Title:	Uniform covering of rings and lines by memoryless mobile sensors
Name:	Paola Flocchini[1]
Affil./Addr.	School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada
Keywords:	Uniform deployment; Scattering; Barrier covering
SumOriWork:	2008; Flocchini, Prencipe, Santoro 2008; Cohen, Peleg

Uniform covering of rings and lines by memoryless mobile sensors

Paola Flocchini[1]

School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada

Years aud Authors of Summarized Original Work

2008; Flocchini, Prencipe, Santoro
2008; Cohen, Peleg

Keywords

Uniform deployment; Scattering; Barrier covering

Problem Definition

The Model: A *mobile robotic sensor* (or simply *sensor*) is modelled as a computational unit with sensorial capabilities: it can perceive the spatial environment within a fixed distance $V > 0$, called *visibility range*, it has its own local working memory, and it is capable of performing local computations [6; 7].

Each sensor is a point with its own local coordinate system, which might not be consistent with the ones of the other sensors. The sensor can move in any direction, but it may be stopped before reaching its destination, e.g. because of limits to its motion energy; however, it is assumed that the distance traveled in a move by a sensor is not infinitesimally small (unless it brings the sensor to its destination).

The sensors have no means of direct communication to other sensors. Thus, any communication occurs in a totally implicit manner, by observing the other sensors' positions. Moreover, they are *autonomous* (i.e., without a central control) *identical* (i.e., they execute the same protocol), and *anonymous* (i.e., without identifiers that can be used during the computation).

The sensors can be *active* or *inactive*. When *active*, a sensor performs a *Look-Compute-Move* cycle of operations: it first observes the portion of the space within its visibility range obtaining a snapshot of the positions of the sensors in its range at that time (*Look*); using the snapshot as an input, the sensor then executes the algorithm to determine a destination point (*Compute*); finally, it moves towards the computed destination, if different from the current location (*Move*). After that, it becomes *inactive* and stays idle until the next activation. Sensors are *oblivious*: when a sensor becomes active, it does not remember any information from previous cycles.

Depending on the degree of synchronization among the cycles of different sensors, three sub-models are traditionally identified: *synchronous*, *semi-synchronous*, and *asynchronous*. In the *synchronous* (FSYNC) and in the *semi-synchronous* (SSYNC) models, 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. In FSYNC, at each clock tick all sensors become active, while in SSYNC some sensors might not be active in each cycle. In the *asynchronous* model (ASYNC), there is no global clock and the sensors do not have a common notion of time. Furthermore, the duration of each activity (or inactivity) is finite but unpredictable. As a result, sensors can be seen while moving, and computations can be made based on obsolete observations.

The Problem: The (distributed) *uniform covering* problem refers to sensors, randomly dispersed in a *bounded* region of space, that must scatter themselves throughout the region so to "cover" it satisfying some optimization criteria. Consider the case of a circular rim $\mathcal{R}$ (i.e., a ring), and let $S = \{s_0, \ldots, s_{n-1}\}$ be the sensors initially arbitrarily placed in different points on $\mathcal{R}$, with s_i preceding s_{i+1} clockwise (the index operations are modulo n). We emphasize that these names are used for presentation purposes only, and are not known to the sensors. If the sensors agree on the notion of clockwise, we say that they have a *common orientation.* Let $d = L_{\mathcal{R}}/n$ where $L_{\mathcal{R}}$ is the length of the ring. In the following, unless otherwise stated, the sensors are assumed to have visibility range $V \geq 2d$. Let $d_i(t)$ be the distance between sensors s_i and s_{i+1} at time t; when no ambiguity arises, we shall omit the time and simply indicate the distance as d_i. The sensors are said to have reached an *Exact Uniform Covering* (*Exact Covering* for simplicity) at time t if $d_i(t) = d$ for all $0 \leq i \leq n-1$. Given $\epsilon > 0$, the sensors are said to have reached an ϵ-*Approximate Covering* at time t if $d - \epsilon \leq d_i(t) \leq d + \epsilon$ for all $0 \leq i \leq n-1$.

Key Results

The Ring

Exact Uniform Covering. There is a strong impossibility result that stresses the importance of having common orientation. If the sensors have only a local notion of left and right, but do not share a *common orientation* of the ring, the exact covering problem is unsolvable. This result holds even if the sensors had unbounded memory and visibility, and under a SSYNC scheduler.

Theorem 1. *[5] Let the sensors be on a ring $\mathcal{R}$. In absence of common orientation, there is no deterministic exact covering algorithm even if the sensors have unbounded persistent memory, their visibility range is unlimited, and the scheduling is* SSYNC.

To see why this is the case, consider the following setting. Let n be even; partition the sensors in two sets, $S_1 = \{s_1, \ldots, s_{n/2}\}$ and $S_2 = S \setminus S_1$, and place the sensors of S_1 and S_2 on the vertexes of two regular $(n/2)$-gons on $\mathcal{R}$, rotated of an angle $\alpha < 360°/n$. Furthermore, all sensors have their local coordinate axes rotated so that they all have the same view of the world. In other words, the sensors in S_1

share the same orientation, while those in S_2 share the opposite orientation of $\mathcal{C}$. If activating only the sensors in S_1, an exact covering (resp. *no* exact covering) on $\mathcal{R}$ is reached at time step t_{i+1}, then the same is true also activating only the ones in S_2. Clearly, in such a case, activating both sets no exact covering would be reached at time step t_{i+1}, and the system would be an analogous configuration as the one of time step t_i, with different angles. Using this property, it is easy to design an adversary that will force any algorithm to never succeed in solving the problem; its behaviour would be as follows: (i) If activating only the sensors in S_1 (resp. S_2) no exact covering on $\mathcal{R}$ is reached, then activate all sensors in S_1 (resp. S_2), while all sensors in S_2 (resp. S_1) are inactive; (ii) otherwise, activate all sensors. Go to (i).

On the other hand, assuming common orientation and knowledge of the final inter-distance d among sensors, a simple algorithm that solves the exact covering in ASYNC is for each sensor to move toward the point at distance d from its clockwise successor (if visible). We remind that $V \geq 2d$.

Protocol RINGCOVERINGEXACT (for sensor s_i)
Assumptions: Orientation, knowledge of d.

1. If s_{i+1} is not visible, move distance d clockwise.
2. else, if $d_i > d$ move toward point x at distance d from s_{i+1}.

Theorem 2. *[5] The exact covering of the ring problem is solvable in* ASYNC, *with common orientation and knowledge of the final inter-distance.*

Approximate Covering. Assuming Common Orientation but no knowledge of the final inter-distance among sensors, an ϵ-approximate covering is still possible for any $\epsilon > 0$, but no exact covering algorithm is known. Also this algorithm is very simple: the sensors asynchronously and independently *Look* in both directions, then they position themselves in the middle between the closest observed sensors (if any). Correctness is shown by proving that the minimum distance between any two neighbouring sensors eventually grows, while the maximum distance eventually shrinks in such a way that there is a time when all sensors are within $d \pm \epsilon$ distance.

Theorem 3. *[5] The approximate covering of the ring problem is solvable in* ASYNC *with common orientation.*

Algorithm RINGCOVERINGAPPROX (for sensor s_i)
Assumptions: Orientation

- If no sensor is visible clockwise (resp. counterclockwise), let $d_i = V$ (resp. $d_{i-1} = V$).
- If $d_i \leq d_{i-1}$ do not move.
- If $d_i > d_{i-1}$ move distance $\frac{d_i+d_{i-1}}{2} - d_{i-1}$ clockwise.

Note that the covering problem has been also studied in discrete rings [4].

The Line

The case of a line segment is quite different from the one of the ring and, perhaps surprisingly, it is not easier. Let $S = \{s_0, \ldots, s_{n-1}\}$ be the sensors initially arbitrarily placed in different points on a line $\mathcal{L}$ with s_0 and s_{n-1} being two special immobile sensors delimiting the segment to be covered, and with s_i preceding s_{i+1} $(0 < i < n-2)$. Let $d = L_{\mathcal{L}}/(n-1)$, where $L_{\mathcal{L}}$ denotes the length of the segment. *Exact Covering* and ϵ-*Approximate Covering* are defined analogously to the case of the ring.

Exact Uniform Covering. With Common Orientation and known final inter-distance, an algorithm has been recently shown for oriented sensors in ASYNC [3]. The algorithm works even if the visibility range is just enough to sense the final inter-distance ($V = d$). Let $\delta \leq \frac{d}{2}$ be a fixed positive (arbitrarily small) constant the sensors agree upon.

Protocol CORRIDORCOVERINGEXACT (for sensor s_i)
Assumptions: Orientation, knowledge of d, $V = d$

- If s_{i-1} is not visible, move distance $\frac{d}{2}$ to the left.
- else, let $a := d - d_{i-1}$
  If $d_i \geq d$ and $a > 0$, move distance $min(\frac{d}{2} - \delta, a)$ to the right.

Theorem 4. *[3] The exact covering of the line problem is solvable in* ASYNC *with common orientation and knowledge of the final inter-distance.*

With fixed visibility, a distributed algorithm has been proposed for FSYNC in a discrete setting, to solve the slightly different problem of barrier coverage [2].

Approximate Covering. Approximate covering has been studied in a slightly different visibility model where each sensor is able to perceive up to the next sensor on the line [1]. In other words, in each direction, a sensor sees the closest sensor (if it exists), regardless of its distance, but its visibility is blocked by it (*neighbour visibility*). For presentation purposes, a global linear coordinate system (not known to the sensors) is used here with $s_0(t) = 0$ and $s_{n-1}(t) = 1$. For the sensors to be spread uniformly, sensor s_i should then occupy position $\frac{i}{n-1}$. The following is a simple approximate covering algorithm.

Protocol CORRIDORSPREAD (for sensor s_i)
Assumptions: SSYNC, neighbour visibility

- If no sensor is visible in either direction, do nothing.
- Otherwise, move toward point $x = \frac{1}{2}\ (s_{i+1} + s_{i-1})$.

The idea of the convergence proof in FSYNC is sketched below. Let $\mu_i[t]$ be the *shift* of the s_i's location at time t from its final position. According to the protocol, the position of sensor s_i changes from $s_i(t)$ to $s_i(t+1) = \frac{1}{2}\ (s_{i-1}(t) + s_{i+1}(t))$ for $1 \leq i \leq n-2$, while sensors s_0 and s_{n-1} never move. Therefore, the shifts changes with time as $\mu_i[t+1] = \frac{1}{2}\ (\mu_{i+1}[t] + \mu_{i-1}[t])$. Considering the *progress* measure: $\psi[t] = \Sigma_{i=1}^{i=n}\ \mu^2[t]$, it can be shown that $\psi[t]$ is a decreasing function of t unless the sensors are already equally spread; more precisely, it is shown that every $O(n^2)$ cycles, $\psi[t]$ is at least halved thus reaching approximate covering. More complex but analogous reasoning is followed for SSYNC.

Theorem 5. *[1] The approximate covering of the line problem is solvable in* SSYNC *with* neighbour visibility.

With a simple modification of the algorithm, the result above can be extended to any fixed visibility $V > d$, provided that d is known, as described below [3].

Protocol CORRIDORSPREAD2 (for sensor s_i)
Assumptions: SSYNC, d known, $V > d$

- If only one sensor $s_j \in \{s_{i+1}, s_{i-1}\}$ is visible to s_i and $d' = dist(s_i, s_j) < d$: move distance $\frac{d-d'}{2} + \frac{V-d}{2}$ away from s_j
- If both s_{i+1}, s_{i-1} are visible and $d_1 = dist(s_{i-1}, s_i) < d_2 = dist(s_{i+1}, s_i)$ (resp. $d_1 = dist(s_{i+1}, s_i) < d_2 = dist(s_{i-1}, s_i)$): move $\frac{d_2-d_1}{2}$ toward s_{i+1} (resp. toward s_{i-1})

Applications

Uniform Covering problems are important in many applications; covering of a circular rim occurs, for example, when the sensors have to surround a dangerous area and can only move along its outer perimeter. On the other hand, coverings of the line (often called *barrier coverings*) guarantee that any intruder attempting to cross the perimeter of a protected region (e.g., crossing an international border) be detected by one or more of the sensors. These problems are studied under a variety of assumption; the majority of the studies uses sensors provided with memory, explicit communication devices, global localization capabilities (eg GPS), and centralized approaches. The advantage of memoryless sensors are self-stabilization and tolerance to loss of sensors, the use of local coordinate systems has clear advantages over the full strength of a GPS; finally, decentralized solutions offer better fault-tolerance.

Open Problems

It is known that the exact covering of the ring is impossible without orientation in SSYNC, but the impossibility does not extend to FSYNC where, however, no algorithm is known. Moreover, the only existing exact covering algorithm in ASYNC assumes orientation, which is needed, and knowledge of the inter distance d, which is possibly not needed, so a tighter result might be possible. Finally, approximate covering is achieved in the ring in SSYNC assuming orientation, which is not shown to be necessary, furthermore, no solution exists for ASYNC.

In the case of the line: the only impossibility result for exact covering [3] holds for fully disoriented sensors (not even able to locally distinguish between their two directions), and with small visibility range $V = d$. As for approximate covering, the only known result in this model is for SSYNC, and it is not known whether an algorithm exists for the ASYNC model.

Recommended Reading

1. Cohen R, Peleg D (2008) Local spreading algorithms for autonomous robot systems. Theoretical Computer Science 399:71–82
2. Eftekhari Hesari M, Kranakis E, Krizanc D, Morales Ponce O, Narayanan L, Opatrny J (2013) Distributed algorithms for barrier coverage using relocatable sensors. In: ACM Symposium on Principles of Distributed Computing (PODC), pp 383–392
3. Eftekhari Hesari M, Flocchini P, Narayanan L, Opatrny J, Santoro N (2014) Distributed barrier coverage with relocatable sensors. In: 21th Int. Colloqium on Structural Information and Communication Complexity (SIROCCO)
4. Elor Y, Bruckstein AM (2011) Uniform multi-agent deployment on a ring. Theoretical Computer Science 412:783–795
5. Flocchini P, Prencipe G, Santoro N (2008) Self-deployment algorithms for mobile sensors on a ring. Theoretical Computer Science 402(1):67–80
6. Flocchini P, Prencipe G, Santoro N (2011) Computing by mobile robotic sensors. Chapter 21 in *Theoretical Aspects of Distributed Computing in Sensor Networks*, S. Nikoletseas and J. Rolim Eds, Springer, ISBN 978-3-642-14849-1
7. Flocchini P, Prencipe G, Santoro N (2012) Distributed Computing by Oblivious Mobile Robots. Morgan & Claypool