DACYCLEM: A decentralized algorithm for maximizing coverage and lifetime in a mobile wireless sensor network

Jean-Matthieu Etancelin, André Fabbri, Frédéric Guinand, Martin Rosalie

DACYCLEM: a Decentralized Algorithm for Maximizing Coverage and Lifetime in a Mobile Wireless Sensor Network

Jean-Matthieu Etancelin[a], André Fabbri[b], Frédéric Guinand[c,d], Martin Rosalie[e,f,g,*]

[a] *Université de Pau et des Pays de l'Adour, LMAP, IPRA, UMR5142, F-64013, France.*
[b] *Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France.*
[c] *Normandie Univ. UNIHAVRE LITIS lab, Le Havre, France*
[d] *Cardinal Stefan Wyszynski University in Warsaw, Warsaw, Poland*
[e] *SnT, University of Luxembourg, Luxembourg*
[f] *Univ. Perpignan Via Domitia, Laboratoire Génome et Développement des Plantes, UMR5096, F-66860, Perpignan, France*
[g] *CNRS, Laboratoire Génome et Développement des Plantes, UMR5096, F-66860, Perpignan, France*

Abstract

The goal of Mobile Wireless Sensor Networks (M-WSN) is to sense a specific environment. A commonly considered objective is to organize the work of the sensors such that they monitor the environment as long as possible and cover a surface as large as possible. While most of the time this problem is formulated as a multi-objective optimization problem we present a new decentralized approach for building a connected dominating set (CDS) coupled with attractive and repulsive forces for the movement of sensors in order to maintain the network connectivity. The approach is implemented as a hybrid decentralized algorithm: DACYCLEM (Decentralized Algorithm under Connectivity constraint with mobilitY for Coverage and LifEtime Maximization). The lifetime and the coverage achieved by our approach are the results of the local interactions between the sensors and were not obtained by the application of a direct optimization method. We also introduce a new metric, the speed of coverage, to evaluate the balance between coverage and lifetime. Finally, our simulation results show that one single parameter of DACYCLEM is responsible for the balancing between coverage and lifetime.

*Corresponding author

Email addresses: `jean-matthieu.etancelin@univ-pau.fr` (Jean-Matthieu Etancelin), `andre.fabbri@universite-lyon.fr` (André Fabbri), `frederic.guinand@univ-lehavre.fr` (Frédéric Guinand), `martin.rosalie@univ-perp.fr` (Martin Rosalie)

URL: `https://lma-umr5142.univ-pau.fr/en/~jmetancelin` (Jean-Matthieu Etancelin), `http://litis.univ-lehavre.fr/~guinand` (Frédéric Guinand), `https://perso.univ-perp.fr/mrosalie/` (Martin Rosalie)

Keywords: WSN, coverage, connectivity maintenance, decentralized algorithm, lifetime maximization

1. Introduction

Robotics is booming nowadays and the future seems to strive for the design of more and more autonomous and adaptive robots. In a near future, we expect them to cope with unpredictable environments without any external help. An efficient and robust approach to achieve it is to involve robots within self-organized and self-coordinated teams in order to fulfill some predefined objectives or eventually take new ones into account after their deployment. When they are equipped with sensors they form a system that can be considered as a mobile wireless sensor network (M-WSN). Current researches in the field are facing different issues: mobility, energy saving, topology control, connectivity maintenance, obstacle detection and avoidance or fault tolerance, to name a few [1, 2, 3, 4, 5]. The problem addressed in this paper is made up several sub-problems among these. It consists in maximizing the lifetime of the network while attempting to cover an area as large as possible under connectivity constraints.

Thus, connectivity and coverage are two crucial elements in the problem we address in this work. In our context, coverage may be considered as an objective and connectivity as a constraint. They both depend on several parameters and in a recent survey, Boukerche and Sun [6], propose to classify the main algorithms and methods for solving issues coping with both connectivity and coverage.

However, in our work lifetime also plays a central role. As underlined by Dietrich and Dressler in [7], many different definitions of the term lifetime are present in the literature. Three main elements underlie these definitions: the number of alive nodes, the sensor coverage and the network connectivity. In most cases, the lifetime limit is achieved when one (or a combination) of those elements drops under a given threshold. According to their survey, our definition of the network lifetime refers mainly to the notion of network connectivity. In their work the authors note that mobility may play a key role in the evaluation of the lifetime and that it should be mentioned within any relevant definition of the lifetime, while they also note that most works consider stationary sensor nodes.

In our work mobility is taken into account, and at any moment, sensors are either active or idle (from a sensing point of view). The connectivity constraint applies both to the network restricted to active sensors and to the network composed of all the sensors. Dietrich and Dressler also introduced the notion of "graceful degradation" allowing some criteria to be fulfilled during non continuous time intervals. In the context of this work, the loss of connectivity signals the end of the lifetime of the network, irrespectively

of the number of alive sensors, of the coverage value or of the size of the resulting connected components, so we do not consider "graceful degradation" at that stage. It should be noted that when one sensor disappears from the network (for instance it runs out of battery) but does not entail a connectivity loss, the network is still alive.

The problem we address starts from a random distribution of mobile sensors within the environment. The first step aims at building a connected WSN gathering all the sensors. Then, the proposed method aims at maintaining as long as possible the connectivity constraints of the network, while covering an area as large as possible. However, the aim of the current work is not to propose another optimization method but to present a new approach for addressing this problem by focusing on the behavior of the sensors instead of improving the value of some specific objective functions. Thus, no objective function is explicitly considered by the sensors in their decision process and the resulting coverage and lifetime of the network are obtained as the result of the collective behavior of the sensors. The optimization, as long as this term can still be considered relevant in this context, is purely implicit. The main advantages that we can expect from such an approach is a better robustness in case of failure and a better scalability since all decisions are taken at the sensor level and are only based on local information and on interactions with immediate neighbors. This type of approach is not new. In 2012 a decentralized approach for modeling the swarming behavior of bees considering the lack of information transmission between the members was proposed in [8]. Their work also illustrates that a group can achieve a global coherent behavior while all the members do not share the same objectives. The seminal work of Heinzelman et al. in 2002 [9] also relies on this paradigm but without taking mobility into account.

Our aim is to show that connectivity, which is the basic property of a WSN for routing data from sensors to sinks, can be achieved and maintained implicitly by taking advantage of the mobility capability of sensors. For that purpose, each sensor has to manage carefully its movement and state in order to save its battery while participating in the network for connectivity maintenance. The behavior of sensors is implemented as a decentralized algorithm (DACYCLEM: Decentralized Algorithm under Connectivity constraint with mobilitY for Coverage and LifEtime Maximization) executed on each sensor. We first show that such an indirect approach can provide acceptable solutions to the considered problem. We also show that changing the behavior of the sensors, by modifying one parameter of the algorithm, entails a modification in the balance between network lifetime and coverage.

This paper is organized as follows. In Section 2, the problem definition is carefully exposed together with the constraints and the implicit objectives. Section 2.1 presents a state of the art describing various configurations of M-WSN: it underlines the new issues and opportunities entailed by sensors mobility. Section 3 reviews existing researches on similar issues and posi-

tions our work within the field. The model, the method for addressing the problem and the resulting DACYCLEM algorithm are detailed in Section 4. We also derived two systems from the generic method for determining boundaries: Time System for improving the lifetime and Area System that focuses on the area covered by the sensors. In Section 5, we evaluate the method developed through computer simulations and compare the results with the boundary systems. This evaluation is performed by means of a new metrics: the speed of coverage that highlights a relation between the covered sensing area and network lifetime. Thanks to this metric, we are able to customize our algorithm in order to promote either coverage or survival time of DACYCLEM. The last Section draws a conclusion of this work and open a discussion on possible future works and improvement.

2. Problem description

The coverage and the connectivity of a sensor network are two problems that have been extensively studied separately. The combination of both problems, as addressed in this paper, brings up new issues. Considering both objective separately or combined may lead to different solutions as it was underlined in the work of Dietrich and Dressler under the term "connected coverage" [7]. While many works have been devoted to the joint problem of coverage and connectivity in WSN [10, 11] this number is much more limited as soon as mobility is taken into account.

Next section (Sec. 2.1) is a quick survey of the literature dedicated to the combination of both coverage and connectivity problem in M-WSN. In a second part (Sec. 2.2) we describe, more formally and with more details, the problem addressed in this paper.

2.1. Related problems

As the main purpose of M-WSN is to collect, monitor and record data, mobility can be considered as an asset. However, this capability induces potential issues, especially on stability, security, and reliability of WSNs [12]. Mobility entails issues for the communication within the network since the topology may change when sensors are moving. The survey of Li *et al.* [13] focuses on the problem of topology control in WSN. Their work underlines that most existing topology control techniques are divided into two categories depending on which metric drives the whole process: network coverage or network connectivity. They conclude their review giving some insights for improving global WSN performances and, balancing coverage and connectivity appears as one of the proposed design guidelines.

Recent surveys address this problem [14, 15] and define mandatory requirements to obtain efficient protocols. Appropriate routing protocols may take into account the lifetime of the network: Chen [16] proposes a protocol with data aggregation that guarantees a Quality of Service (QoS) while it

saves energy. Recently, Yue *et al.* [12] present a survey on M-WSN and compare the time delays, the network size, the energy efficiency, the scalability and the reliability of algorithms for routing protocols. They provide the future trends to consider for developing such algorithms.

As the sensors move, the coverage of a network becomes vital to ensure the reliability of the system. Regarding to the geometrical concerns of the coverage in WSN, Sangwan and Singh [17] realize a survey on this topic. Recently, Fang *et al.* [18] propose two deployment algorithms for M-WSN. They use Voronoi diagrams to find blind-zone in the coverage of the network and the following metrics to evaluate their algorithm: coverage percentage, moving distance, deployment time, uniformity, connectivity and a Testbed-based multi-metric quality measurement (T-MQM). To evaluate the coverage Liu *et al.* [19] define a dynamic coverage metric to detect an intruder with a M-WSN: they propose a coverage metric to consider the persistence of the coverage while the sensors are moving. Banimelhem *et al.* [20] control of a mobile robot to handle uncovered area of the WSN (only a part of the network is mobile) using fuzzy logic in order to maximize the coverage.

To our best knowledge, only few works address simultaneously the coverage,the connectivity and the energy management for a M-WSN. For instance, Rebai *et al.* [21] consider coverage and connectivity constraints for a WSN (without mobility). They define an optimization problem solved with a linear programming model and the energy management aspect is not considered. Abo Zahhad *et al.* [22] use a Multi-objective algorithm to solve the coverage problem. Their algorithm aims to cover the whole area by finding the minimum movement to assign at each sensor; but, though, the energy aspect is not considered. Reina *et al.* [23] propose an evolutionary algorithm to manage UAVs. With several populations in a genetic algorithm, the UAVs are organized using layout to satisfy connectivity and coverage constraints. However, this metaheuristic requires a global knowledge to select population during the evolutionary algorithm. Finally, the closest problem we report here is a connected target coverage problem. Recently, Roselin *et al.* [24] consider this problem of coverage of critical points. Their algorithm, named *energy efficient connected coverage* consider remaining energy, coverage and connectivity aspects to ensure QoS. However, this problem focus on the coverage of critical points contrary to our problem without such specific area to cover.

2.2. Problem definition

In this paper, we consider each robot as an autonomous wireless mobile sensor with a non-rechargeable battery and the ability to change its state: active or standby (idle). If a sensor is active, it collects data, otherwise, it saves its battery. Therefore, a sensor has to manage his movements and his state in order to aim at the following objectives: increase the total

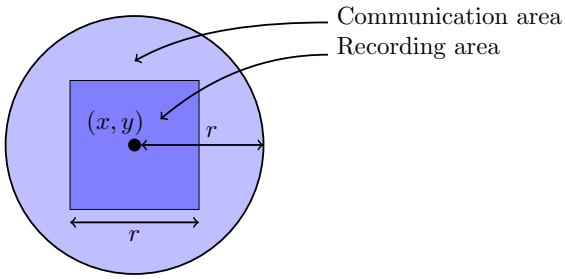


Figure 1: Sensor characteristics.

covered area (being in active state) and increase the lifetime of the sensor network. We consider battery operated sensors with an initial battery level. These objectives have to be fulfilled with the following constraints: all the sensors need to be connected to ensure good communication, referenced as the communication constraint; and the sensors have to maintain a connected active sensors network, referenced as the connectivity constraint.

Sensor characteristics are detailed as follows. Sensors are located on a time-dependent two dimensional environment by means of coordinates (x, y). Each sensor has a time-dependent battery level (b), a fixed wireless emission range (r) and a velocity (v). We assume that the communication area is circular of radius r. This area is the local eyesight of the environment from a sensor viewpoint (Fig. 1). The sensing area (R) is taken r side square. Each sensor can change its state between *active* and *standby* which corresponds to an active sensing $(R = r^2)$ or not $(R = 0)$. If the sensor is active, then it collects data and the area is considered as covered. Sensing area overlapping is not taken into account.

The problem is formulated under several hypothesis:

- simulation time is discrete
- sensors within range of communication are able to exchange information about their state, battery level and position
- sensors can detect parts of their communication area already recorded by other sensors
- state change costs are negligible in terms of energy consumption
- battery is decremented at each step depending on the activity of the sensor
- environment is considered as a squared area
- recording area is a square centered on the sensor regardless of the sensor position

- network formed by all the sensors must stay connected throughout network life as well as the network formed by all active sensors
- sensors initial position is random as well as the initial level of battery.

In accordance with these hypothesis, we introduce a distributed algorithm to foster, both, the recording area and the lifetime of the whole network. Locally, sensors adjust their position in a two dimensional space in order to aim at these two objectives satisfying the connectivity constraints. The proposed method is based on the decentralized management of the sensors moves and states regarding their battery level and their neighbors positions.

3. Related works

The authors of [25] provide a classification of the WSN with respect to their abilities: sensor-type, deployment strategy, network architecture, mobility, coverage and sensing. According to their taxonomy (see Fig. 1 of [25]), our problem belongs to:

- Sensor Type: mobile (M-WSN)
- Deployment Strategy: random
- Network Architecture: flat
- Mobility: intentional
- Coverage: type (partial)
- Sensing: binary disk sensing model

With respect to this classification there is no existing work matching exactly our hypotheses and objectives. However, our contributions are related to several areas highlighted by the authors of the survey. First, the initial random deployment may not ensure the connectivity of the network, thus the first step of our work consists in moving the sensors in order to obtain a connected networks. In Section 3.1 this problem is briefly discussed. Then, we will consider each single objective individually: maximizing the lifetime of the network and maximizing its coverage. During the lifetime of the network some sensors can run out of battery. The system is considered to remain operational as long as the connectivity constraint is satisfied. As a consequence, the lifetime of the network corresponds to the amount of time the whole network is connected. Existing works about both sub-problems are presented in Sections 3.2 and 3.3. Section 3.4 is dedicated to the state-of the art related to the maximization of the coverage.

3.1. Initialization of the network

We assume that sensors are randomly distributed over a given area. The considered network can be modeled as a geometric graph since two sensors are connected if, and only if, they are located in their respective communication range. Considering our assumptions , large area, limited communication range and limited number of sensors, it is unlikely that the original network is connected [26]. Thus, our first aim is to find a solution for connecting all the sensors. Minimizing the number of movements for achieving connectivity is an NP-hard problem for which it is possible to provide approximation algorithms [27]. However these approximation methods are all centralized. In our context we can only rely on decentralized processes, and we thus choose a naive solution that consists in defining a *meeting point* arbitrarily chosen before the deployment of all sensors. This is the only global information used within our method. Then, the movements are computed to reach this point for all connected components of the graph.

3.2. Connectivity insurance

Due to the connectivity constraint, we need to keep the network connected at each moment. Therefore, the dynamic model must guarantee this connectivity. It exists several methods to control sensors movements keeping connectivity from a fully connected initial state of the network. Zavlanos & Pappas [28] proposed a centralized method based on linear programming to minimize an energy function: this method requires a linear solver.

In an algebraic sense, Kim & Mesbahi [29] solve this problem by studying the second smallest eigenvalue of a state-dependent Laplacian graph. More recently, this problem has been solved by local methods [30, 2]. These methods are based on a discussion protocol between sensors in order to manage connection losses. These methods need a synchronization protocol to avoid multiple simultaneous deletion of links. In these two references, the network is made up of a leader governing the global movement. Although the decentralized aspect of these methods is interesting, it seems tedious to implement though. Moreover, as presented in their results, the solution makes the robots connected along a chain and this kind of topology could lead to a weak solution for our problem. Another distributed method was proposed by Silvestri and Goss [31] for the k-barrier-coverage problem. It consists in building, in a distributed way, k connected line of sensors between two borders and the whole network has to be also connected. Their deployment algorithm, called MobiBar, uses some localization information stemmed from low cost GPS devices thus with possible bounded errors. The algorithm is composed of four main tasks that are executed by each sensor in an asynchronous way. While their work does not consider explicitly the maximization of the lifetime, they notice that the movements are locally optimized in order to enhance energy consumption. As pointed out in [32], the

connectivity can be broken because of loosing sensors. The authors propose a reinforcement of the network by forcing the network to be bi-connected. So there exists always two different links between any pair of sensors. Therefore, the network is still connected in case of losing a sensor. This method gives delay to deal with sensors loses.

Finally, Couzin *et al.* [33] introduced swarm behavior methods that is very well adapted to our problem because of their robustness. Such systems shows a strong cohesion while perturbations occurs. The behavior of each entity makes the group avoid predators or obstacles and keep the group intact.

3.3. Energy management

The main purpose of a standard WSN is to collect information and transmit them to a sink [34]. These transmissions implies numerous problems, for instance, routing information, power of wireless transmission, management of sensor battery.

3.3.1. Static network

Considering a static WSN, Long *et al.* [3] show a hierarchic network where the sensors with the higher level of battery are grouped: they compose the sink. The lower the battery level is, the more distant they are. This hierarchy enables a coverage by capillarity. An other approach introduced by Basagni *et al.* [1] presents a static network where mobile sink move to collect information from sensor. This method finds the best place for the sink in order to collect the information.

3.3.2. Standby

To let the network persisting over time, sensors can standby. Cardei *et al.* [35] propose a centralized method where there are numerous sensors to ensure the coverage of an area. Their method also find redundant sensors where sets of sensors are defined to collect information alternatively. The idea of maximizing the lifetime of the network thanks to scheduling the idle/sensing periods was also reported recently in the work of Roselin et al. [24], the problem addressed in their work is target coverage and as for the previous mentioned work, the number of sensors is redundant allowing a distinct distribution of roles according to their battery level. The proposed method is also centralized. This is also the objective of the SPAN algorithm [36] except that the latter is a decentralized one. Sensors are independent and use a satisfaction function to decide whether standby or not. A modification of this algorithm, named SPAN-CCP [37] enables a coverage maximization. These approaches emphasis a backbone structure composed by active sensors whose purpose is to transmit information. This strategy is also defined in [38] (§11.2.7) where a way to save energy is to build an active line of communication nodes and let standby non-critical nodes.

3.3.3. Mobile network

About MANET, authors of [39] and [40] propose algorithms searching a dominating set. The set determines the topology properties of the network and is used to find a path for transmitting information. Building a backbone corresponds to find a CDS (Connected Dominating Set) because it is a specific set composed by the fewest nodes. Bao & Garcia-Luna-Aceves [41] consider the battery level of the sensors in a decentralized algorithm electing the sensors that have to be part of the backbone. In the same way, Wang *et al.* [42] propose another decentralized algorithm that can build a robust backbone with respect to small position modifications.

Liu & Gupta [43] propose a synthesis of the two previous methods including the fact that a sensor can have two states: active or standby. These algorithms maintain a backbone robust to small variations of topology. In the opposite side, there is an algorithm based on a Steiner tree [44] tending to move sensors for keeping a robust backbone. We point out that finding a MDS (Maximum Dominating Set) in a centralized way is a NP hard problem [43]. Otherwise, all these decentralized algorithms are other ways to solve part of our problem. Shi *et a.* [45] proposed a new problem for energy harvest networks (EHNs) including rechargeable nodes named Energy Harvest CDS and they proved that this problem in NP hard. The authors present centralized and decentralized algorithms to solve this problem and some variants where mobile nodes can replace nodes with lower battery level. Finally, Roselin *et al.* [24] present an energy efficient connected coverage (EECC) algorithm that is a *connected target coverage algorithm* where critical points have to be covered. The authors introduce the notion of *crucial coverage sensors* to ensure the connection from the sensors to the sink. One algorithm find the connectivity to link nodes directly to the sink and another one to find the additional relay node for establishing the overall connectivity. These algorithms rely on heuristics values for coverage and connectivity. According to these values, the node move to a direction until it can satisfy the constraints.

3.4. Maximize the surface

Gage [46] defines three ways to cover an area: the *blanket coverage* is used to maximize the total area covered without any other consideration, the *barrier coverage* focuses on a more restricted area without uncovered hole inside and the *sweep coverage* consists in a robot patrol which sweeps a critic area. According to these definitions, our problem will belong to the *blanket* kind of coverage.

Maximizing the surface while keeping the connectivity could be tackled by many ways. Several articles establish a survey of the methods used to solve this problem [35, 36]. Since we do not have any centralized control, we focus on the robot deployment methods . This section is dedicated a

review of the main methods used to deploy a group of explorer-robots in an unknown area.

3.4.1. Potential fields methods

Potential or vector fields methods comes from the molecular physics domain. As for the particles, the interactions are defined by forces. As an example, two electrons repel each other according to the electrostatic force. The closer are the electrons, the higher is the force magnitude. Thus this force vanishes as the electrons are too far from each other. When several forces apply to the same body, the resultant force draws the body's trajectory according to Newton's second Law :

$$\sum_i \boldsymbol{F}_i = m\boldsymbol{a} \ .$$

In a vector field method, each sensor is considered as a single electron. They repel each other and therefore they will cover a larger area while being close enough to keep the connectivity.

The forces could also be used as a method to generate a social behavior among a group of robots [40]. Some methods even propose to avoid obstacles by adding a repulsive forces to them [47, 4]. This model leads the robots to use bypass strategies as they deploy themselves.

Another method [48] adds a new force $\boldsymbol{F}_{\text{degree}}$ to the electrostatic one in order to ensure that each sensor has at least k neighbors. The method proposed in [49] tackles the problem of the resulting sensor's distribution by adding another force called $\boldsymbol{F}_{\text{boundary}}$. This force will only apply on the boundary of the network to hold the sensor's expansion. As a result the covered area will be more uniform as for a *barrier* kind of coverage.

To restrain the sensor's movements and also the energy consumed, most of the model introduce a friction force. Thus the sensors will avoid oscillating movements and the network will reach easily a static equilibrium [47, 48]. At last, [50] assigns different roles to the sensors. The forces applied to the sensors depend on their role with respect to a specific structure to the network.

3.4.2. Topology based methods

Voronoï diagrams can be used to model the influence of each sensor. We assume that each sensor can be considered as a vertex. For each vertex we define its Voronoï region as the set of points which are closer to this vertex than the others. According to this diagram we could determine if a sensor is correctly placed regarding to its neighbors. If the edges of a region are equally distant to its vertex, this means that the sensor is well placed regarding to its neighbors. Otherwise the furthest edges corresponds to areas which are not well-covered by our network. Thus topology based methods use Voronoï diagram in order to optimize the network coverage. However

the way to move the vertex according to its Voronoï region differs from one method to the other. The VOR and MiniMax methods [50] select the furthest edge at first and then move closer to it as fast as possible. Contrary to the VD-based Deployment Algorithm [51] where each vertex moves closer to the middle of its Voronoï region. Further to this work, these authors also proposed a Grid-Quorum algorithm with Voronoï diagram and cascaded movement to identify and relocate redundant sensors by minimizing the message complexity and the energy consumption [52]. Voronoï diagrams has also been used to identify area without coverage (coverage holes) to relocate sensors for the maximization of the coverage [53]. The authors proposed several approaches to obtain their Voronoï blind-zone: the position of the neighborhood or the target location of the sensors.

3.4.3. Behavioral methods

At last, some methods determine the sensor movement based on behavioral models: each sensor tends to maximize its own satisfaction regarding to its environment. The authors of [54] assume that the satisfaction of each sensor depends on its connectivity. When the neighborhood does not fulfill the sensor' satisfaction, it moves randomly to another place. Contrary to [55] where each sensor tends to maximize its own covered area. According to the method, two sensors are in conflict when they cover the same area. To solve it, the sensors adopt different behaviors from the repulsion to the dance, where one sensor will move around the other.

4. DACYCLEM: Decentralized Algorithm under Connectivity constraint with mobilitY for Coverage and LifEtime Maximization

As we can see in this state of art, the issues addressed in this paper could not be tackled separately. We propose a decentralized method which will involve several techniques presented above in order to maximize the covered area and the lifetime of the network together. Model performances are evaluated regarding systems with restricted single objectives: a surface oriented system and a time oriented system. The method proposed in this article is decentralized with one initial centralized information and avoid broadcast storm between sensors [56].

4.1. Global method

Here we remind that sensors are randomly deployed on the area with an information: the *meeting point*; it is the only centralized information of the method. The battery level is also given randomly at the beginning.

4.1.1. Abilities and objectives

Sensors abilities must be used to fulfill our objectives. We remind that a sensor can move and change its state. Also, the individual objectives of the

sensors are to maximize the covered area, and the network lifetime as well. Firstly, sensors can move toward uncovered areas to enlarge covered surface. They could also move to maximize network lifetime and satisfy connectivity constraints by going toward sensors with low battery level. Here we can see the antagonist effect of moving. From a sensor viewpoint, moving to an uncovered area means going away from other sensors which is in conflict with the other goal. There is an other conflict in an energy sense: moving consume energy and consequently reduce network lifetime (Fig. 2).

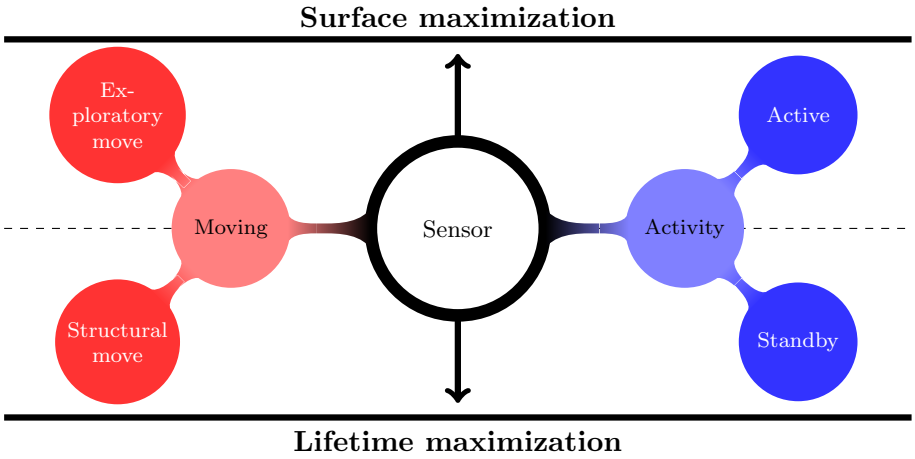


Figure 2: Abilities and objectives.

Secondly, sensors can change their state to save battery (standby mode) which helps to maximize network lifetime. Otherwise, they can change to become active and contribute to the coverage. Here again, this ability leads to two conflicting effects (Fig. 2): being active or standby in order to maximize covered area or network lifetime. On top of that, when a sensor become standby, it is withdrawn from the active sensors network and it also contributes to its weakening. The two sensors abilities and the two main objectives are extremely linked. The following method will deal with theses discrepancies.

4.1.2. Constraints satisfaction

Based on the sensors' activity state: active or standby, two different networks can be defined. The first is composed of the whole set of sensors and the second one is composed only by the active ones. On both networks, the connectivity constraint applies. We detail later, in section 4.2.2, how it is fulfilled for both networks.

4.2. A local method

DACYCLEM is a decentralized method executed by each sensor. In this section after an exposition of our algorithm (Alg. 1) for a sensor at high level

of abstraction we detail each part. First, the method runs phases (steps) during which objective is to build a connected network (**Initialization** 4.2.1). As soon as some sensors are connected and until it remains battery, the sensors move to address the problem (**CDS construction** 4.2.2 and **Mobility** 4.2.3).

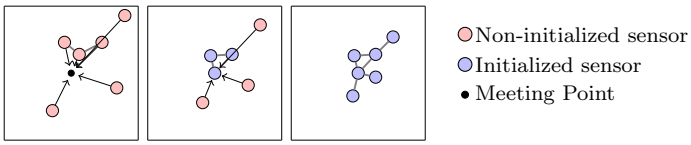
Algorithm 1 DACYCLEM algorithm for a sensor

```
while not initialized do
    moveToMeeting                          ▷ Initialization (4.2.1), Alg. 2
while b > 0 do                             ▷ while it remains battery
    chooseState                            ▷ CDS construction (4.2.2), Alg. 3
    moveWRTForces                          ▷ Mobility (4.2.3), Alg. 4
```

Even if the convergence of the algorithm towards an optimal solution cannot be demonstrated, some comments can be made. With the initialization method, our algorithm is ensured to start properly since the graph of sensors is connected. The algorithm is ensured to stop when all sensors are out of energy because sensors must loose energy even if they stay in a standby state.

4.2.1. Initialization

At the beginning, each sensor is not initialized. A sensor is initialized when it reaches the meeting point or when one of its neighbors is already initialized. To achieve this goal, the sensor moves toward a meeting point initially known by every sensor. An example of this process is given in Fig. 3 chronologically.



(a) t_0 (b) $t_1 > t_0$ (c) $t_2 > t_1$

Figure 3: Example of initialization process.

By this simple method, as detailed on Alg. 2, our random geometric graph is connected in a few steps. Once initialized, each sensor continues the execution of the algorithm for deciding its state and its movement.

Algorithm 2 moveToMeeting: Initialization algorithm

```
Compute movement direction towards the meeting point
Move toward the direction
```

4.2.2. Connected Dominating Set construction

Considering that each sensor applies the same decentralized algorithm, each sensor will decide to be active or standby according to local-only information. We denote by A (and resp. $\bar{A}$) the set of *active* sensors (respectively the set of standby sensors). From the minimizing energy consumption point of view, set A should be as small as possible and should also fulfill the connectivity constraint which corresponds to the notion of Connected Dominating Set (CDS). We denote by B (and resp. $\bar{B}$) the set of sensors part of CDS (resp. the set of sensors not part of CDS). Sensors of the set B are the *backbone* of our network. This CDS is based on MDS (Maximum Dominating Set) with additional sensors to connect them. Thus, we obtain that sensors of the set B are connected to each other and sensors of $\bar{B}$ are directly connected to at least one sensor of B. Sensors of B must be active in order to have a connected network of active sensors and therefore ensure both connectivity constraints, thus $B \subset A$.

To construct the backbone, we use the method described in [41] which consists in building a robust backbone using properties of MDS considering the battery level of sensors. The main advantage of this method is its robustness to little topology modification, in term of battery and movement. The first step of this algorithm is to find the MDS. To be part of the MDS a sensor should have a battery level higher than the battery level of its neighbors or higher than the battery level of the neighbors of all of its neighbors. Such a sensor is called a clusterhead. The following step is to connect the clusterheads to some sensors in order to obtain the Connected Dominating Set. These additional sensors are called doorway and gateway. Thanks to these sensors, two clusterheads that are within three hops from each other are connected by a path in the CDS (see [41] for the proof). Refer to the original research paper [41] for details about the decentralized process executed by each sensor for deciding if it becomes a gateway or a doorway.

This methodology permits to satisfy the connectivity constraints. Finally each sensor of $\bar{B}$ can choose to standby or to activate (part of A or $\bar{A}$): this choice is made either to minimize the use of the energy or to increase global covered area. We define a parameter: the *threshold for recording area* as the percentage of the area that one sensor can record with no overlap with other sensors. If the sensor can record more than this percentage threshold value, then it actives and becomes part of A; else, it chooses to standby and be part of $\bar{A}$. Indeed, each are neighbor of a member of the set B, so, if it comes active, constraints are still enforce. This choice depends of the gain of cover that implies this activation. We set a limit over that the activation seems to be useful to the coverage of the system. We have to take into account that this choice activation may reduce the lifetime of the system. The two parts of this methods are described in Alg. 3.

At this point both connectivity constraints are fulfilled and have to be maintained despite the mobility of sensors. This is achieved thanks to mechanisms described in the next section.

Algorithm 3 chooseState: Choose next state regarding neighborhood and battery level

```
Compute belonging to MDS          ▷ Sensors with high battery level in 2-neighborhood
Compute belonging to CDS          ▷ Sensors between MDS sensors
if I am in CDS then
    State ← Active
else if contribution to recording area > a then
    State ← Active
else
    State ← Standby
```

4.2.3. Mobility: forces and behavioral model

At this point, sensor motion remains to be described but the moves must not disconnect the set of active sensors (set A) as well as the whole network. Thus, sensor moves must depend on their state and their membership to the backbone (set B). Globally, we decide that sensors firstly move towards backbone members with low battery level; otherwise they move away from the backbone to explore new areas. After choosing its state, a sensor will compute its movement regarding its neighbors and more precisely, their distances and states. We first present the force-based model and then the behavioral model to apply these forces.

For two sensors i and j, we define the position of a sensor by $\boldsymbol{X}_i$, the distance between i and j by the Euclidean distance

$$d_{ij} = ||\boldsymbol{X}_i - \boldsymbol{X}_j||^2 \ .$$

Let $\boldsymbol{F}_{ij}$ be a force applying from i to j along the unit vector defined as follow:

$$\boldsymbol{u}_{ij} = \frac{\boldsymbol{X}_j - \boldsymbol{X}_i}{d_{ij}} \ .$$

Interactions are defined by means of the following forces applying only if $d_{ij} < r$ (the communication radius) [47, 49, 40]:

- Attractive, $\boldsymbol{F}_{ij}^A$
- Repulsive, $\boldsymbol{F}_{ij}^R$
- Equilibrium, $\boldsymbol{F}_{ij}^E$.

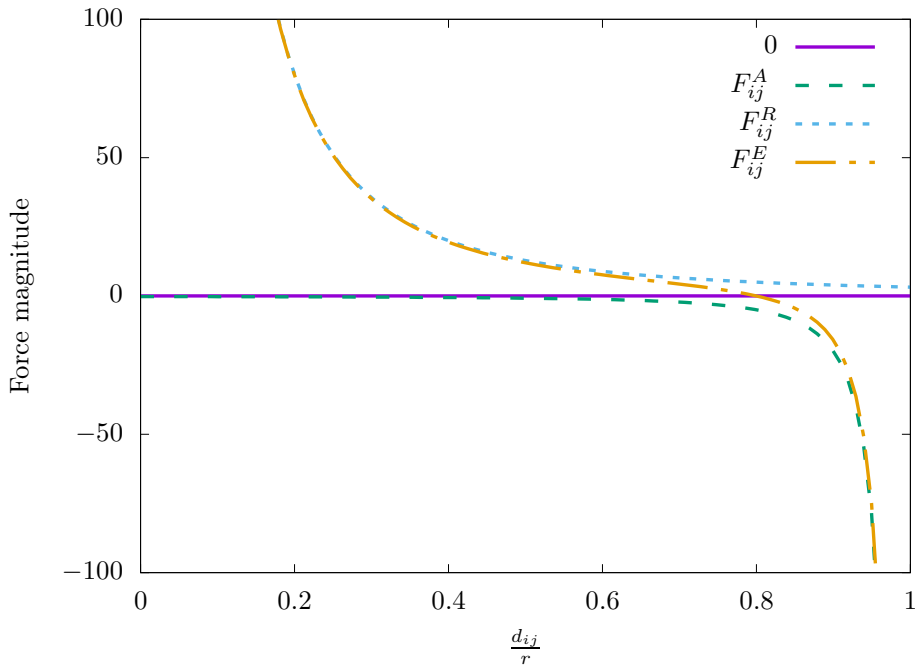


Figure 4: Examples of force behaviors used for the movements with parameters $\gamma = 1$, $\lambda = 0.8$, $\delta = 1$, $\alpha_1 = 2$ and $\alpha_2 = 2$.

The forces have to satisfy a few specifications. Their magnitude depends on the communication radius and on the inverse of the distance between the sensors (as for the electrostatic force). Since the forces take an important part of our system, we decided to moderate them by parameters described as follow and reported in Fig. 4:

- λ: set the normalized distance of equilibrium between the sensors ($0 < \lambda < 1$); the distance where the equilibrium force equals 0
- δ: smooth the curve for the two extremal distances of the communication radius (0 and r) and avoid division by 0
- α_i: strengthen the slope of the curves around the extremal distances of the communication radius to trigger easily a movement at these distances.

The purpose of *attractive force* is to make the sensors closer from each other in order to ensure the overall connectivity of the network. This force needs to be important when the distance between the sensors is close to the communication radius and very small otherwise:

$$\boldsymbol{F}_{ij}^A = -\frac{r(1-\lambda)^{\alpha_1}}{(r+\delta-d_{ij})^{\alpha_1}}\boldsymbol{u}_{ij} \ .$$

Contrary to the attractive one, the *repulsive force* intends to repel the sensors in order to widen the sensing area of the overall network. Similarly to the attractive force, this force needs to be large when the distance between the sensors is close to zero and very small otherwise:

$$\boldsymbol{F}_{ij}^{R} = \frac{r\lambda^{\alpha_2}}{(\delta + d_{ij})^{\alpha_2}} \boldsymbol{u}_{ij} \ .$$

The *equilibrium force* is the sum of the attractive and the repulsive force:

$$\boldsymbol{F}_{ij}^{E} = \boldsymbol{F}_{ij}^{R} + \boldsymbol{F}_{ij}^{A} \ .$$

This force allows each sensor to position itself to the right distance from other sensors. The distance of equilibrium where $\boldsymbol{F}_{ij}^{E} = 0$. This distance should maximize the sensing area and also keep the network connectivity robust to neighbor's movements. That's why the distance of equilibrium does not correspond to the communication radius of the sensor. Actually this force is a trade-off between the two contradictory objectives of our method.

Once a sensor computes all the forces from its neighbors, it also computes the resultant force:

$$\boldsymbol{F}_j = \sum_k \boldsymbol{F}_{k,j}$$

The next movement of the sensor will be determined by its acceleration according to Newton's second law:

$$\boldsymbol{A}_j = \frac{\boldsymbol{F}_j}{m_j} = e_j \boldsymbol{F}_j \tag{1}$$

We define the mass m_j of each sensor by the inverse of his battery level e_j. Thus, a sensor with a low battery level will have a great inertia.

The system described combines a force-based method to maximize the surface with a *behavioral model* to keep the network connectivity (cf. 3.4). According to the sensor's state, we define four different behaviors described in Tab. 1 and hereinafter. The behavior of the sensor will determine which force is applied to its neighbor depending on its own behavior and the behavior of the neighbor.

Backbone. Each sensor who belongs to the backbone has to keep the other sensors to the equilibrium distance in order to cover the maximum surface and also to guarantee the connectivity of the whole, even if one sensor moves (see Section 4.2.2). According to this behavior the sensor will apply an equilibrium force to all its neighbors, whatever their behavior.

Critical backbone. According to the backbone's construction method, the sensors are selected with the maximum battery possible. If one of them has a very low battery level, it means that our network is going to disconnect. We need therefore to adjust the sensor behavior in order to keep the network connected as long as possible. Broadly speaking, the critical sensor has to

	Current sensor state j			
	Backbone	*Critical Backbone*	*Active*	*Standby*
Backbone	$\boldsymbol{F}_{ij}^{E}$	-	$\boldsymbol{F}_{ij}^{E}$	$\boldsymbol{F}_{ij}^{E}$
Critical Backbone	$\mu(e_j)\boldsymbol{F}_{ij}^{A}$	$\mu(e_j)\boldsymbol{F}_{ij}^{A}$	$\mu(e_j)\boldsymbol{F}_{ij}^{A}$	$\mu(e_j)\boldsymbol{F}_{ij}^{A}$
Active	$\boldsymbol{F}_{ij}^{E}$	-	$\boldsymbol{F}_{ij}^{R}$	$\boldsymbol{F}_{ij}^{R}$
Standby	-	-	-	-

Table 1: Behavioral interactions for neighbor sensors i.

find a substitute. The sensor then attracts all its neighbors in order to prevent disconnection. We introduce a function $\mu(e_i)$ in order to increase the virtual mass of sensor and, as a result, reinforce the attractive force it applies to its neighbors (whatever their behavior). A backbone sensor is considered as critical when its battery level is lower than a configured *critical threshold* (c). The reinforcing function equals to $\mu(e_i) = c/e_i$.

Active. The active sensors that do not belong to the backbone behave as *Active*. These sensors will stay at a reasonable distance to a backbone sensor using equilibrium forces in order to comply with the connectivity constraint. If the latter is critical, the sensors will follow its incentive. Else, the sensor will try to repel all its neighbors in order to maximize the covered surface [47].

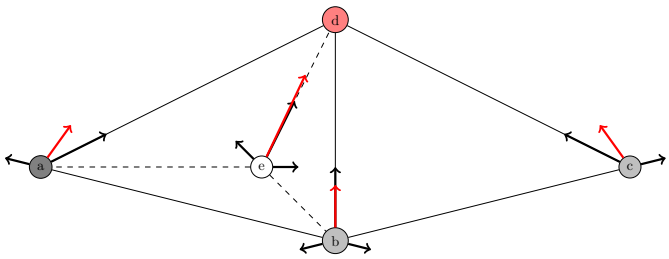
Standby. When the sensors are not active, they will not apply any forces to their neighbors (whatever their behavior) but they only react to forces applied by their neighbors.

Some examples are also given in Fig. 5 for several values of the parameter λ. Algo. 4 presents this part of the method.

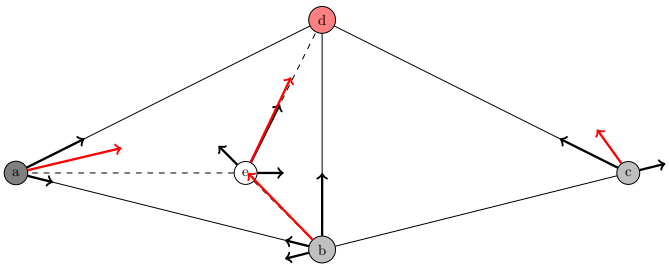
4.3. Boundaries systems

Defining a objective functional is often difficult in the context of mutil-objectives problems, here we are facing a supplementary difficulty of a full decentralized approach. That is the reason why we do not define any objective function. In order to compare the solutions among each other, we introduce boundary systems as references for a qualitative comparison. The boundaries systems maximize only one of the two objectives but must respect all the constraints to be compared with our results.

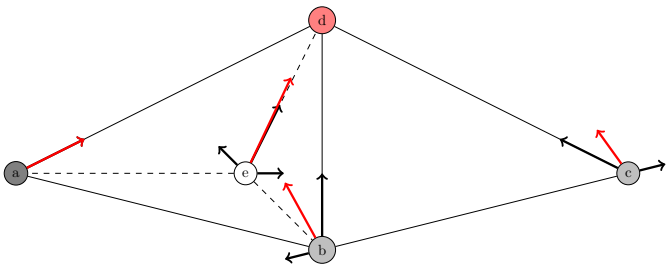
The original addressed problem contains two objectives that have to be maximized simultaneously. Even if there are two objectives, we also consider them independently to enable comparison and evaluate performance of DACYCLEM. The results of these one-objective system give boundaries of the bi-objective problem. In fact, with same initial conditions, each solution of one-objective system must be better for each maximized objective than



(a) Straight lines represents a distance $d_{ij} < r\lambda$



(b) Straight lines represents a distance $d_{ij} > r\lambda$



(c) Straight lines represents a distance $d_{ij} = r\lambda$

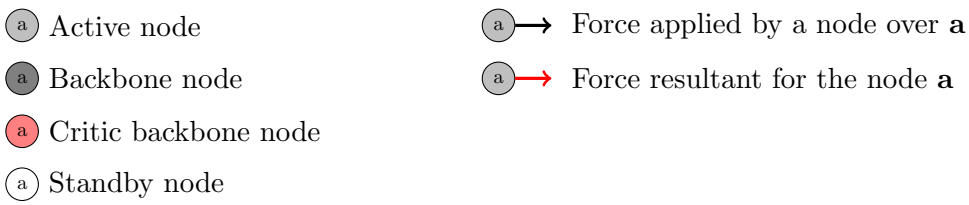


Figure 5: Application examples for the forces.

Algorithm 4 moveWRTForces: Move along forces computed regarding neighborhood states and positions

```
F ← 0                                                    ▷ Initialize force
if I belong to CDS And b < c then                        ▷ Sensor is Critical backbone
    for all n ∈ Neighborhood do
        if n belongs to CDS And n(b) < c then
            F ← F + μ(e)F^A(n)
else if I belong to CDS then                             ▷ Sensor is Backbone
    for all n ∈ Neighborhood do
        if n belongs to CDS And n(b) < c then
            F ← F + μ(e)F^A(n)
        else if n is Active then
            F ← F + μ(e)F^E(n)
else                                                     ▷ Sensor is Active or Standby
    for all n ∈ Neighborhood do
        if n belongs to CDS And n(b) < c then
            F ← F + μ(e)F^A(n)
        else if n belongs to CDS  then
            F ← F + μ(e)F^E(n)
        else if n is Active  then
            F ← F + μ(e)F^R(n)
if F ≠ 0 then
    move toward F
```

the bi-objective problem. Before going any further, we remind that the initial position of each sensor and its battery level are random, both elements impacting largely the initial structure of the network; this point can deeply influence the results. As a matter of fact, to ensure a good evaluation of the performance of the system, we need to carry out several simulations because we cannot achieve optimal solutions for each initial settings.

4.3.1. Time system

To build the *Time system*, we use the meeting point defined previously. For this scenario, after initialization phase, even if the topology is already connected, the sensors keep on moving towards the neighborhood of the meeting point until they are located in a circle of radius $r/2$ around this point. In that context, the resulting network is a full-connected graph. Then, choosing only one active sensor (all the others are standby) ensures the constraints satisfaction. To maximize the lifetime, sensors will never move until the depletion of their battery. Finally, at any time, only one sensor needs to be active. Considering our objective, the best choice is to keep active the sensor with the lower battery level: this choice saves the

battery of the sensors with higher battery level. When a sensor run off power supply — the last active sensor — the sensor with lowest battery level becomes active to replace him. This algorithm permits to save the maximum of battery.

4.3.2. Area system

Maximizing area covering, without taking into account the lifetime is also simple, we call this *Area system*. All sensors are active at the same time and, in order to not waste their energy, we choose to apply them the forces presented previously and consider that each sensor is an active member of a backbone with a equilibrium force. This guarantee to obtain a huge covered area. This choice implies that the system will not survive a long time because of the connectivity constraints.

5. Results

In this part, we give the results obtained by applying our method. The implementation of the method previously detailed was done using the library GraphStream v.1.0 developed at LITIS - RI2C [57, 58]. We first present overall results with screenshots of our application. Then we present a partial parametric study. Finally, we present a comparison with previously mentioned boundaries systems. Unless otherwise specified, in the following section we consider the following values for parameters (Tab. 2). At the initialization, the battery level is set using a random variable *rand* with a Gaussian distribution as defined in the following equation:

$$b = b_{\min} + rand \times (b_{\max} - b_{\min}) \ .$$

Table 2: Main experimental parameters.

Parameter Name	Parameter Value
Simulation area	
Geographical Area	1000×1000
Sensor parameters	
Battery parameters	$b_{\max} = 5000$ and $b_{\min} = 1000$
Speed	$v = 2$
Communication radius	$r = 120$
Force parameters	$\gamma = 1$, $\lambda = 0.8$, $\alpha_1 = 1$, $\alpha_2 = 1$ and $\delta = 1$
Critical threshold	$c = 1500$
Threshold for recording area	$a = 40$
Experiments	
Algorithm	[Area system, DACYCLEM(c), Time system]
Independent runs	100

Asynchrony of the algorithm (Alg. 1) is simulated sequentially by a scheduler that randomly chooses one sensor at each iteration. This scheduler

defines an overall clock for time measurements (see Section 5.2). If during its execution the sensor asks its neighbors to update, this request trigger their execution without the need to be chosen by the scheduler, but it does not change the battery level of these sensors (cf. Section 4.2.2).

When a sensor is chosen, according to its state, its battery level changes. If the sensor is not moving and if its state is standby, its battery looses 2 units and 10 units when the sensor is active. If the sensor is moving, if its state is standby, the battery decreases by 12 units and by 20 units when the sensor is active. During the initialization phase (see Alg. 1), the evaluation metrics (see Section 5.2) are not collected. The measurements and evaluation stop when the connectivity constraint is no longer satisfied.

5.1. Overall results

On Fig. 6, we present a screenshot of our application. The presented network is connected. Two sensors connected by a thick edge are part of the *backbone*. The battery level is shown on each sensor by mean of a colored scale. The covered area is shown in gray. On Fig. 7, we show the evolution of a smaller network at four simulation times [1]. The evaluation is performed as long as the network remains connected. However, a connectivity loss can occur during the simulations, for instance when an articulation point in the graph (member of the backbone) is running out of battery and its neighbor sensors fail to compensate. In that case, each connected component could still proceed with the same method, however this is not taken into account for the evaluation.

5.2. Metrics for the evaluation

As said previously, the two objectives are lifetime and covered area. As observers of the networks, we can know if the connectivity constraints are satisfied. Time evolution is inferred from the sequential scheduler and an iteration is defined by the random choice of a sensor. From an overall view of the system, we define two significant dates:

- T_s as the connected time, the iteration number on which all sensors are connected to the network,
- T_e as the connection lost time, the iteration number on which the network disconnect for the first time.

Therefore, we consider as a lifetime metrics the difference $T_e - T_s$.

For coverage metric, we define the cumulative covered area during the lifetime of the network as follows:

$$\sum_{i=T_s}^{T_e} Area\left(\bigcup_{\text{sensor } j} R_j\right)$$

[1] A video is available (attached to the submission) for 50 sensors.

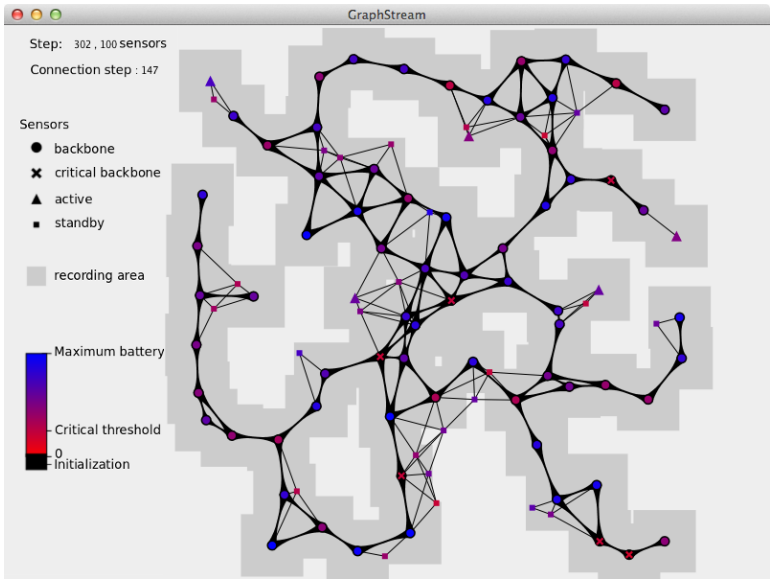


Figure 6: Simulation screenshot for 100 sensors.

The evaluation of our system with the two objectives requires these two metrics: the cumulative covered area and lifetime.

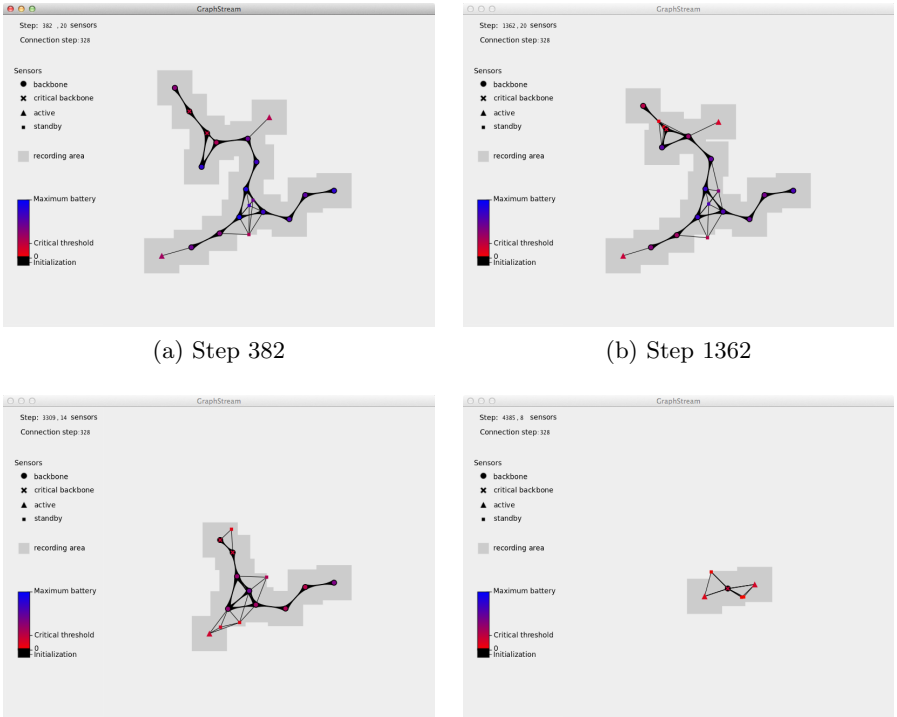
5.3. Boundaries results

We carry out 100 simulation runs for each algorithm: DACYCLEM, Area system and Time system. We first note that the total area covered is strongly dependent on the lifetime (Fig. 8). This can be explained simply because, the longer the lifetime of the network, the larger the covered area is. The lifetime depends on the topology of the initial network especially when the k-vertex-connectivity is small (equals to one when some vertices are articulation points), but for a given system, it seems that the covered area only depends on the lifetime. In fact, we find out a linear dependence between lifetime and cumulative covered surface. From this dependence we can compute the slope of the linear regression and, using an analogy with physics, we can derive a kind of *speed of coverage* of the system.

Using Fig. 8 with a linear regression (Tab. 3 with $y = a \times x$ where x is the lifetime, y is the cumulative covered area and a is the coverage speed), we find these values:

- 1.84 $[m]^2.[s]^{-1}$ for the Time system;
- 34.67 $[m]^2.[s]^{-1}$ for DACYCLEM;
- 59.33 $[m]^2.[s]^{-1}$ for the Area system.

DACYCLEM system is in-between with a coverage speed equal to 34.67 $[m]^2.[s]^{-1}$, half-way from the boundaries. A main result of this study is that



(a) Step 382

(b) Step 1362

(c) Step 3309

(d) Step 4385

Figure 7: Evolution of a 20 sensors network.

we can compute this slope for each system, which mean that for a given set of parameters (for instance changing thresholds), we can obtain its speed of coverage that determine if the system gave priority to lifetime or covered area. Considering these results, DACYCLEM is a trade-off between the two single objective methods Time System and Area System.

5.4. Parametric study

As illustrated in the previous section, our method is able to solve the initial problem. We use the *speed of coverage* as a tool that can define a way to adjust parameters in order to foster one of the objectives. We remind that the *critical threshold* is a value under which the battery level of a backbone

Table 3: Linear regression computation of the *speed of coverage*.

System	Speed of coverage	Asymptotic Standard Error	
Time system	1.83643	± 0.002115	(0.1152%)
DACYCLEM	34.6748	± 0.4222	(1.218 %)
Area system	59.3308	± 0.1764	(0.2973%)

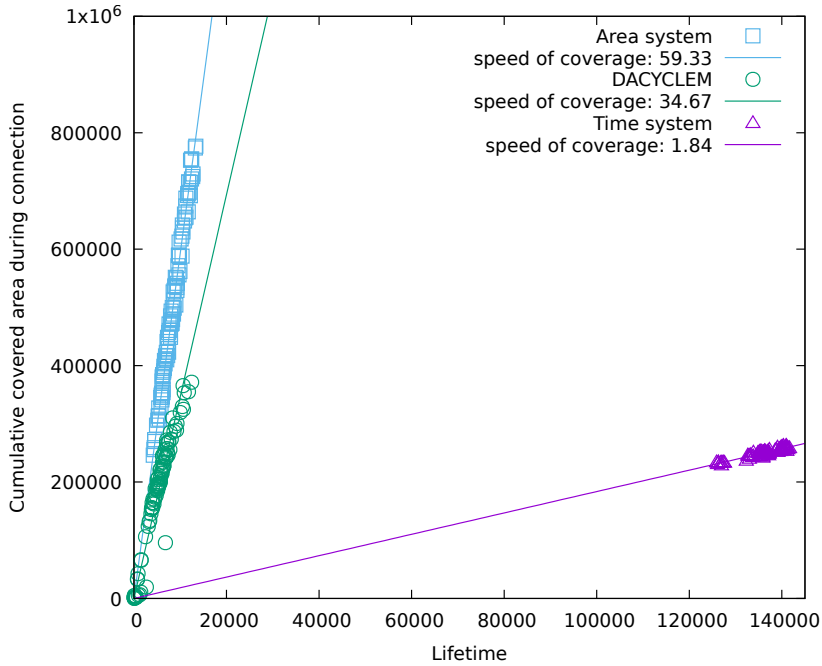


Figure 8: Evaluation of the three systems with the associated speed of coverage computed by a linear regression indicating how the cumulative covered area (y) is proportional to the lifetime (number of iterations or x) with respect to the equation $y = a \times x$.

sensor is considered as critical. If a sensor part of the backbone has a battery level lower than this threshold it triggers a movement of its neighbors in its direction in order to replace him as member of the backbone. We make this value, c, varying in the range 10 to 3000. After doing 100 simulation runs for each value with the same initial topology and battery level, we obtained the results of lifetime versus cumulative coverage. As the linear dependency is convincing (Fig. 8), we assume that a linear regression of these 100 runs are enough to see the influence of the parameter (Tab. 4). It should be noticed however that for some initial configurations, the obtained connected topologies are not robust enough from the k-vertex-connectivity point of view, entailing the production of bad results by DACYCLEM.

Fig. 9 shows that the increase of c is translated by a decrease of the *speed of coverage* until a plateau. As a consequence, increasing the *critical threshold* (c) increases the lifetime time while reducing the cumulative coverage. However, a plateau is reached for $c > 450$. Thus the reduction of the speed of the coverage is possible by increasing the critical threshold up to 450. Also, with $c = 10$, we obtain a value of coverage speed close to the value of the Area system which means that this parameter is good enough to allow DACYCLEM to explore the solution space up to the one of Area System. However, even when changing only this parameter value, DACYCLEM is

Table 4: Linear regression computation of the *speed of coverage* for 100 simulation runs

Critical threshold (c)	Speed of coverage	Asymptotic Standard Error	
10	51.2017	± 0.2036	(0.3977%)
30	49.2692	± 0.2407	(0.4886%)
50	48.179	± 0.2502	(0.5192%)
100	46.4047	± 0.2567	(0.5533%)
150	44.1436	± 0.3094	(0.7008%)
200	43.6983	± 0.3191	(0.7302%)
250	43.1296	± 0.2818	(0.6535%)
300	42.4747	± 0.5776	(1.36%)
350	37.9636	± 0.5331	(1.404%)
400	30.6464	± 0.6861	(2.239%)
450	34.431	± 0.6091	(1.769%)
500	33.583	± 0.605	(1.802%)
600	35.2652	± 0.4234	(1.201%)
700	34.213	± 0.5636	(1.647%)
800	34.8888	± 0.5037	(1.444%)
900	34.4132	± 0.5389	(1.566%)
1000	33.4556	± 0.5714	(1.708%)
1500	34.6748	± 0.4222	(1.218 %)
2000	34.4041	± 0.5265	(1.53%)
3000	34.2018	± 0.5144	(1.504%)

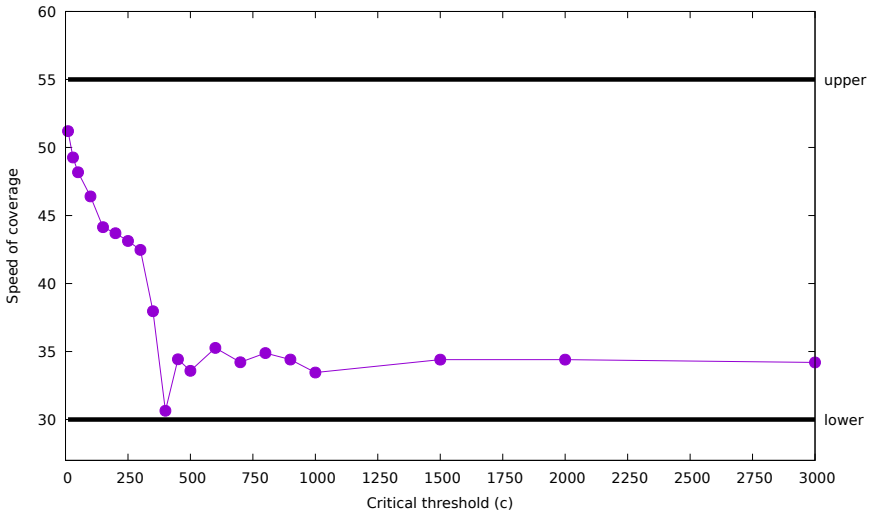


Figure 9: Influence of the *critical threshold* (c) to the *speed of coverage* where c is obtained by linear regression (see Fig. 8 for upper and lower boundaries and Tab. 4 for details).

unable to produce results close to the one of the Time System. We suppose that this plateau comes from DACYCLEM's structure because the sensors with higher battery level are part of the backbone while in the Time system the backbone is made up with sensors with the lower battery level. Thus, even by increasing the critical threshold up to 3000, it cannot permit to reach the speed of coverage of the Time system. Finally, in order to allow DACYCLEM to produce results closer to Time System, it should be possible to tune the λ parameter for strengthening the links between the sensors. The closer to 0 is λ, the stronger the attractive force.

6. Conclusion and future works

In this paper we address the maximization problem of both coverage and lifetime in a mobile wireless sensor network. In particular, the sensors have the ability to be either active (sensing the environment) or idle (saving their lifetime). As a matter of fact, the connectivity constraint apply not only to active sensors but also to the whole set of sensors.

For that purpose, we have proposed DACYCLEM, a new decentralized algorithm based on several known methods to solve the problem. Our method relies on a decentralized construction of a backbone to satisfy the connectivity constraint (lifetime) and a force-based model to determine the sensors' positions (coverage). To balance both objectives we use hierarchical strategy to assign roles to sensors depending on their battery level. The forces moving the sensors in our force-based model depends on the hierarchical level of the sensor and also its belonging to the backbone. In order to analyze the performances of DACYCLEM, we have proposed two simple methods , called boundary systems, that maximize only one of the two objectives under the same constraints.

An implementation of DACYCLEM has been done and its boundary systems as well. When the method is ran with the same parameters on different instances, we observe a linear dependence of the cumulative total covered area with the lifetime. The linear regression of this dependence gives us a value: the *speed of coverage*. This speed of coverage can be used as a metric for qualifying the performances of DACYCLEM for our two initial objectives: coverage and lifetime. We show that the modification of one of the parameters of our method, the critical threshold, entails a change on the speed of coverage and thus the predominance of one objective to the other. The *speed of coverage* could be considered as a new metric for comparing different algorithms for solving the same problem. In future work we plan to improve our method to prevent failures, in the initialization process, due to specific random initial conditions that cause a disconnection after a few iterations.

As mentioned before, we observe that the critical threshold has an high influence on the speed of coverage but even though produces speed of cov-

erage quite far from the one of the boundary system (Area System). Thus, we would like to investigate the influence of other parameters to obtain a coverage speed closer to the Area system (eg. the λ parameter that influences the force behavior). According to that, we finally plan to identify a restricted set of parameters for building Pareto fronts for this two-objective problem and to compare the produced solutions with the results obtained with classical multi-objective optimization methods.

The present work has mainly addressed the combined issues of coverage and connectivity with the implicit aim of maximizing the lifetime of the network. We have thus put aside most aspects related to communication, thus, we did not perform any tests and comparisons using simulation tools. These aspects will be the subject of future works. However, our current decentralized method guarantees that connectivity is maintained during the lifetime of the network, thus communication between sensors is possible at any moment. Regarding the sink, we may apply the decentralized method proposed in [59] for building and maintaining a spanning tree within a dynamic graph. The method is based on random moves of tokens. In the current context, the principle would be the following: at the beginning every sensor is a sink and possesses a token. As soon as two sensors are in the communication range with each other, the tokens are merged. The same merging principle applies when new groups of nodes are in communication range with other groups. Based on random moves of tokens within this virtual spanning forest, the number of simultaneous sinks decreases very quickly and guarantees the existence, at any moment, of a single path from any sensor to a sink.

Acknowledgments

The authors also would like to thanks the reviewers for their useful comments and remarks that improve this article.

References

[1] S. Basagni, A. Carosi, C. Petrioli, C. A. Phillips, Coordinated and controlled mobility of multiple sinks for maximizing the lifetime of wireless sensor networks, Wireless Networks 17 (3) (2011) 759–778. doi:10.1007/s11276-010-0313-8.

[2] N. Michael, M. M. Zavlanos, V. Kumar, G. J. Pappas, Maintaining connectivity in mobile robot networks, in: Experimental Robotics, Springer Berlin Heidelberg, 2009, pp. 117–126. doi:10.1007/978-3-642-00196-3_14.

[3] H. Long, Y. Liu, Y. Wang, R. Dick, H. Yang, Battery allocation for wireless sensor network lifetime maximization under cost constraints,

in: IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, 2009, pp. 705–712.

[4] J. Gobel, A. E. Krzesinski, A model of autonomous motion in ad hoc networks to maximise area coverage, in: 2008 Australasian Telecommunication Networks and Applications Conference, IEEE, 2008. doi:10.1109/atnac.2008.4783333.

[5] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer networks 38 (4) (2002) 393–422. doi:10.1016/S1389-1286(01)00302-4.

[6] A. Boukerche, P. Sun, Connectivity and coverage based protocols for wireless sensor networks, Ad Hoc Networks 80 (2018) 54–69. doi:10.1016/j.adhoc.2018.07.003.

[7] I. Dietrich, F. Dressler, On the lifetime of wireless sensor networks, ACM Trans. Sen. Netw. 5 (1) (2009) 5:1–5:39. doi:10.1145/1464420.1464425.
URL http://doi.acm.org/10.1145/1464420.1464425

[8] J. Li, A. H. Sayed, Modeling bee swarming behavior through diffusion adaptation with asymmetric information sharing, EURASIP Journal on Advances in Signal Processing 2012 (1) (2012) 18. doi:10.1186/1687-6180-2012-18.

[9] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications 1 (4) (2002) 660–670. doi:10.1109/twc.2002.804190.

[10] C. Zhu, C. Zheng, L. Shu, G. Han, A survey on coverage and connectivity issues in wireless sensor networks, Journal of Network and Computer Applications 35 (2012) 619–632. doi:10.1016/j.jnca.2011.11.016.

[11] A. Ghosh, S. K. Das, Coverage and connectivity issues in wireless sensor networks: A survey, Pervasive and Mobile Computing 4 (2008) 303–334. doi:10.1016/j.pmcj.2008.02.001.

[12] Y.-G. Yue, P. He, A comprehensive survey on the reliability of mobile wireless sensor networks: Taxonomy, challenges, and future directions, Information Fusion 44 (2018) 188–204. doi:10.1016/j.inffus.2018.03.005.

[13] M. Li, Z. Li, A. V. Vasilakos, A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues, Proceedings of the IEEE 101 (12) (2013) 2538–2557. doi:10.1109/jproc.2013.2257631.

[14] M. Bouaziz, A. Rachedi, A survey on mobility management protocols in wireless sensor networks based on 6lowpan technology, Computer Communications 74 (2016) 3–15. doi:10.1016/j.comcom.2014.10.004.

[15] A. Achour, L. Deru, J. C. Deprez, Mobility management for wireless sensor networks a state-of-the-art, Procedia Computer Science 52 (2015) 1101–1107. doi:10.1016/j.procs.2015.05.126.

[16] D.-R. Chen, An energy-efficient QoS routing for wireless sensor networks using self-stabilizing algorithm, Ad Hoc Networks 37 (2016) 240–255. doi:10.1016/j.adhoc.2015.08.022.

[17] A. Sangwan, R. P. Singh, Survey on coverage problems in wireless sensor networks, Wireless Personal Communications 80 (4) (2014) 1475–1500. doi:10.1007/s11277-014-2094-3.

[18] W. Fang, X. Song, X. Wu, J. Sun, M. Hu, Novel efficient deployment schemes for sensor coverage in mobile wireless sensor networks, Information Fusion 41 (2018) 25–36. doi:10.1016/j.inffus.2017.08.001.

[19] B. Liu, O. Dousse, P. Nain, D. Towsley, Dynamic coverage of mobile sensor networks, IEEE Transactions on Parallel and Distributed Systems 24 (2) (2013) 301–311. doi:10.1109/tpds.2012.141.

[20] O. Banimelhem, M. Naserllah, A. Abu-Hantash, An efficient coverage in wireless sensor networks using fuzzy logic-based control for the mobile node movement, in: 2017 Advances in Wireless and Optical Communications (RTUWO), IEEE, 2017. doi:10.1109/rtuwo.2017.8228541.

[21] M. Rebai, M. L. berre, H. Snoussi, F. Hnaien, L. Khoukhi, Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks, Computers & Operations Research 59 (2015) 11–21. doi:10.1016/j.cor.2014.11.002.

[22] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, S. Sasaki, Rearrangement of mobile wireless sensor nodes for coverage maximization based on immune node deployment algorithm, Computers & Electrical Engineering 43 (2015) 76–89. doi:10.1016/j.compeleceng.2015.04.003.

[23] D. Reina, H. Tawfik, S. Toral, Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks, Ad Hoc Networks 68 (2018) 16–32. doi:10.1016/j.adhoc.2017.09.005.

[24] J. Roselin, P. Latha, S. Benitta, Maximizing the wireless sensor networks lifetime through energy efficient connected coverage, Ad Hoc Networks 62 (2017) 1–10. doi:10.1016/j.adhoc.2017.04.001.

[25] S. M. Mohamed, H. S. Hamza, I. A. Saroit, Coverage in mobile wireless sensor networks (M-WSN): A survey, Computer Communications 110 (Supplement C) (2017) 133 – 150. doi:10.1016/j.comcom.2017.06.010.

[26] M. Penrose, Random geometric graphs, no. 5, Oxford University Press, 2003.

[27] E. D. Demaine, M. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar, S. Oveisgharan, M. Zadimoghaddam, Minimizing movement, ACM Transactions on Algorithms 5 (3) (2009) 1–30. doi:10.1145/1541885.1541891.

[28] M. Zavlanos, G. Pappas, Controlling connectivity of dynamic graphs, in: Proc. of 44th IEEE Conference on Decision and Control (CDC-ECC'05), IEEE, 2005, pp. 6388–6393. doi:10.1109/cdc.2005.1583186.

[29] Y. Kim, M. Mesbahi, On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian, IEEE Transactions on Automatic Control 51 (1) (2006) 116–120. doi:10.1109/tac.2005.861710.

[30] M. Zavlanos, G. Pappas, Distributed connectivity control of mobile networks, IEEE Transactions on Robotics 24 (6) (2008) 1416–1428. doi:10.1109/cdc.2007.4434525.

[31] S. Silvestri, K. Goss, Mobibar: An autonomous deployment algorithm for barrier coverage with mobile sensors, Ad Hoc Networks 54 (2017) 111–129.

[32] P. Basu, J. Redi, Movement control algorithms for realization of fault-tolerant ad hoc robot networks, IEEE Network 18 (4) (2004) 36–44. doi:10.1109/mnet.2004.1316760.

[33] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, N. R. Franks, Collective memory and spatial sorting in animal groups, Journal of Theoretical Biology 218 (1) (2002) 1–11. doi:10.1006/jtbi.2002.3065.

[34] F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (2001) 393–422.

[35] M. Cardei, M. Thai, Y. Li, W. Wu, Energy-efficient target coverage in wireless sensor networks, in: Proc. IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies. (INFOCOM), Vol. 3, IEEE, 2005, pp. 1976–1984. doi:10.1109/infcom.2005.1498475.

[36] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Wireless Networks (5) (2002) 481–494. doi:10.1023/a:1016542229220.

[37] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration for energy conservation in sensor networks, ACM Transactions on Sensor Networks (TOSN) 1 (1) (2005) 36–72. doi:10.1145/1077391.1077394.

[38] D. Simplot-Ryl, I. Stojmenović, J. Wu, Energy-Efficient Backbone Construction, Broadcasting, and Area Coverage in Sensor Networks, Handbook of Sensor Networks: Algorithms and Architectures (2005) 343–380doi:10.1002/047174414x.ch11.

[39] M. Min, F. Wang, D. Du, P. Pardalos, A reliable virtual backbone scheme in mobile ad-hoc networks, in: Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2004, pp. 60–69. doi:10.1109/mahss.2004.1392074.

[40] J. Reif, H. Wang, Social potential fields: A distributed behavioral control for autonomous robots* 1, Robotics and Autonomous Systems 27 (3) (1999) 171–194. doi:10.1016/s0921-8890(99)00004-4.

[41] L. Bao, J. J. Garcia-Luna-Aceves, Topology management in ad hoc networks, in: Proc. of the 4th ACM international symposium on Mobile ad hoc networking & computing - MobiHoc'03, ACM Press, 2003. doi:10.1145/778415.778432.

[42] Y. Wang, W. Wang, X. Li, Distributed low-cost backbone formation for wireless ad hoc networks, in: Proc. of the 6th ACM international symposium on Mobile ad hoc networking and computing - MobiHoc'05, 2005, pp. 2–13. doi:10.1145/1062689.1062692.

[43] H. Liu, R. Gupta, Selective backbone construction for topology control in ad hoc networks, in: Proc. of International Conference on Mobile Ad-hoc and Sensor Systems, IEEE, 2004, pp. 41–50. doi:10.1109/mahss.2004.1392069.

[44] A. Srinivas, G. Zussman, E. Modiano, Mobile backbone networks: construction and maintenance, in: Proc. of ACM international symposium on Mobile ad hoc networking and computing, 2006, pp. 166–177. doi:10.1145/1132905.1132924.

[45] T. Shi, S. Cheng, Z. Cai, Y. Li, J. Li, Exploring connected dominating sets in energy harvest networks, IEEE/ACM Transactions on Networking 25 (3) (2017) 1803–1817. doi:10.1109/tnet.2017.2657688.

[46] D. Gage, Command control for many-robot systems, Unmanned Systems 10 (4) (1992) 28–34.

[47] A. Howard, M. Mataric, G. Sukhatme, Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem, Vol. 5, Springer Japan, 2002, pp. 299–308. doi:10.1007/978-4-431-65941-9_30.

[48] S. Poduri, G. Sukhatme, Constrained coverage for mobile sensor networks, in: Proc. of IEEE International Conference on Robotics and Automation (ICRA'04), Vol. 1, 2004, pp. 165–171. doi:10.1109/ROBOT.2004.1307146.

[49] G. Peng, Z. Guangxi, F. Liang, An Adaptive Coverage Algorithm for Large Scale Mobile Sensor Networks, Springer Berlin Heidelberg, 2006, pp. 468–477. doi:10.1007/11833529_48.

[50] N. Heo, P. Varshney, Energy-efficient deployment of intelligent mobile sensor networks, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 35 (1) (2005) 78–92. doi:10.1109/tsmca.2004.838486.

[51] G. Wang, G. Cao, T. La Porta, Movement-assisted sensor deployment, in: Proc. of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 4, IEEE, 2004, pp. 2469–2479. doi:10.1109/TMC.2006.80.

[52] G. Wang, G. Cao, T. La Porta, W. Zhang, Sensor relocation in mobile sensor networks, in: Proc. of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 4, IEEE, 2005, pp. 2302–2312. doi:10.1109/TMC.2006.80.

[53] W. Fang, X. Song, X. Wu, J. Sun, M. Hu, Novel efficient deployment schemes for sensor coverage in mobile wireless sensor networks, Information Fusion 41 (2018) 25–36. doi:10.1016/j.inffus.2017.08.001.

[54] N. Correll, J. Bachrach, D. Vickery, D. Rus, Ad-hoc wireless network coverage with networked robots that cannot localize, in: 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009. doi:10.1109/robot.2009.5152742.

[55] M. A. Batalin, G. S. Sukhatme, Spreading out: A local approach to multi-robot coverage, in: Distributed Autonomous Robotic Systems 5, Springer Japan, 2002, pp. 373–382. doi:10.1007/978-4-431-65941-9_37.

[56] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, J.-P. Sheu, Wireless Networks 8 (2/3) (2002) 153–167. doi:10.1023/a:1013763825347.

[57] A. Dutot, F. Guinand, D. Olivier, Y. Pigné, Graphstream: A tool for bridging the gap between complex systems and dynamic graphs, in: Emergent Properties in Natural and Artificial Complex Systems. Satellite Conference within the 4th European Conference on Complex Systems (ECCS'2007), 2007.

[58] LITIS – RI2C, Documentation GraphStream, `http://graphstream-project.org/doc/`, [Online; accessed 21-Feb-2018] (2017).

[59] A. Casteigts, S. Chaumette, F. Guinand, Y. Pigné, Distributed maintenance of anytime available spanning trees in dynamic networks, in: Adhoc, Mobile, and Wireless Network, Springer Berlin Heidelberg, 2013, pp. 99–110. `doi:10.1007/978-3-642-39247-4_9`.