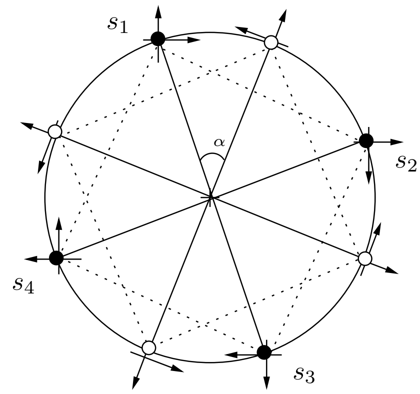
Theorem 1 ([15]). $\mathbb{AS} \subset \mathbb{SS}$.

3 Impossibility of Exact Self-Deployment

In this section, we show that the exact self-deployment problem is unsolvable; in other words, given a set of sensors placed on the rim of a circle, there exists no deterministic algorithm that, in a finite number of cycles, places the sensors uniformly on the ring.



a.



b.

Figure 2: (a) An example of starting configuration for the proof of Theorem 2. The black sensors are in S_1, while the white ones in S_2. (b) Theorem 2: the adversary moves only sensors in S_1.

Theorem 2. *Let $s_1, \dots, s_n$ be all on a ring $\mathcal{C}$. Then, in* SSYNC*, there is no deterministic exact self-deployment algorithm.*

Proof. By contradiction, let us assume there exists a deterministic algorithm $\mathcal{A}$ that solves the problem in a finite number of cycles. Furthermore, let us assume that there is an even number of sensors placed on $\mathcal{C}$, and that the n sensors can be split in two subsets according to their views of the world. In particular, in the first subset, call it S_1, there are $s_1, \dots, s_{n/2}$, and in the second subset, call it S_2 the other sensors. The sensors in S_1 and S_2 are placed on the vertices of two regular $n/2$-gons, and the two polygons are rotated of an angle smaller than $360°/n$. Furthermore, all sensors have their local coordinate axes rotated so that they all have the same view of the world (refer to Figure 2.a for an example).

Lemma 1. *If activating only the sensors in S_1 no exact self-deployment on $\mathcal{C}$ is reached, then also activating only the ones in S_2 no exact self-deployment on $\mathcal{C}$ is reached.*

Lemma 2. *If activating only the sensors in S_1 an exact self-deployment on $\mathcal{C}$ is reached, then also activating only the sensors in S_2 an exact self-deployment on $\mathcal{C}$ is reached. Moreover, activating both sets no exact self-deployment on $\mathcal{C}$ is reached.*

Proof. Let s_1 be an arbitrary sensor in S_1 (refer to Figure 3). By construction, s_1 has two neighbors on $\mathcal{C}$, s_2' and s_2'', and both of them are in S_2. Let $\beta = \min(s_1\widehat{c}s_2', s_1\widehat{c}s_2'')$ (clearly, $s_1\widehat{c}s_2'$ cannot be equal to $s_1\widehat{c}s_2''$, otherwise the sensors would be uniformly placed on $\mathcal{C}$). By hypothesis, by activating only the sensors in S_1, the sensors would reach an exact self-deployment on $\mathcal{C}$. In other words, they would all rotate of an angle γ so that, at time $t+1$, $\beta + \gamma = 360°/n$. Symmetrically, if only the sensors in S_2 would be activated, they would

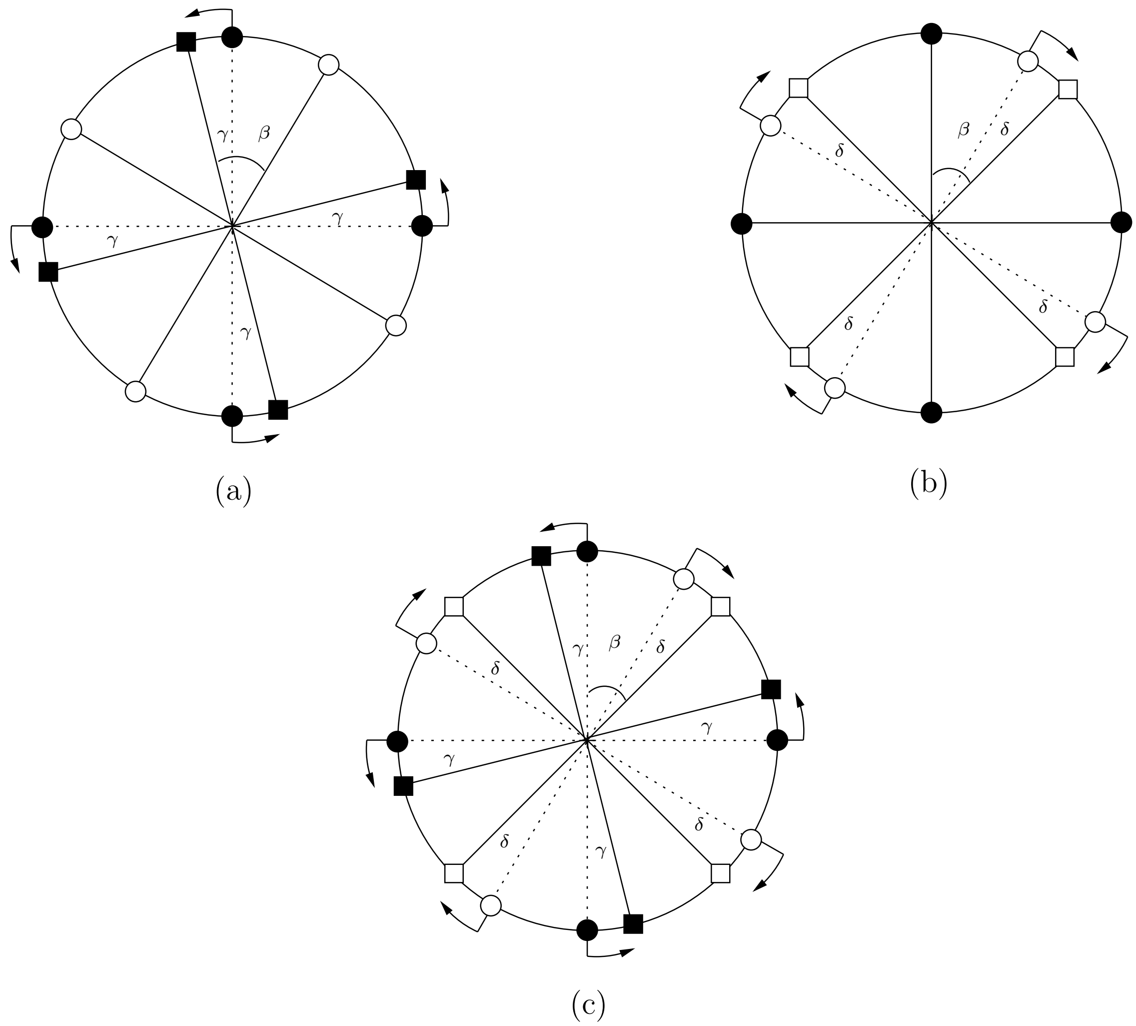


Figure 3: Theorem 2. (a) If only the sensors in S_1 are activated at t, all sensors would be uniformly placed at time $t+1$, with $\beta + \gamma = 45°$. (b) If only the sensors in S_2 are activated at t, all sensors would be uniformly placed at time $t+1$, with $\beta + \gamma = 45°$. (c) Therefore, if all sensors would be activated at t, they would not be in an exact self-deployment on $\mathcal{C}$, having $\gamma + \beta + \delta \neq 2\pi/n = 45°$. In all figures, the squares represent the destination of the active sensors.

Algorithm 1 The Adversary

(a) If activating only the sensors in S_1 no exact self-deployment on $\mathcal{C}$ is reached. Activates all sensors in S_1, while all sensors in S_2 are inactive, and goto (c). Otherwise,

 (b) If activating only the sensors in S_2, no exact self-deployment on $\mathcal{C}$ is reached. In this case, it activates all sensors in S_2, while all sensors in S_1 are inactive, and goto (c). Otherwise, all sensors are activated, and goto (c).

(c) If activating only the sensors in S_2 no exact self-deployment on $\mathcal{C}$ is reached. In this case, it activates all sensors in S_2, while all sensors in S_1 are inactive, and goto (a). Otherwise,

 (d) If activating only the sensors in S_1 no exact self-deployment on $\mathcal{C}$ is reached. In this case, it activates all sensors in S_1, while all sensors in S_2 are inactive, and goto (a). Otherwise, all sensors are activated, and goto (a).

---

rotate of an angle δ so that, at time $t+1$, $\beta+\delta = 360°/n$. Therefore, since $\beta+\gamma+\delta \neq 360°/n$, by activating all sensors, an uniform placement on $\mathcal{C}$ cannot be reached at time $t+1$. $\square$

In the following, we define an adversary so that $\mathcal{A}$ never succeed in solving the problem. Algorithm 1 reports the protocol followed by the adversary.

First we note that, by the way the adversary is defined and since the sensors in S_1 (resp. S_2) have the same view, these sensors will always move together (when all activated). In the following, we will prove by induction the following property $\mathcal{P}rop$:

> for all $t \geq 0$, the sensors all have the same view of the world and are not in an exact self-deployment on $\mathcal{C}$.

By construction, $\mathcal{P}rop$ is clearly true at $t = 0$. Let us assume it is true at a given time $t > 0$. We distinguish the possible cases.

1. If the check performed in (a) is true, then clearly at time $t+1$ there is no exact self-deployment on $\mathcal{C}$. Furthermore, all sensors will still have the same view of the world (see the example depicted in Figure 2.b).

2. If the check performed in (a) is true, then rule (b) is executed. Two subcases can occur.

 3.1. If the check of rule (b) is false, then at time $t+1$ there is no exact self-deployment on $\mathcal{C}$, and all sensors have the same view of the world.

 3.2. Otherwise, all sensors are activated at time t, and by Lemma 2 no exact self-deployment on $\mathcal{C}$ is reached at time $t+1$.

3. Rules (c) and (d) are handled symmetrically to previous rules (a) and (b).

Therefore, there is no time $t' \geq t$ so that the sensors are in a exact self-deployment on $\mathcal{C}$, having a contradiction. □

By Theorem 1, we have

Corollary 1. *Let $s_1, \ldots, s_n$ be all on $\mathcal{C}$. Then, in* Async *there is no deterministic algorithm that brings them uniformly distributed on $\mathcal{C}$ in a finite number of cycles.*

4 Self-Deployment in an Oriented Ring: Interdistance Known

In this section we assume that the final distance d between two sensors is known to them. Moreover, the sensors have a fixed visibility radius of $2d$ and they can only locate up to such distance.

4.1 The Algorithm

The algorithm is very simple: sensors asynchronously and independently observe clockwise at distance $2d$, then they position themselves at distance d from the closest observed sensor (if any).

Protocol Uniform Known (for sensor s_i)

- Locate clockwise at distance $2d$. Let d_i be the distance to next sensor. If none, $d_i = 2d$.
- If $d_i \leq d$ do not move.
- If $d_i > d$ move clockwise and place yourself at distance d from s_{i+1}.

4.2 Correctness

We say that a sensor is *white* if its distance to the clockwise neighbor is greater than or equal to d. We say that a sensor is *gray* if such a distance is smaller than d. Moreover we say that a white sensor is *good* if its distance to the clockwise neighbor is exactly d, it is *large* if its distance is strictly greater than d.

We call a *white bubble* a sequence of consecutive white sensors delimited by grey sensors. Let $W = s_i, s_{i+1}, \ldots, s_{i+m}$ be a white bubble. Sensor s_{i-1} is said to be the predecessor of the bubble, sensor s_{i+m+1} is the successor. Clearly predecessors and successors of a white bubble are gray, unless the ring contains white sensors only; notice that in this case all sensors are good. The size of W , indicated as $|W|$ is the number of white sensors composing the bubble (in this example m), its length, indicated by $l(W)$, is the length of the ring between the predecessor of the white bubble and its successor (assuming not all sensors are white); i.e., $l(W) = \sum_{j=-1}^{m} d_{i+j}$. Similarly, we define a *gray bubble* $G = s_i, s_{i+1}, \ldots, s_{i+m}$ as a sequence of consecutive gray sensors delimited by white sensors. Its size $|G|$ is the number of gray sensors in G; the length $l(G)$ is defined as the length of the ring between the first and the last gray sensor in G (note that this definition is different from $l(W)$).

The next two lemmas contain some simple facts.

Lemma 3. *At each point in time, if there are gray sensors, then the number of white bubbles equals the number of gray bubbles.*

Lemma 4. *At each point in time, if there are grey sensors there must be at least a bubble (i.e., a large sensor).*

Lemma 5. *A white sensor cannot become gray.*

Proof. In order for a white sensor s_j to become gray, its distance to the next sensor s_{j+1} should become smaller than d. By definition, sensors move clockwise and move according to the algorithm; so sensor s_{j+1} will never get closer to s_j. On the other hand, by definition of our algorithm, sensor s_j will never move at a distance smaller than d to s_{j+1}. □

Lemma 6. *Let $W = s_i, s_{i+1}, \ldots s_{i+m}$ be a white bubble in the ring at time t. If $l(W) \geq d \cdot (|W| + 1)$, in finite time, say at time t', the size of the bubble increases.*

Proof. We want to prove by induction on the sensors in W that, by time t', all sensors in the white bubble are good, and the predecessor s_{i-1} is white (which means that the bubble has become bigger).

By definition of our algorithm, in finite time, say at time t_1, s_{i+m} becomes good placing itself at distance d to the successor of W. Let us assume that at time $t_j < t'$ all sensors $s_{i+m}, s_{i+m-1} \ldots, s_{i+m-j}$ are good. Let us consider now sensor $s_{i+m-j-1}$. If this sensor is not already good, by definition of the algorithm and since by hypothesis the successor of W does not become white, $s_{i+m-j-1}$ will move to place itself at distance d to s_{i+m-j}, thus becoming good at time t_{j+1}.

Thus, in finite time, say at t', all sensors in the bubble are good, which means that the distance between sensor s_i and sensor s_{i+m+1} is equal to $d \cdot m = d \cdot s(W)$. Since, by hypothesis, $l(W) \geq d \cdot (s(W) + 1)$,it follows that the distance between s_{i-1} and s_i becomes greater than or equal to d, which means that s_{i-1} has become white. □

Lemma 7. *Let $W_1, \ldots W_z$ be the white bubbles present in the ring at time t. At least one of these bubble W_k is such that $l(W_k) \geq d \cdot |W_k| + 1$.*

Proof. By contradiction, let $l(W_i) < d \cdot (|W_i| + 1)$, for all W_i. The length L of the ring is the sum of the lengths of all white bubbles and all gray bubbles. That is, from Lemma 3, $L = \sum_{i=1}^{z}(l(W_i) + l(G_i))$. By hypothesis, $\sum_{i=1}^{z} l(W_i) < d \sum_{i=1}^{z} |W_i| + d \cdot z$. Moreover, by definition of gray bubble, $\sum_{i=1}^{z} l(G_i) < d \sum_{i=1}^{z}(|G_i| - 1) = d \sum_{i=1}^{z} |G_i| - d \cdot z$. Summing up, we have $L < d \sum_{i=1}^{z}(|G_i| + |W_i|) = d \cdot n$, a contradiction. □

By Lemmas 6 and 7, we have that:

Lemma 8. *The number of grey sensors decreases.*

Finally, by Lemmas 5 and 8 we derive the main theorem.

Theorem 3. *In finite time all sensors are* good.

5 Self-Deployment in an Oriented Ring: Interdistance Unknown

In this section we assume that each sensor has a fixed visibility radius of v, and does not know the final interdistance d between the sensors. Although d is not known, we must have that $v > 2d$ for our algorithm to work.

5.1 The Algorithm

Also this algorithm is very simple: sensors asynchronously and independently locate in both directions at distance v, then they position themselves in the middle between the closest observed sensor (if any).

Protocol UNIFORM UNKNOWN (for sensor s_i)

- Locate around at distance v. Let d_i be the distance to next sensor, d_{i-1} the distance to the previous (if no sensor is visible clockwise, $d_i = v$, analogously for counterclockwise).
- If $d_i \leq d_{i-1}$ do not move.
- If $d_i > d_{i-1}$ move to $\frac{d_i+d_{i-1}}{2} - d_{i-1}$ clockwise.

5.2 Correctness

Let $d_{min}(t) = Min\{d_i(t)\}$ and $d_{max}(t) = Max\{d_i(t)\}$. Let C be the length of the circumference of the ring. First observe the following simple fact:

Lemma 9. *We have that:* $\forall t$, $d_{min}(t) \leq d$ *and* $d_{max}(t) \geq d$.

Proof. By contradiction. Let the minimum distance be greater than d. We would have that $C > k \cdot d$, which is impossible since by definition $C = k \cdot d$. Same argument holds for d_{max}. □

The next lemma shows that if, at some point there is a unique minimum (resp. maximum) interval, it will become bigger (resp. smaller).

Lemma 10. *If at time t there is a unique minimum interval, we have that: $\forall t, \exists t' > t$: $d_{min}(t') > d_{min}(t)$. If at time t there is a unique maximum interval, we have that: $\forall t, \exists t' > t$: $d_{max}(t') < d_{max}(t)$.*

Proof. Let s_{j-1} and s_j be the sensors that delimit the minimum interval $[s_{j-1}, s_j]$, whose length is $d_{j-1}(t) = d_{min}(t)$ at time t. First observe that, since $d_{j-2}(t) > d_{j-1}(t)$, by the algorithm we know that sensor s_{j-1} does not move at time t; actually, it will not be able to move as long as d_{j-2} remains greater than d_{j-1} (i.e., as long as s_j does not move). Consider now the first time t' when s_j is activated. Since s_{j-1} has not moved from time t to time t', we have that, at time t', $d_{j-2}(t')$ is still greater than $d_{j-1}(t')$. At time t', s_i then moves following the rule of the algorithm and $d_{j-1}(t') = \frac{d_{j-1}(t)+d_j(t')}{2} \geq \frac{d_{j-1}(t)+d_j(t)}{2} > d_{j-1}(t)$. Similar argument holds for d_{max}. □

We now show that if at some point there are several minimum (resp. maximum) intervals of a certain length, their number will decrease.

Lemma 11. *If at time t there are $r > 1$ minimum intervals of length $d_{min}(t)$, either all intervals have length d and the sensors are deployed, or there exists a time $t' > t$ when the number of minimum intervals of length $d_{min}(t)$ is $r' < r$.*

Proof. First notice that, if at time t a sensor s_j delimiting a minimum interval $[s_{j-1}, s_j]$ is activated, it will not move if $d_{min}(t) = d_{j-1}(t) = d_j(t)$ (i.e., if $[s_j, s_{j+1}]$ is another minimum interval), it will instead move if $d_{j-1}(t) < d_j(t)$.

Consider the first time t' when a sensor s_j delimiting a minimum interval $[s_{j-1}, s_j]$, which is *not* followed by another minimum interval, is activated. Notice that such a sensor must exist otherwise we would be in a situation when all sensors are deployed at distance d from each other. In this case we know that at time t' there are still at most s minimum intervals and that $d_{j-1}(t') < d_j(t')$. Sensor s_j then moves and $d_{j-1}(t') = \frac{d_{j-1}(t)+d_j(t')}{2} \geq \frac{d_{j-1}(t)+d_j(t)}{2} > d_{j-1}(t)$, thus it is not minimum anymore and the number of minimum intervals is now strictly smaller than r. □

Analogously,

Lemma 12. *If at time t there are $r > 1$ maximum intervals, either all intervals have length d and the sensors are deployed, or there exists a time t' when the number of maximum intervals is $r' < r$.*

We now show that the minimum intervals converge to a value $A = d - \gamma_{min}$, with $\gamma_{min} \geq 0$, and the maximum intervals converge to a value $B = d + \gamma_{min}$, with $\gamma_{max} \geq 0$.

Lemma 13. *Let $d_{min}(t)$ (resp $d_{max}(t)$) be the distance of a minimum (resp. maximum) interval at time t. We have that, for any arbitrary small $\epsilon > 0$ there exists a time $t' > t$ such that, $\forall t'' > t'$: $|d_{min}(t'') - A| \leq \epsilon$, and, $\forall t'' > t'$: $|d_{max}(t'') - B| \leq \epsilon$.*

Proof. From Lemmas 10 and 11 the intervals must converge; from Lemma 9 the minimum must converge to a value smaller than (or equal to) d, and the maximum must converge to a value greater than (or equal to) d. □

Let us call *A-regular* at time t an interval that, at time t is ϵ-close to A; that is an interval whose length $d_j(t)$ is such that $|d_j(t) - A| \leq \epsilon$. Analogously, we call *B-regular* an interval that is ϵ-close to B. We call *A-irregular* at time t an interval that, at time t, is smaller than d, but not ϵ-close to A; *B-irregular* one that is greater than d, but not ϵ-close to B.

The following lemma shows that there exists a time t, after the time when the previous Lemma 13 holds, when any interval greater than the minimum (and smaller than d) is *A-regular*, and any interval smaller than the maximum (and greater than d) is *B-regular*. In other words, each interval is either ϵ-close to A or to B. Notice that this property is not obvious; in fact, the only thing we know up to now is the convergence to A and B of the minimum/maximum intervals over time, while nothing is known about the other intervals.

Lemma 14. *Let $\epsilon > 0$ be arbitrarily small, and let t'_ϵ be a time when Lemma 13 holds. There exists a time $t''_\epsilon > t'_\epsilon$ when: for all intervals $[s_j, s_{j+1}]$ with $d_j(t'') \leq d$, $|d_j(t''_\epsilon) - A| \leq \epsilon$; for all intervals $[s_i, s_{i+1}]$ with $d_i(t''_\epsilon) \geq d$, $|d_i(t'') - B| \leq \epsilon$.*

Proof. By contradiction, assume such a situation never happens. Then, there must exist a time t when there are both *A-irregular* and *B-irregular* intervals.

Consider the following execution: 1) if there are *A-regular* intervals followed by *B-regular* intervals, let the sensors between them move. Notice that whenever a sensor between a *A-regular* and a *B-regular* intervals move, both intervals become irregular. Further notice that, after this activation rule, we are guaranteed that a sequence of regular intervals delimited by irregular intervals contains only intervals of the same type (*A-regular* or *B-regular* only). 2) Consider any *A-irregular* interval $[s_j, s_{j+1}]$. Let it be preceded by $k \geq 0$ *A-regular* intervals (delimited by sensors $s_{j-1} \dots s_{j-k}$) and followed by $h \geq 0$ *B-regular* interval (delimited by sensors $s_{j+2} \dots s_{j+h-1}$). Activate sensors $s_{j+1}, s_{j+2} \dots s_{j+h-1}$, $s_{j-1} \dots s_{j-k}$, in this order. It is easy to see that their movement transforms all those interval in irregular intervals. 3) Apply the same schedule to all *B-irregular* intervals (preceded by *A-regular* intervals and followed by *B-regular* intervals).

Notice that, by the above activation rules, a sequence of *A-regular* intervals becomes irregular if it is followed by *B-irregular* intervals or if it is preceded by *A-regular* intervals. Thus, after the activation rules of 2) and 3) we are in a situation where all intervals (included the minimum) are irregular and thus Lemma 13 is violated. □

Lemma 15. *Let t be a time when Lemma 14 holds. If at some time $t' > t$ at least an interval becomes irregular, then there exists a time $t'' > t'$ when all intervals are irregular.*

Proof. The argument is very similar to the one of Lemma 14. □

We now show that, after a time when Lemma 14 holds, all intervals actually converge to d (i.e., $A = B = d$).

Lemma 16. *Let $\epsilon > 0$ be arbitrarily small, and let t'_ϵ be a time when Lemma 14 holds. If $B - A > 2\epsilon$, at least an interval becomes irregular.*

Proof. Let $t_1 = t'_\epsilon$. We will show that, under the conditions of the statement there exists a movement of a sensor at time t_1 that create an irregular interval.

Consider two consecutive intervals $[s_i, s_{i+1}]$ and $[s_{i+1}, s_{i+2}]$ such that $d_i(t_1) < d$ and $d_{i+1}(t_1) > d$. Such intervals must exist because otherwise all the sensors would be deployed at precisely distance d from each other. By Lemma 14, we have that:

$$|d_i(t_1) - A| \leq \epsilon \tag{1}$$

$$|d_{i+1}(t_1) - B| \leq \epsilon \tag{2}$$

Let sensor s_{i+1} move at time t_1. As a result of the movement, at any time $t_2 > t_1$ before any other movement of the sensors, we have that:

$$d_i(t_2) = \frac{d_i(t_1) + d_{i+1}(t_1)}{2} = d_{i+1}(t_2) \tag{3}$$

We now consider several different cases.

Case 1. $A + \epsilon \geq d_i(t_1) > A$ and $B + \epsilon \geq d_{i+1}(t_1) > B$. From Equation 3 and for the assumption, we have that:

$$\frac{A+B}{2} < d_i(t_2) = d_{i+1}(t_2) \leq \frac{A+B+2\epsilon}{2} \tag{4}$$

We now consider the two case $d_i(t_2) > d$ and $d_i(t_2) < d$ and in both we will derive a contradiction.

1.1) Let $d_i(t_2) > d$. In this case we would have that $\frac{A+B+2\epsilon}{2} \geq d_i(t_2) > d$. We now consider the two cases: $d_i(t_2) > B$, and $d_i(t_2) < B$. If $d_i(t_2) > B$ it must be that $\frac{A+B+2\epsilon}{2} \geq B$, which would imply $A + 2\epsilon > B$, which is a contradiction with the assumption that $B - A > 2\epsilon$. It follows that $d < d_i(t_2) < B$. However, from Equation 2, we must have that $B - d_i(t_2) \leq \epsilon$, which would imply $B - \frac{A+B+2\epsilon}{2} \leq \epsilon$, that is $B - A \leq \epsilon$, which is a contradiction.

1.2) Let $d_i(t_2) < d$. In this case we would have to show that, by Equation 1, $d_i(t_2) - A \leq \epsilon$. However, $d_i(t_2) - A > \frac{B}{2} - \frac{A}{2}$, which is clearly greater than ϵ. Contradiction.

Case 2. $d_i(t_1) < A$ and $d_{i+1}(t_1) < B$. From Equation 3 and for the assumption, we have that: $d_i(t_2) = d_{i+1}(t_2) < \frac{A+B}{2}$.

By Equations 1 and 2 we must have that $A - d_i(t_1) \leq \epsilon$ and $B - d_{i+1}(t_1) \leq \epsilon$. In other words, $d_i(t_1) \geq A - \epsilon$, and $d_{i+1}(t_1) \geq B - \epsilon$. By Equation 3 and by the above, we have that $d_i(t_2) \geq \frac{A+B-2\epsilon}{2}$ (notice that, since $B > A$, this implies that $d_i(t_2) > A$). Thus we have:

$$\frac{A+B-2\epsilon}{2} \leq d_i(t_2) < \frac{A+B}{2} \tag{5}$$

Consider now the two possibilities $A < d_i(t_2) < d$ and $d_i(t_2) > d$: in both cases, we will show a contradiction.

2.1) If $A < d_i(t_2) < d$, Equation 1 must hold, that is $d_i(t_2) - A \leq \epsilon$. However, $d_i(t_2) - A \geq \frac{B}{2} - \frac{A}{2} - \epsilon$, which is clearly greater than ϵ, since $B - A > 2\epsilon$.

2.2) Consider now the case $d_i(t_2) > d$, in this case, by Equation 2, we must have $|d_i(t_2) - B| \leq \epsilon$. Since $A < B$, and thus $\frac{A+B}{2} < B$, we have that $d_i(t_2) < B$, so, by Equation 2 it must be: $B - d_i(t_2) \leq \epsilon$, or, in other words, $d_i(t_2) \geq B - \epsilon$. However from Equation 5, we know that $d_i(t_2) < \frac{A}{2} + \frac{B}{2}$ which is clearly smaller than $B - \epsilon$ (because $B - A > 2\epsilon$). Contradiction.

Case 3. $A + \epsilon \geq d_i(t_1) > A$ and $d_{i+1}(t_1) < B$. We have $d_i(t_1) > A$, and by definition we have $B - d_{i+1}(t_1) \leq \epsilon$; thus, from Equation 3 we obtain: $d_i(t_2) = d_{i+1}(t_2) \geq \frac{A+B-\epsilon}{2}$. Moreover, by the assumptions we get $d_i(t_2) = d \leq \frac{A+B+\epsilon}{2}$. Thus

$$\frac{A+B-\epsilon}{2} \leq d_i(t_2) < \frac{A+B+\epsilon}{2} \tag{6}$$

3.1) If $d_i(t_2) < d$ we should have (by Equation 1) that $d_i(t_2) - A \leq \epsilon$. However, by Equation 6, we have $d_i(t_2) - A \geq \frac{B-A-\epsilon}{2} \geq \epsilon$. Contradiction.

3.2) Let $d_i(t_2) > d$. First observe that $d_i(t_2)$ cannot be greater than B because we have $d_i(t_2) \leq \frac{A+B+\epsilon}{2} < B$); thus $d_i(t_2) < B$. We should have (by Equation 2) that $B - d_i(t_2) \leq \epsilon$. However, from Equation 6 we know that $d_i(t_2) - B \leq \frac{A+B+\epsilon}{2} < \epsilon$. Contradiction.

Case 4. $d_i(t_1) < A$ and $B + \epsilon \geq d_{i+1}(t_1) > B$. We have $d_i(t_1) < A$ and $d_{i+1}(t_1) \leq B + \epsilon$; thus, from Equation 3 we obtain: $d_i(t_2) = d_{i+1}(t_2) < \frac{A+B+\epsilon}{2}$. Moreover, by assumption $d_{i+1}(t_1) \leq B + \epsilon \geq$, and by definition $A - d_{i+1}(t_1) \leq \epsilon$, so we get: $d_i(t_2) \geq \frac{A+B-\epsilon}{2}$. Thus

$$\frac{A+B-\epsilon}{2} \leq d_i(t_2) < \frac{A+B+\epsilon}{2} \tag{7}$$

The rest of the proof proceeds like for Case 3. □

Theorem 4. *For any arbitrary small $\epsilon > 0$ there exists a time t, such that $\forall t' > t, \forall i$: $|d_i(t') - d| \leq \epsilon$.*

Proof. By contradiction. Let $A \neq B$. From Lemma 14, there is a time t when all intervals are ϵ-close to A and B. From Lemma 16, at least one interval will become irregular at some time $t' > t$. However, by Lemma 15 there is a time $t'' > t'$ when all intervals become irregular (including the minimum and the maximum). This contradicts Lemma 13. □

Acknowledgments. The authors would like to thank Vincenzo Gervasi, Toni Mesa, and Linda Pagli for the many discussions and suggestions.

References

[1] Y. U. Cao, A. S. Fukunaga, A. B. Kahng, and F. Meng. Cooperative Mobile Robotics: Antecedents and Directions. In *IEEE/TSJ International Conference on Intelligent Robots and Systems*, pages 226–234, 1995. Yokohama, Japan.

[2] I. Chatzigiannakis, M. Markou, and S. Nikoletseas. Distributed Circle Formation for Anonymous Oblivious Robots. In *Experimental and Efficient Algorithms: Third International Workshop (WEA 2004)*, volume LNCS 3059, pages 159 –174, 2004.

[3] R. Cohen and D. Peleg. Convergence Properties of the Gravitational Algorithm in Asynchronous Robot Systems. In *Proc. of the 12th European Symposium on Algorithms*, volume LNCS 3221, pages 228–239, 2004.

[4] R. Cohen and D. Peleg. Robot Convergence via Center-of-Gravity Algorithms. In *Proc. of the 11th Int. Colloquium on Structural Information and Communication Complexity*, volume LNCS 3104, pages 79–88, 2004.

[5] R. Cohen and D. Peleg. Local Algorithms for Autonomous Robot Systems. In *Proc. of the 13th Colloquium on Structural Information and Communication Complexity*, volume LNCS 4056, pages 29–43, 2006.

[6] X. Défago and A. Konagaya. Circle Formation for Oblivious Anonymous Mobile Robots with No Common Sense of Orientation. In *Workshop on Principles of Mobile Computing*, pages 97–104, 2002.

[7] E. W. Dijkstra. *Selected Writings on Computing: A Personal Perspective*, pages 34–35. Springer, New York, 1982.

[8] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Hard Tasks for Weak Robots: The Role of Common Knowledge in Pattern Formation by Autonomous Mobile Robots. In *Proc. 10th International Symposium on Algorithm and Computation*, pages 93–102, 1999.

[9] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Pattern Formation by Autonomous Robots Without Chirality. In *Proc. 8th Int. Colloquium on Structural Information and Communication Complexity*, pages 147–162, June 2001.

[10] N. Heo and P. K. Varshney. A Distributed Self Spreading Algorithm For Mobile Wireless Sensor Networks. In *In Proceedings IEEE Wireless Communication and Networking Conference*, volume 3, pages 1597–1602, 2003.

[11] A. Howard, M. J. Mataric, and G. S. Sukhatme. An Incremental Self-deployment Algorithm for Mobile Sensor Networks. *Autonomous Robots*, 13(2):113–126, 2002.

[12] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile Sensor Network Deployment Using Potential Fields: A Distributed, Scalable Solution to The Area Coverage Problem. In *In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS'02)*, pages 299–308, 2002.

[13] B. Katreniak. Biangular Circle Formation by Asynchronous Mobile Robots. In *Proc. of the 12th Int. Colloquium on Structural Information and Communication Complexity*, 2005.

[14] L. Loo, E. Lin, M. Kam, and P. Varshney. Cooperative Multi-Agent Constellation Fonnation Under Sensing and Communication Constraints. *Cooperative Control and Optimization*, pages 143–170, 2002.

[15] G. Prencipe. The Effect of Synchronicity on the Behavior of Autonomous Mobile Robots. *Theory of Computing Systems*, 38:539–558, 2005.

[16] S. Samia, X. Défago, and T. Katayama. Convergence Of a Uniform Circle Formation Algorithm for Distributed Autonomous Mobile Robots. In *In Journés Scientifiques Francophones (JSF), Tokio, Japan*, 2004.

[17] K. Sugihara and I. Suzuki. Distributed Algorithms for Formation of Geometric Patterns with Many Mobile Robots. *Journal of Robotics Systems*, 13:127–139, 1996.

[18] I. Suzuki and M. Yamashita. Distributed Anonymous Mobile Robots: Formation of Geometric Patterns. *Siam J. Computing*, 28(4):1347–1363, 1999.

[19] O. Tanaka. Forming a Circle by Distributed Anonymous Mobile Robots. Technical report, Department of Electrical Engineering, Hiroshima University, Hiroshima, Japan, 1992.

[20] G. Wang, G. Cao, and T. La Porta. Movement-assisted Sensor Deployment. In *In Proceedings of IEEE INFOCOM*, volume 4, pages 2469–2479, 2004.

[21] Y. Zou and K. Chakrabarty. Sensor Deployment and Target Localization in Distributed Sensor Networks. *ACM Transactions on Embedded Computing Systems*, 3(1):61–91, 2004.