

# Greedy Algorithms for Target Coverage Lifetime Management Problem in Wireless Sensor Networks

Babacar Diop, Dame Diongue and Ousmane Thiaré

*Department of Computer Science  
Faculty of Applied Sciences and Technology  
University Gaston Berger, BP. 234 Saint-Louis, Senegal  
babacardiop96@yahoo.fr, {ddiongue.ep2112812, ousmane.thiare}@ugb.edu.sn*

## Abstract

*When several low power sensors are randomly deployed in a field for monitoring targets located at fixed positions, managing the network lifetime is useful as long as replacing battery of dead sensors is not often feasible. The most commonly investigated mechanism for coverage preserving while maximizing the network lifetime is to design efficient sleep scheduling protocols, so that sensors can alternate their state between being active or not. Maximizing lifetime of a sensor network while satisfying a predefined coverage requirement is an optimization problem, which most of times cannot be optimally solved in polynomial time. In this paper, we address this problem by using set cover approach. We propose a greedy algorithm that distributes sensors among disjoint and non-disjoint set covers with the requirement that each set cover satisfies full targets coverage. The algorithm is an improvement of the classical greedy set cover algorithm, and its approximation ratio is verified to be not worse than  $\log(m)$ . Simulation results show good performance over some other solutions found in the literature. We provide also a comparison of several greedy techniques found in the literature addressed in the context of different design choices linked to the target coverage problem.*

**Keywords:** Target coverage, greedy algorithms, wireless sensor networks

## 1. Introduction

Wireless sensor networks (WSN) are systems of small, low-power networked sensing devices, often deployed over an area of interest (AoI) for monitoring events and/or performing application specific tasks in response to the detected events. Coverage problem is one of the fundamental issues in wireless sensor networks. It essentially refers to how well the sensors observe the physical area. Coverage problems in the literature of WSN can be categorized into three classes [18]: area coverage (the objective is to monitor each location within an AoI with at least one sensor), target coverage (which aims to monitor or track a discrete set of targets points in a sensing field) and barrier coverage (which focus on minimizing the probability of undetected intrusion in a barrier constructed by a sensor network). An important part of the study of coverage in WSN is devoted to target coverage. Target coverage application echoes in various fields (military, medical, industrial) where the studied subjects (military target, medical patient) can be observed or tracked by nearby sensors, embedded or attached on it.

In this paper we address the target coverage problem in the context of target coverage application. We assume that several sensors equipped with limited energy

supplies are randomly deployed in an AoI for monitoring a finite set of targets with known locations. Sensors are redundantly scattered around targets for compensating the lack of precise positioning. Hence, it is likely that some sensors will cover more targets than others. The network coverage requirement consists of monitoring all the targets during the lifetime of the network. The network lifetime is defined as the time during which all targets in the sensing field are monitored until there exist at least one breached target in the network. If subsets of sensors satisfying the coverage requirement can be extracted from the initial set of sensors, it is not worth to make all of them operate at the same time. Furthermore, if sensors can be partitioned into different sets, each of them satisfying the coverage requirement, and activating these sets consecutively, a long-lived network can be constructed. The problem of finding the optimal number of sets that satisfy the coverage requirement is NP-complete [2].

The contributions of this paper are described as follow. 1) We first propose a greedy set cover algorithm for non-disjoint set covers generation, based on a modified version of the classical greedy set cover [3, 4], and prove the proposition that our algorithm has approximation ratio not worse than  $\log(m)$ , where  $m$  represents the number of targets. 2) We simulate our proposition along with similar works in the literature in order to compare the performance of the proposed algorithm in terms of computed set covers and lifetime maximization. 3) Finally, we provide a brief synthesis of greedy paradigm-based solutions proposed in the literature for solving target coverage problems.

The rest of the paper is organized as follow: in section 2, we address the target coverage problem as the maximum set covers problem and introduce the minimum set cover problem for set cover modeling. In section 3, we introduce the partitioning mechanism by discussing the use of a flexible profit function that helps to perform accurate selection. The proposed algorithm (Greedy-MSSC) is presented in section 4, followed by an analysis of its approximation ratio and complexity. Performance evaluation of our algorithm is presented in section 5. In section 6, we provide a synthesis of related works in the literature based on greedy techniques. Section 7 concludes this paper.

## 2. Target Coverage Problem

### 2.1. Assumptions

We consider a scenario where  $n$  battery-powered sensors  $s_1 \dots s_n$  are randomly scattered in proximity of  $m$  targets  $z_1 \dots z_m$  located at fixed positions within a sensing field. The number of deployed sensors is greater than the required optimum to initially satisfy the coverage requirement. Each sensor  $s_k$  has a predefined sensing range  $r_k$ , an initial energy supply  $e_k$  and is supposed to be equipped with required module for locating its geographical positions. We consider that the coverage measure between one sensor  $s_k$  and target  $z_j$  equals to 1 if  $s_k$  covers  $z_j$ , and 0 otherwise. A sensor  $s_k$  covers a target  $z_j$  if the Euclidean distance between  $s_k$  and  $z_j$  is less than the sensing range  $r_k$ . We also assume the existence of a base station controller that dispose sensors and targets positions and performs the sleep scheduling process. Each sensor is also equipped with energy control module so that it can alternate between active and sleep states. When being active, the sensor consumes energy for sensing and processing physical data gathered from monitored targets, and communicating with the sink. The sleep state allows sensors to conserve energy since being not able to sense or communicate data in the sensing field. For sensors activities synchronization, the base station may periodically send short beacons to sensors so that they can alternate their

state. We do not deal with connectivity problem; therefore it is considered that sensors are able to transmit gathered data to the sink using one hop communication. For data gathering, protocols like LEACH [12] can be used.

## 2.2. Problem Definition

**Definition 1.** (Target coverage problem) Based on the assumptions stated above, the target coverage problem consists on scheduling sensors activities such that all the targets are continuously observed and the lifetime is maximized.

Since the deployment ensures that every target is covered by at least one sensor and that the targets are redundantly covered, sensors scheduling can be performed by partitioning sensors into different groups, each one satisfying the coverage requirement, and then activate them in a round robin manner. Only the activated group contains active sensors in the field, while sensors in inactive groups are in sleep state. In what follow, we propose to solve the target coverage problem by addressing it with the maximum set cover problem.

## 2.3. Maximum Set Cover Problem

**Definition 2.** (Maximum set cover problem) The main goal of the maximum set cover problem is to partition sensors into a maximum number of set covers, namely  $C_1 \dots C_k$  with lifetimes  $t_1 \dots t_k$  such that each set can monitor all the targets and the sum  $t_1 + \dots + t_k$  is maximized. A set cover can be viewed as a subset of sensors that satisfy the coverage requirement, in our case covering all targets in the area of interest.

An intuitive proposition is to maximize the number of computable set covers by finding a way to minimize the number of sensors included in each set. This problem was initially addressed in [10] as the minimum set cover problem, which is one of the oldest and most studied NP-hard problem.

**Definition 3.** (Minimum set cover problem) Given a universe  $U$  of  $m$  elements (representing the set of targets), and a collection of  $n$  subsets of elements in  $U$  (denoting the subset of targets covered by every sensor), the goal is to cover  $U$  with the smallest possible number of subsets, *i.e.*, select the minimum number of sensors to cover  $U$ .

The greedy set cover is one of the best polynomial time algorithms for approximating the minimum set cover problem, with performance ratio equal to  $\log(m)$ . If each sensor is assigned a weight, which is a non-negative value, the score of a solution represents the total weight of used sensors for monitoring all the targets. Yet the goal of the problem is to minimize the score.

## 2.4. Non-disjoint Sets Constraint

We add another constraint to the maximum set covers problem. This states that computed set covers need not to be disjoint, as we refer to in [14]. Building non-disjoint set covers allow sensors to be member of multiple sets with the condition that a sensor  $s_k$  cannot be active for more than its initial energy supply.

# 3. Sensors Selection Mechanism

## 3.1. Theoretical Bound

The randomness associated with the redundancy effect may bring disproportion in the number of sensors covering each target. We denote by  $S_j$  the subset of sensors

covering target  $z_j$ , and  $Z_k$  the number of targets covered by sensor  $s_k$ . After the deployment, some targets may be weaker covered than others. We denote by  $z_{min}$  the target covered by the minimum number of sensors after the deployment.  $z_{min}$  is called the critical target (*ct.*). Sensors that cover *ct.* are called critical sensors (*cs.*). The number of sensors that cover the critical target places an upper bound on the number of computable set covers, namely the theoretical maximum. The relation between the number of computable sets and the theoretical maximum is described in equation (1).

$$|C(I)| = d \min_{z_j, j=1 \dots m} (|S_j|) \quad (1)$$

Where  $C(I)$  denotes the collection of set covers computed by an approximation algorithm for an instance  $I$  of the problem, and  $d$  the number of set covers a sensor can be member of. To compute the closest number of set covers to the theoretical maximum, the number of sensors in a set cover must be as minimal as possible. This establishes clearly that the maximization of set covers is driven by the minimization of sensors in each set cover. Hence, an efficient sensor selection strategy must be adopted. This latter must avoid including more than one sensor covering the critical targets in one set cover. Accurate selection can be achieved by designing a profit function that represents the contribution of each sensor relatively to the set of targets.

### 3.2. Sensors Cost Evaluation

Each sensor has a contribution, which can be considered as its cost effectiveness value in relation with other sensors and targets in the network. Sensors sensing areas overlap and hence targets may be densely covered. In our algorithm design, we give priority to sensors that both cover the most poorly covered targets and the maximum number of uncovered targets. For example when building set cover, there may be some target covered by just one sensor. These latter will be member of this set cover, regardless how many targets they cover. The sensor selection process in this first case is trivial. In the second case, which means that targets are simultaneously covered by multiple sensors, the most cost effective sensor is selected. The cost effectiveness of one sensor  $s_k$  is measured relatively to the number of covered targets, and the cost effectiveness of other sensors which sensing areas overlap with  $s_k$ 's sensing area.

We first introduce the coverage relationship function (CRF), which describes the coverage status of one sensor. The CRF between  $s_k$  and  $z_j$  equals to the ratio between the detection value of  $s_k$  and  $z_j$ , and the sum of detection values of all sensors that cover  $z_j$ . If these latter can monitor  $z_j$  with great coverage values, this implies  $z_j$  is densely covered, resulting in a lower CRF for  $s_k$  contrasting with the case where  $z_j$  is weakly covered. We define the CRF in equation (2).

$$w_j^k = \frac{p_j^k}{\sum_{s_i \in S_0} p_j^i} \quad (2)$$

Where  $p_j^k$  is a binary function which equals to 0 if target  $z_j$  is covered by  $s_k$ , and 1 if else,  $S_0$  represents the initial set of sensors. A binary representation of the detection value is an idealistic sensor coverage model, which can only be applied to some kind of sensors (temperature sensor), and not to others types (acoustic or infrared sensors). Considering  $Z_0$  as the initial set of targets, the cost effectiveness of  $s_k$  with respect to  $Z_0$  equals to the normalized sum of CRF between  $s_k$  and all targets that belong to  $Z_0$ .

$$w_{z_0}^k = \sum_{z_j \in Z_0} \frac{p_j^k}{m e^k \sum_{s_i \in S_0} \frac{p_i^j}{e^i}} \quad (3)$$

When building set covers, sensors can be iteratively selected, applying equation (3) in each stage for selecting the maximum profit sensor. The profit function that we apply here brings more flexibility for sensors selection during set cover building, and is not computation expensive. Considering that the algorithm runs through iteration, there is just one parameter of the cost effectiveness function which is susceptible to change: the remaining energy of the sensors whose sensing ranges overlap with the evaluated sensor's sensing range. Furthermore, it implicitly takes into account the criticality of sensors that cover the critical target. We do not need to identify the critical target first, before selecting among the critical sensors, which cover it, the one covering the maximum number of remaining uncovered targets during this stage. The ideal case is presented when one sensor covers a huge number of uncovered targets, and being the only sensor covering these targets. In this case, the value of the cost effectiveness function equals to 1. This means that all the targets covered by this sensor are critical. Even if there exist one critical target in the network, the corresponding sensor will be selected, whatever the number of uncovered targets this sensor covers. The roughest case is presented when several unselected sensors cover the same number of uncovered targets and that there is a high redundancy in targets coverage. In this case, almost all evaluated sensors have the same values and the sensor having the smaller subscript is selected. The selection in this case may be mitigated.

## 4. Using Greedy Algorithms

In this section, we first design the greedy minimum set cover algorithm, which tries to select the minimum number of sensors to form a set cover. Secondly we present the greedy maximum set cover algorithm, which the role is to iterate the set cover building process with available sensors.

### 4.1. Greedy Minimum Set Cover

We begin to introduce a kind of greedy minimum set cover algorithm that is essentially based on the application of equation (3) to a form set cover. The pseudo-code of the algorithm is shown in Algorithm 1. The algorithm is composed of one loop during which the algorithm holds the set of uncovered targets, initialized to the initial set of targets, and tries to select the sensor having the greatest cost effectiveness value according to equation (3). Once a sensor has been selected, the covered targets are removed from the list (*Uncvd*) and the sensor is added to the current set cover. The process is iterated until there remains no more uncovered target.

---

#### Algorithm 1: Greedy-Minimum Sensors Set Cover (Greedy-MSSC)

---

$S_0 = (s_1 \dots s_n), Z_0 = (z_1 \dots z_m), Z_1 \dots Z_n$   
**01:** *Uncvd* =  $Z_0$ ;  
**02:**  $C = \emptyset$ ;  
**03:** **while** *Uncvd*  $\neq \emptyset$  **do**  
**04:**     **foreach**  $s_k \in S_0$  **do**  
**05:**         Compute the cost effectiveness function  $w_{z_0}^k$ ;  
**06:**     **end**

---

---

**07:** Pick the sensor that has the maximum cost ( $S_{\max}$ );  
**08:**  $C = C \cup S_{\max}$ ;  
**09:**  $Uncvd = Uncvd - Z_{\max}$ ;  
**10: end**  
**11: return C;**

---

## 4.2. Approximation Ratio of Greedy-MSSC

**Proposition.** Approximation ratio of Greedy-MSSC is not worse than  $\log(m)$ .

**Proof.** Let  $m_t$  be the number of uncovered targets after  $t$  iterations of the algorithm, and  $k$  the cardinality of one set cover computed by the optimal solution  $Opt(I)$ , for an instance  $I$  of the problem. Then  $Opt(I)$  runs through  $k$  iterations to compute the set cover. At  $t_{th}$  iteration, since each remaining target is covered by at least one of the  $k$  sensors, one of the available sensors in  $Opt(I)$  covers at least  $m_t/k$  targets. This implies that:

$$m_{t+1} \leq m_t/k$$

$$m_t \leq m_0(1 - 1/k)^t, \forall t, 1 \leq t \leq k$$

Equation 5 states that the number of remaining targets at  $t_{th}$  iteration still remains inferior to the cardinality of the initial set of targets to which  $m_t/k$  is subtracted. Furthermore, it can be proved (a simple 2D graph plot of these two functions is sufficient) that:

$$(1 - X) \leq \exp(-X)$$

Where  $(1 - X) = \exp(-X)$ ; if  $X = 0$ ; and  $(1 - X) < \exp(-X)$  for all values of  $X \neq 0$ . If we set  $X = 1/k$ , then we can combine (5) and (6) to obtain:

$$(1 - 1/k)^t < (\exp(-1/k))^t$$

$$\Rightarrow m_t \leq m(1 - 1/k)^t < m(\exp(-t/k))$$

When  $t = k \ln(m)$

$$m_{k \ln(m)} < m(\exp(-\ln(m))) = 1$$

Therefore every target is covered after at most  $k \ln(m)$  iterations of our greedy algorithm. The number of sensors included in one set cover computed by our greedy algorithm is  $\ln(m)$  times  $Opt(I)$ .

## 4.3. Greedy Maximum Set Cover

We present the greedy maximum set cover algorithm in Algorithm 2.

---

### Algorithm 2: Greedy-Maximum Sensors Set Cover (Max-SSC)

---

**01: while** there is available sensors to form a set cover  
**02:** Execute Greedy-MSSC;  
**03:** Update lifetimes of selected sensors;  
**04: end**  
**05: return** set covers;

---

## 4.4. Complexity Analysis

Let us consider that we have  $n$  sensors randomly deployed in an AoI to monitor  $m$  targets. When constructing one set cover, sensors are iteratively selected until all targets are covered. When the set cover is completed, all selected sensors battery lifetimes are updated. If the remaining battery lifetime of one sensor allows it to be part of another set cover, this sensor can be selected further. Knowing that every sensor

spends uniformly  $r$  energy unit when its set cover is activated, a sensor can be selected at most  $d$  times by the algorithm, with  $d = e_k/r, \forall s_i \in S_0$ , where  $e_k$  represents the initial battery lifetime of  $s_k$ . The worst-case runtime would have included the complete sensors' set of length  $n$ , each of them being used  $d$  times with  $m$  targets. The runtime of the algorithm is equivalent to the following summation:

$$T(n) = dnm + d(n-1)(m-1) + \dots + d(n(n-1))(m - (n-1) \bmod m)$$

$$T(n) = \sum_{k=0}^{n-1} (n-k)(m - k \bmod m)$$

We can deduce that the worst case complexity of the Max-SSC equals to  $O(dn^2m)$ .

## 5. Simulation Results

To evaluate the performance of our algorithm, we simulate a network with static sensor nodes and target points randomly located in a 100m X 100m area. An implementation prototype of Greedy-MSSC is developed in MATLAB 7.0. For topology generation, targets are spread first in the area by generating random coordinates, followed by sensors such that each target is covered by at least one sensor.

### 5.1. Experiment 1

In the first experiment, we vary the number of sensors from 30 to 100, fixing the number of targets to 10, and the sensing range to 30 m. The initial energy of sensors is set to 1J and the runtime of each cover set equals 0,5 time units. A linear energy consumption model is considered for simplification, *i.e.*, if a sensor is active for  $e$  time units then it consumes  $e$  energy units. The algorithms are executed 10 times with different sensors and targets positioning and the mean values of the number of computed set covers are plotted in Figure 1. We begin to compare our results with the theoretical maximum. In parallel with the evolution of the number of set covers for disjoint sets when the number of sensors increases, we can remark the closeness of the results output by our algorithm with the theoretical maximum. These results highlight the flexibility of our cost effectiveness function during sensors selection. The critical targets and the corresponding critical sensors are well managed.

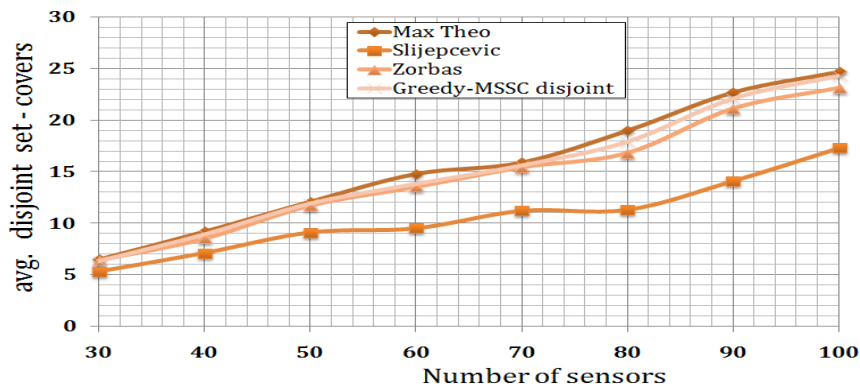


Figure 1. Average of Disjoint Covers vs. Number of Sensors

Beside the closeness of our algorithm with the theoretical maximum, we compare in another hand the average of set covers computed by disjoint *Greedy - MSSC*, *MC - MCh* of Slijepcevic and Potkonjak [1] and *B{GoP}* of Zorbas [3]. Considering the

disjoint set covers, simulation results show through the increase of the number of sensors, a constant evolution and superiority tendency in terms of computed set covers when we compare our algorithm with the other ones. The distance between the results of Slijepcevic and Potkonjak [5] and the other shows that the adopted strategy for managing the critical targets does not always guarantee the lack of double or multiple targets covering. So each time a new set cover is built, there is at least one target covered more than one time. It becomes more complicated when this target is critical. Clearly if a critical target is covered by  $k$  sensors ( $k \geq 0$ ) in one set cover, then  $k$  set covers are sacrificed and the theoretical maximum will not be reached.

### 5.2. Experiment 2

We used the same parameters than in experiment 1, and plot the results in Figure 2 after 10 times running of the evaluated algorithms with different sensors and targets positioning.

We can remark that the number of non-disjoint set covers computed by the algorithms given a fixed number of sensors, is twice greater than the number of disjoint set covers. For example with 30 and 100 deployed sensors, the corresponding cardinalities are respectively 6 and 25 considering disjoint set covers, and 16 and 61 considering non-disjoint set covers. Knowing that a sensor in a non-disjoint set cover  $C_i$  is susceptible to be activated  $d$  times, this sensor will be active for  $e_k^i = e_k/d$  time units, while a sensor in a disjoint one will be activated one time to operate for  $e_k^i = e_k$  time units. With such an arrangement, we expect to get a longer lifetime in the case of non-disjoint set covers.

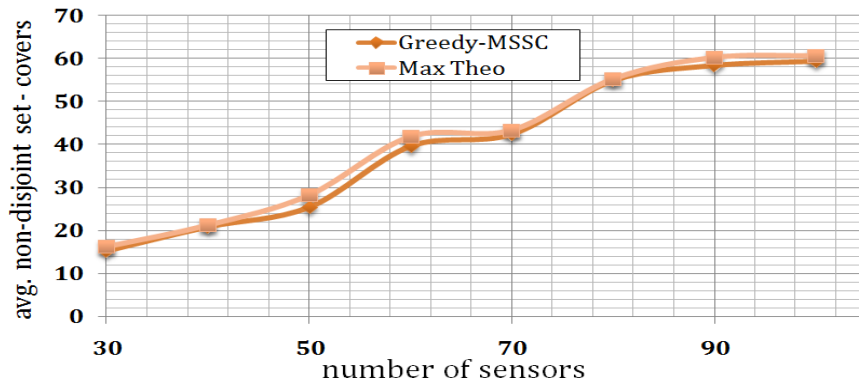


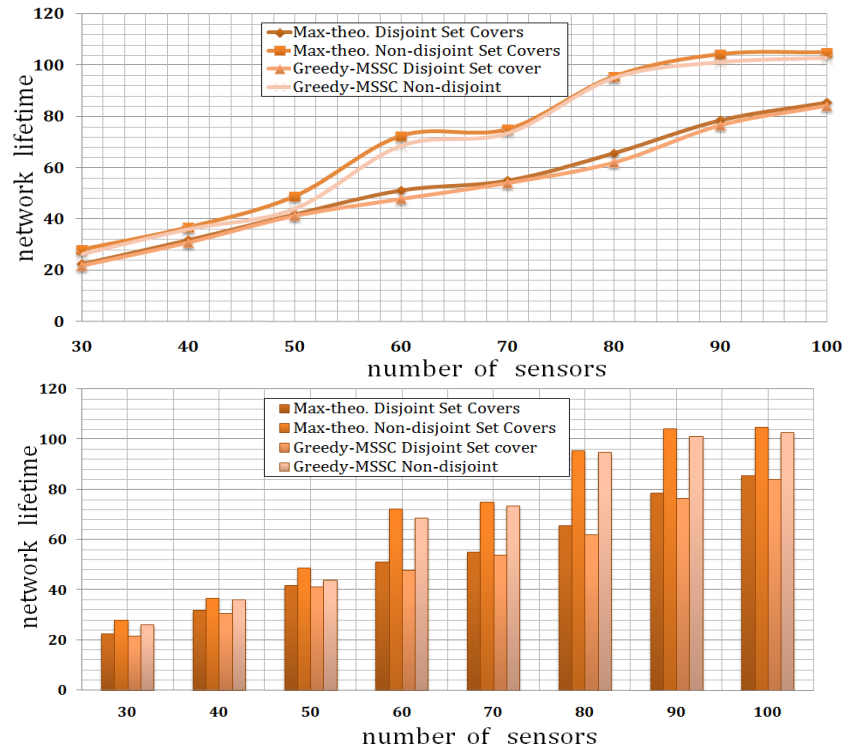
Figure 2. Average of Non-disjoints Set Covers vs. Number of Sensors

### 5.3. Experiment 3

In experiment 3, we study the evolution of the network lifetime while increasing number of sensors. The base station executes the algorithm and sends a start message to sensors that are member of one set cover. In the case of disjoint set covers; these sensors are active until they run out of power. Since the initial energy of sensors is the same, all disjoint set covers share the same operational time value. This latter is then linearly proportional to the initial energy of sensors. The lifetime of the network in using disjoint set covers is equal to lifetimes of one set cover time the number of disjoint sets. Figure 3 shows that the lifetime of the network is highly improved when non-disjoint set covers are produced. This remark becomes more interesting when the number of sensors grows higher. For example, with 30 and 100sensors, the lifetime is



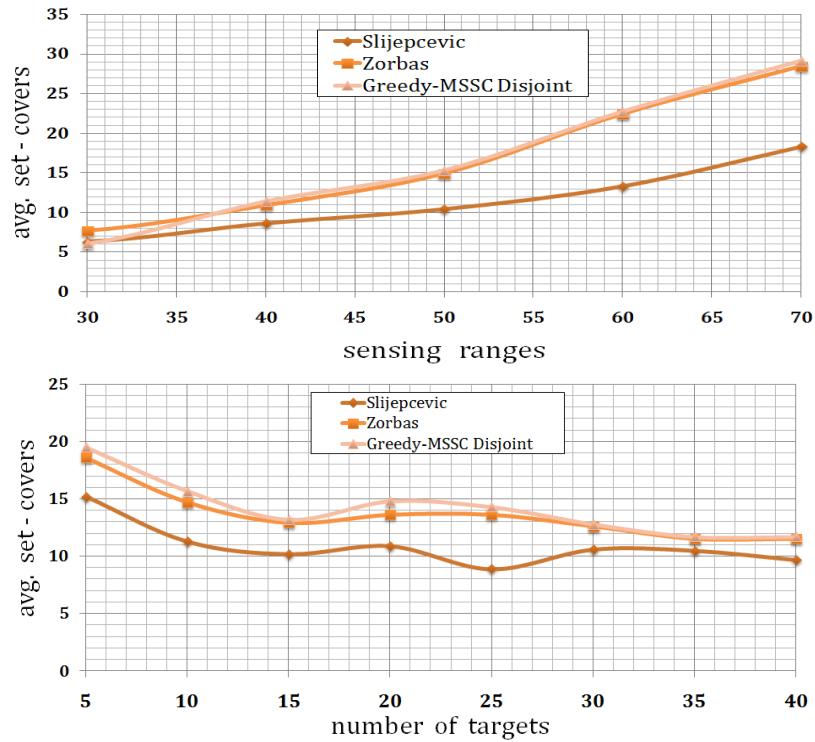
increased from 4 to 18. The non-disjoint constraint associated with the targets coverage redundancy may improve the network lifetime, due to the implication of sensors in multiple set covers.



**Figure 3. Network Lifetime vs. Number of Sensors**

#### 5.4. Experiment 4

Finally, we vary the number of targets and the sensing range to observe the impact on the number of set covers. The general remark is that the same tendency is kept when we compare the number of disjoint set covers computed by the algorithms (Figure 4). However, when the sensing range increases, this may improve consistently the number of sets since each sensor may cover more targets. This implies a growth in the coverage redundancy, which may increase at the same time the number of sensors that cover the critical targets. When the number of targets increases while fixing the number of sensors, the likelihood of the coverage redundancy may be reduced, and the number of set covers may also decrease.



**Figure 4. Average Set Covers vs. Sensing Range and Number of Targets**

## 6. Related Works

For monitoring targets, sensors can be deployed in two ways: either deterministically (by manual placement) or randomly (by dropping sensors randomly in proximity of targets). When deployed deterministically, the objective of the target coverage problem is to settle a compromise between optimizing coverage quality and minimizing design costs. This problem is addressed in the literature as the sensor placement optimization problem. In the case of random deployment, the goal is to control data detection and communication, in order to optimize the coverage of targets in the area, and extend the network lifetime. In this case, the main goal can be to design efficient algorithms to perform sensors sleep scheduling mechanisms. Sometimes, authors refer to other constraints and design choices such as connectivity and other QoS constraints.

### 6.1. Sensors Placement Optimization Problem

Let us consider that we have  $T$  types of sensors, each of them with a cost  $c_t$  and a sensing range  $r_t$ , with  $t = 1 \dots T$ . We suppose that the greater the sensing range, the higher the cost of sensor. Let us call  $D_t(s_i)$  the subset of targets that can be covered by a sensor placed at site  $i$ .  $D_t(s_i)$  is defined by :

$$D_t(s_i) = \{z_j: d(s_i, z_j) \leq r_t, i = 1, \dots, n, j = 1, \dots, m\}$$

With  $d(s_i, z_j)$  denoting the Euclidean distance between sensor  $s_i$  and target  $z_j$ . A Boolean function  $\phi_i^t$  is defined such that  $\phi_i^t = 1$  if a sensor of type  $t$  is placed at the position  $i$ , and  $\phi_i^t = 0$  otherwise. The problem of sensors placement optimization can be formalized by a linear programming defined by equation (2, 3 and 4):

$$\begin{aligned} \text{Minimize : } & \sum_{i=1}^n \sum_{t=1}^T c_t \phi_i^t \\ \text{Subject to: } & \sum_{t=1}^T \sum_{j \in D_t(s_i)} \phi_j^t \geq k, j = 1, \dots, n \\ & \sum_{t=1}^T \phi_i^t \leq 1 \end{aligned}$$

The first objective of the problem is to minimize the number and the overall cost of deployed sensors. Some applications need some robustness, which result in a  $k$  – coverage. When a target  $z_j$  is  $k$  – covered, it can resist to at most  $k - 1$  sensors breakdown. The second constraint guarantees that each site cannot be occupied by more than one sensor.

The mathematical model given below (equation 2, 3 and 4) presents a basic framework of the sensors placement optimization problem. Many variants of this model have been proposed in the literature, depending most of times, on the design choices linked to the problem. Some approximation algorithms like greedy algorithms [14–15], genetic algorithms [16-17] have been proposed, with the aim to efficiently solve the problem, when it is no longer possible to perform an exhaustive search. Approximation algorithms propose sub-optimal solutions that can avoid the complexity of the exhaustive search, which can solve efficiently the problem and in a reasonable running time.

## 6.2. Sleep Scheduling Mechanisms

Target coverage problem related to network lifetime maximization could be viewed as the first challenging constraint in designing target coverage application in random deployment. The sleep scheduling mechanism is often performed by grouping sensors into different sets that will be activated consecutively. Sometimes the computed sets may satisfy other constraint that go along with the coverage requirement. In what follow, we present some works on TCP related to connectivity and QoS constraints.

**6.2.1. Disjoint set covers:** In disjoint set cover, sensors are partitioned into disjoint sets that will be activated consecutively. Two sets are called disjoint if their intersection is an empty set, this means  $C_i \cap C_{i'} = \emptyset, \forall i \neq i'$ . If all the sensors have the same amount of initial energy and the same consumption rate in active mode, and that sensors are continuously active during set covers scheduling, all sensors in one set cover will die at the same time. The objective of the TCP can be converted to the problem of finding the maximum number of disjoint set covers, each of them satisfying the coverage requirement. With such a management, the lifetime of the network is multiplied by an equal factor to the number of computed set covers.

Slijepcevic and Potkonjak [1] introduced first the problem in the context where targets are represented by structures called fields, and they proposed a greedy algorithm ( $MC - MCH$ ) that generates disjoint sets by selecting consecutively the sensor having the maximum profit; the profit function apply a specific treatment on the most sparsely

covered targets, often called critical target. The algorithm has worst case runtime of  $O(n^2)$ , where  $n$  is the number of sensors.

Cardei and Du tackled the disjoint set cover (DSC) problem in [2]. In order to compute the maximum number of set covers, they first transform DSC into a maximum-flow problem, which is then formulated as a mixed integer programming (MIP). Based on the solution of the MIP, a heuristic is designed for computing the number of set covers.

Zorbas and al. [3] proposed a greedy algorithm called  $B\{GoP\}$  producing disjoint sets, based on a refinement of the sensors selection strategy introduced in [1]. When partitioning sensors, these latter are first classified into four categories: "best", "good", "ok", "poor", according to their coverage status. If one candidate sensor belongs to the "best" class, the selection is trivial; in the other case all sensors are evaluated using a profit function for performing the best selection. The algorithm has complexity  $O(n^2m)$ , where  $n$  and  $m$  represent the number of sensors and targets.

**6.2.2. Non-disjoint set covers:** In non-disjoint set cover partitioning, sensors are partitioned into non-disjoint sets that will be activated consecutively. Two sets are called non-disjoint if their intersection does not result in an empty set, *i.e.*,  $C_i \cap C_{i'} \neq \emptyset, \forall i \neq i'$ . In this case, a sensor  $s_k$  can be member of at most  $d$  set covers such that  $d = e_k/e_i^k$ .

Cardei and al. [4] extended their works in [2], with the objective to generate the maximum number of non-disjoint set covers. They first propose a Linear Programming solution with high complexity  $O(m^3n^3)$  where  $m$  is the number of set covers and  $n$  the number of sensors. They proposed also a greedy algorithm called Greedy-MSD with a lower complexity  $O(dm^2n)$ , where  $d$  represents the number of sensors that cover the critical target and  $m$  the number of targets. The same strategy of Slijepcevic and Potkonjak [1] is used in the Greedy-MSD, to avoid including more than one sensor covering the critical target in a set cover.

Zorbas and al. [5] proposed a greedy algorithm based on an improvement of  $B\{GoP\}$ , which generates non-disjoint set covers. Two variations of the algorithm are presented: the *Static – CCF* and the *Dynamic – CCF* that differ on the way the weight used for describing the association of a sensor with a critical target is calculated (statically or dynamically).

### 6.3. Connected Target Coverage

In connected target coverage problem, the main challenge is to efficiently monitor a finite set of targets, while maintaining the connectivity of the network. The lifetime of the network consists on the time duration during which coverage and connectivity requirements are both satisfied. Interesting solutions are proposed in the literature, most of time based on weighted graph theory concepts to solve connectivity.

In [7] the authors address the connected target coverage problem. They provided a generic greedy heuristic algorithm to solve the problem. The algorithm builds in the first time a set cover that provides full coverage of targets. The sensors selection's strategy is based on critical targets avoidance policy. In the second time, the algorithm checks whether the selected sensors are connected to the base station or not. If connectivity is reached, relay nodes are added to the set cover and the lifetime of

selected sensors is updated according to a predefined energy consumption model. Performance evaluation as well as optimization features of *OCCH* are highlighted; however complexity analysis was not provided.

In [6], Zhao and Gurusamy modeled the connected target coverage as the maximum cover tree (*MCT*) problem. A cover tree is a set of sensors that satisfies full target coverage and connected to the sink through a virtual backbone. They first show the NP-completeness of the *MCT* problem, and provided an upper bound lifetime for the *MTC* problem. They also proposed a greedy heuristic *Communication Weighted Greedy Cover* that runs in three phases to guarantee coverage and connectivity. In phase 1, link weights is assigned to each link between sensors and sink reflecting both communication energy consumption in this link and the residual energy of the sensor; then a minimum weight communication tree *MWCT* is built using Dijkstra's shortest path tree algorithm. In phase 2, a set cover is built through a greedy selection of sensors having the highest profit values until all targets are covered. And finally, in phase 3, the cover tree is extracted from the *MWCT* and the selected sensors in phase two. The operational time of the constructed cover is calculated and the residual energy of selected sensors is updated according to a predefined energy consumption model.

#### 6.4. Target Coverage under QoS Constraint

In [8], Pyun, *et al.*, address the multiple-target coverage problem. They fixed the objective to propose a sensors scheduling algorithm that considers the transmitting energy according to the number of targets covered by the sensors and removes the redundancy of overlapped targets, *i.e.*, targets covered by multiple sensors simultaneously. Since the energy used to transmit collected data will be in proportion to the number of targets that the sensor covers, and that some targets may be sensed by adjacent sensors at the same time, they introduce *RSSA*, an algorithm that selects one of the redundant sensors (*responsible sensor*) which is responsible for transmitting the overlapped data to the base station, in order to avoid multiple transmission of the same data. *RSSA* is integrated in a heuristic algorithm that proceed by an iterative selection of sensors that both cover the critical target and the maximum number of uncovered targets to build sensors set; after a set has been formed, *RSSA* is executed for electing responsible sensors. The complexity is expressed as  $O(jm^2n)$ , however consistent simulation results comparisons are missing since it is just compared with the conventional scheme.

In [9], Diop, *et al.*, proposed a sensors scheduling mechanism in probabilistic target coverage. The target detection model is based on the path loss lognormal shadowing model [13], which takes into account distance parameter and the sensor's physical characteristics. The major change in this study compared with earlier works lies on the consideration of a non-idealistic sensor coverage model. They attempts to partition sensors into a maximum number of set covers that guarantee full targets coverage. The proposed algorithm is a greedy heuristic, which selects iteratively the sensors having the greatest cost evaluation function. This latter is measured relative to the number of critical and uncovered targets covered by the evaluated sensor, as well as the coverage probability values between sensors and targets. The algorithm has worst-case complexity of  $O(dn^2m)$ , and the minimum and maximum targets detection values provided by set covers are also estimated in the performance evaluation.

## 7. Conclusion

In this paper, we address the TCP in wireless sensor networks. When several battery-powered sensors are dispersed in a field to monitor a finite set of targets, managing sensors energy dissipation is useful to obtain a long-lived network. A well-known mechanism to perform energy saving is to schedule sensors activities, such as to keep active only the optimum necessary to satisfy the coverage requirement. To achieve this, we used the set cover approach and proposed a greedy algorithm that selects the minimum number of sensors to cover the entire set of targets. Even if interesting solutions have been proposed, they are usually assuming a lot of considerations which make the study impractical with realistic sensor networks. We also provide a brief literature review on TCP linked with connectivity and QoS requirements. We are willing to study, in our future work, connectivity and QoS constraints such as k-coverage problem and the probabilistic target coverage.

**Table 1. An Overview of Greedy Solutions for Solving the Target Coverage Problem Associated with other Design Constraints**

<u>Authors and reference</u>	<u>Addressed problem</u>	<u>Sensor selection strategy</u>	<u>Generated set covers</u>	<u>Complexity analysis</u>
Slijepcevic, <i>et al.</i> , [1] – (MC-MCh)	CLMP <sup>1</sup> with full fields coverage	Iterative selection favoring $cs^2$ . covering the maximum nb. of targets	DSC <sup>3</sup> that satisfy full fields coverage	$O(n^2)$ without considering fields computation
Cardei, <i>et al.</i> , [2] – (DSC)	CLMP with full targets coverage	Transform the problem to a max. flow problem - Use a MIP-based heuristic to compute sets	DSC that satisfy full targets coverage	Complexity of the Mixed Integer Programming (MIP)
Zorbas, <i>et al.</i> , [3] – (B{GoP})	CLMP with full targets coverage	Sensors classification in 4 classes – Iterative sensors selection managing the <i>ct.</i>	DSC that satisfy full targets coverage	$O(n^2m)n$ and $m$ : sensors $ S_0 $ and targets $ Z_0 $
Cardei, <i>et al.</i> , [4] – (Greedy-MSc)	CLMP with full targets coverage	LP-based solution – Iterative selection of sensor covering the max. number of uncovered targets with enough energy	Non- DSC that satisfy full targets coverage	$O(m^3n^3)$ for LP – $O(dn^2m)$ for Greedy-MSc
Zorbas, <i>et al.</i> , [5] – (Static/dynamic-CCF)	CLMP with full targets coverage	Improved B{GoP} – Weight calculation describing sensor	Non-DSC that satisfy full targets coverage	$O(km^2n)$ - $k$ : nb. set covers a sensor can be member of
Zhao, <i>et al.</i> , [6] – (CWGC)	Maximum connected target coverage	Use Djisktra's SPT to build a min. weighted tree- extract cover sets from the tree	Non-disjoint connected set covers	$O(n^2)$ + $O(\min(n, m))$
Zorbas, <i>et al.</i> , [7] – (OCCH)	CLMP and connectivity	Sensors selection avoiding <i>ct.</i> – Add relay sensors to connect with the BS	Non-disjoint connected set covers	Omitted
Pyun, <i>et al.</i> , [8] – (RSSA)	CLM under QoS constraint	Same strategy in [1]- Elect 1 responsible sensor for sending data	Set covers with responsible sensors for each overlapping sensors	$O(jm^2n)$

<sup>1</sup> Coverage Lifetime Maximization Problem

<sup>2</sup> Critical sensor, i.e. sensors covering critical targets

<sup>3</sup> Disjoint Set Covers

		gathered by overlapped sensors	patterns	
Diop, <i>et al.</i> , [9] – (WMSC)	CLMP with probabilistic target coverage	Distance-based sensors selection using a cost effectiveness function	Non-DSC that satisfy full targets coverage- Coverage estimation of set covers	$O(dn^2m)$

## Acknowledgement

This paper is a revised and expanded version of a paper entitled “Managing Target Coverage Lifetime in Wireless Sensor Networks with Greedy Set Cover” presented at MulGraB 2014, Hainan, China, 20-23 Dec.2014.

## References

- [1]. S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks”, In Communications, 2001. ICC 2001. IEEE International Conference on IEEE, vol. 2, (2001), pp. 472-476.
- [2]. M. Cardei and D. Z. Du, “Improving wireless sensor network lifetime through power aware organization”, Wireless Networks, vol. 11, no. 3, (2005), pp. 333-340.
- [3]. D. Zorbas, D. Glynos, P. Kotzanikolaou and C. Douligeris, "B {GOP}: An adaptive algorithm for coverage problems in wireless sensor networks”, In 13th European Wireless Conference, EW, (2007) April.
- [4]. M. Cardei, M. T. Thai, Y. Li and W. Wu, “Energy-efficient target coverage in wireless sensor networks”, In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE, vol. 3, (2005) March, pp. 1976-1984.
- [5]. D. Zorbas, D. Glynos, P. Kotzanikolaou and C. Douligeris, “Solving coverage problems in wireless sensor networks using cover sets”, Ad Hoc Networks, vol. 8, no. 4, (2010), pp. 400-415.
- [6]. Q. Zhao and M. Gurusamy, “Lifetime maximization for connected target coverage in wireless sensor networks”, IEEE/ACM Transactions on Networking (TON), vol. 16, no. 6, (2008), pp. 1378-1391.
- [7]. D. Zorbas and C. Douligeris, “Connected coverage in WSNs based on critical targets”, Computer Networks, vol. 55, no. 6, (2011), pp. 1412-1425.
- [8]. S. Y. Pyun and D. H. Cho, “Power-saving scheduling for multiple-target coverage in wireless sensor networks”, Communications Letters, IEEE, vol. 13, no. 2, (2009), pp. 130-132.
- [9]. B. Diop, D. Diongue and O. Thiaré, “Target coverage management in wireless sensor networks”, IEEE International Conference on Wireless networks. IC Wise, (2014).
- [10]. D. S. Johnson, “Approximation algorithms for combinatorial problems”, In Proceedings of the fifth annual ACM symposium on Theory of computing, ACM, (1973) April, pp. 38-49.
- [11]. U. Feige, “A threshold of  $\ln n$  for approximating set cover”, Journal of the ACM (JACM), vol. 45, no. 4, (1998), pp. 634-652.
- [12]. W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-efficient communication protocol for wireless micro sensor networks”, In System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on IEEE, (2000) January, pp. 10.
- [13]. T. S. Rappaport, “Wireless communications: Principles and practice”, In New Jersey: Prentice Hall, (1996), pp. 1584–1592.
- [14]. X. Xu and S. Sahni, “Approximation algorithms for sensor deployment”, Computers, IEEE Transactions on, vol. 56, no. 12, (2007), pp. 1681–1695.
- [15]. J. Wang and N. Zhong, “Efficient point coverage in wireless sensor networks”, Journal of Combinatorial Optimization, vol. 11, no. 3, (2006), pp. 291–304.
- [16]. J.-H. Seo, Y.-H. Kim, H.-B. Ryou and S.-J. Kang, “A genetic algorithm for sensor deployment based on two-dimensional operators”, In Proceedings of the 2008 ACM symposium on Applied computing, ACM, (2008), pp. 1812–1813.
- [17]. Q. Wu, N. S. V. Rao, X. Du, S. S. Iyengar and V. K. Vaishnavi, “On efficient deployment of sensors on planar grid”, Computer Communications, vol. 30, no. 14, (2007), pp. 2721–2734.
- [18]. M. Cardei and J. Wu, “Coverage in wireless sensor networks”, Handbook of Sensor Networks, (2004), pp. 422-433.

## Author



**Babacar Diop**, he received a B.Sc. and M. Sc. degrees in Computer Science at Gaston Berger University. I am currently working as a PhD student at LANI laboratory at Gaston Berger University of Saint-Louis, Senegal. My research essentially focused on Wireless networks and Wireless sensor networks, and more particularly algorithms and heuristics design in Wireless sensor networks.



**Dame Diongue**, he received a B.Sc and M. Sc. degrees in Computer Science at Cheikh Anta Diop University of Dakar, Senegal, respectively in 2007 and 2011. I received a PhD student in Computer Science at Gaston Berger University of Saint-Louis, Senegal, in 2014. My research essentially focused on Networking, Wireless Mesh Networking and Wireless Sensor Networks, and particularly in scheduling algorithms and coverage heuristics design in Wireless Sensor Networks.



**Ousmane Thiare**, he received a B.Sc. and M.Sc. degrees both in Applied Mathematics from Gaston Berger University of Saint-Louis, Senegal in 1997, and 2000 respectively. I received also a M.Sc. in Computer Science at Polytech' Tours, France in 2002. I received a PhD degree in Computer Science from the University of Cergy Pontoise France, in 2007. I was an Assistant Professor at Gaston Berger University of Saint-Louis, in 2002. I am now an Associate Professor at Gaston Berger University of Saint-Louis, Senegal since 2010. I am also the coordinator of the Master Sc. of Computer science. Since November 2013, I am the Director of the IT Computing Center of the University Gaston Berger of Saint-Louis. I am also member of the Scientific and Pedagogical Council of the Doctoral School of Sciences and Technologies of the University Gaston Berger of Saint-Louis. My research focuses on field applications of Parallel and Distributed Systems, Peer-to-Peer Systems, Grid and Cloud Computing, Wireless Ad Hoc and Sensor Networks in particular. My work appears in highly respected international journals and conferences.