A Weight-based Greedy Algorithm for Target Coverage Problem in Wireless Sensor networks

Babacar DIOP
Gaston Berger University
Saint-Louis, Senegal BP: 234
Email: babacardiop96@yahoo.fr

Dame DIONGUE
Gaston Berger University
Saint-Louis, Senegal BP: 234
Email: ddiongue.ep2112812@ugb.edu.sn

Ousmane THIARE
Gaston Berger University
Saint-Louis, Senegal BP: 234
Email: ousmane.thiare@ugb.edu.sn

Abstract—**Recent improvements in affordable and efficient integrated electronic devices have enabled a wide range of applications in the estate of wireless sensor networks. An important issue addressed in wireless sensor networks is the coverage problem. This latter is centered on a fundamental question: how well do the sensors observe the physical space? A major challenge in coverage problem is how to maximize the lifetime of the network while ensuring coverage of a set of targets. To achieve this, the usual process, consists on scheduling sensors activity, which enables energy dissipation control. Scheduling process goes by activating sensors by round such that in each round, only one subset of sensors that satisfies the coverage requirement is activated, while all other sensors are in a low energy mode and will be activated later. In this paper, we propose a weight-based greedy algorithm (WGA) which organizes sensors in multiple subsets. Our objective is to partition an initial set of sensors into a maximum possible number of sensors set covers (SSCs), which can completely monitor targets in a region of interest. Performance evaluation of WGA have proven its efficiency over some well-known algorithms proposed in the literature, in term of computed set covers.**

Keywords: ***Wireless Sensor Network, Target coverage, Sensor Set Cover, Greedy Algorithm***.

I. INTRODUCTION

Wireless sensor networks (WSNs) form one category of wireless networks, consisting of tiny devices called sensors. Sensors are often deployed in large number in a region of interest to monitor physical phenomena (military targets, climate changes, . . .). Sensors are usually battery powered, and can be equipped with sensor module(s) and communication interface to capture data from physical space and report them to a sink (base station) [1]. Sensors are characterized by limited resources in terms of energy and communication system. These physical constraints are some issues of major concern in this area. This is the case of the energy constraint which impacts widely on the lifetime of WSN.

Coverage problem is one of the fundamental issues in wireless sensor networks. It refers to how well physical space is observed by sensors. An important part of the study of coverage in WSN is devoted to target coverage problems. The main objective of target coverage is to monitor a discrete set of targets in a region of interest. Target coverage echoes in various fields (military, medical, industrial), where the studied subjects (military targets, medical patient, etc.) can be observed or followed by nearby sensors, embedded or attached to it. Coverage Control Service (CCS) ensures optimal quality of coverage by providing optimization techniques, for a better control of sensors activities.

In this work, we address the sensors set covers maximization problem. We provide a synthesis of some well-known techniques from the literature and we then propose a new centralized solution for sensors set covers construction. The proposed solution is given in the form of a greedy algorithm called WGA (Weight-based Greedy Algorithm), that proceeds by partitioning sensors in multiple set covers, such that each set cover ensures complete coverage of targets. The main goal of WGA is to maximize the number of computed set covers from the partition, which can lead to a network lifetime extension.

The rest of the paper is structured as follows. In section II, we review the related works addressing target coverage problem. Section III gives a formulation of the sensors set covers maximization problem in wireless sensor networks, and introduces some discussion on the subject in the literature, highlighting sensors selection strategies during set covers construction. Section IV describes in detail the proposed algorithm (WGA). Section V presents performance evaluation results of WGA and section VI concludes the paper.

II. RELATED WORKS

For targets monitoring, sensors are usually deployed in two ways : either deterministically (by manual placement) or randomly (by dropping sensors randomly in proximity of targets). When deployed deterministically, the CCS tries to settle a compromise between optimizing coverage and minimizing design costs. In the case of random deployment, the goal is to control data detection and communication, in order to maximize the coverage degree of the targets [1] in the area, and extend the network lifetime. In this case, the process most frequently used consists on scheduling sensors activities in order to ensure an optimal coverage quality of service.

A. Sensors Placement Optimization Problem

*1) **Problem Modeling**:* 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 $t = 1, .., T$. We suppose that the greater the

[1]The coverage degree of a target refers to how well this target is covered, for example the number of sensors covering this target.

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 :

$$D_t(s_i) = \{z_j / d(s_i, z_j) \leq r_t,\ i = 1, .., n,\ j = 1, .., m\} \quad (1)$$

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

$$Minimize\ :\ \sum_{i=1}^{N} \sum_{t=1}^{T} c_t \phi_i^t \quad (2)$$

$$Subject\,to\ :\ \sum_{t=1}^{T} \sum_{j \in D_t(s_i)} \phi_j^t \geq k,\ j = 1, ..., N \quad (3)$$

$$\sum_{t=1}^{T} \phi_i^t \leq 1 \quad (4)$$

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 litterature, depending most of times, on the design choices linked to the problem.

*2) **Solving techniques**:* Many optimization techniques can be used to model and resolve the sensors placement optimization problem. When the search space is not too large, an exhaustive search can be performed in order to find the global optimum. The complexity of the exhaustive search is defined by equation 5 :

$$C = 2^I - 1 \quad (5)$$

I represents the number of available sites in the region of interest. According to equation 5, the complexity of the exhaustive search increases exponentially with the number of available sites. Some approximation algorithms like greedy algorithms [2]–[4], genetic algorithms [5], [6] and other optimization techniques [7]–[9] have been proposed, with the aim to efficiently resolve 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.

*B. **Target Coverage Lifetime Maximization***

In random deployment, the number of deployed sensors exceeds generally the optimum needed to satisfy the coverage of all the targets. This scenario helps filling the gap of coverage breach, due to the randomness. Some sensors may cover more targets than other after the deployment, and some targets may be covered by several sensors simultaneously. To maximize the network lifetime, sensors are often partitioned into several sensors set covers which are activated by round. A sensors set cover (SSC) is a group of sensors that can monitor targets in an area of interest. A SSC is complete if once activated, all targets in the sensing field can be monitored. Since SSCs are successively activated, the lifetime of the network is equal to the sum of all SSCs lifetimes. Maximizing network lifetime is equivalent in this case to calculate the optimal number of SSCs, which is an Np-complete problem [10]. Sub-optimal solutions have been proposed in the literature particularly those based on greedy algorithms, linear/integer programming, and other optimization techniques. Protocols can be categorized into two classes: centralized and distributed.

A common approach to centralized protocols is to represent the problem as an optimization problem, and to use an approximation technique to solve it. Centralized approximation algorithms proceed by an iterative selection of sensors to form SSCs. The selection is carried out in several rounds, focusing on sensors whose profits are better and covering less sparsely covered targets. The profit of a sensor can be measured relatively to the number of targets it covers, and the remaining energy of this sensor. The first centralized approaches [10]–[12] focused on the problem of finding an optimal number of disjoints SSCs, each being able to cover all targets. Two SSCs C_1 and C_2 are disjoints if $C_1 \cap C_2 = \phi$, where $C_1 \neq C_2$. Furthermore it has been shown that non-disjoint SSCs tend to better improve network lifetime [13], [14], due to sensors inclusion in multiple sets. Two SSCs C_1 and C_2 are non-disjoint if $C_1 \cap C_2 \neq \phi$, where $C_1 \neq C_2$. Most of these first centralized protocols were extended later to address the problem of target coverage lifetime maximization in the context where sensors can dynamically adjust their sensing range [15]. Some works have been devoted to the study of coverage and connectivity problems in one formulation. To satisfy connectivity, techniques like graph search algorithms [16], shortest path algorithms [17] and spanning and cover trees [18] are frequently used. Covering all targets simultaneously can be a too strict requirement for certain applications. For this reason, some works [19]–[21] focused on partial coverage problem, where some targets may not be covered by sensors of one SSC, i.e. $\cup_{s_i \epsilon C_k} Z(s_i) \neq Z_0$, where $k = 1, ..., K$. The often raised problem is the minimization of breach rate, i.e. the proportion of uncovered targets.

III. Maximum Disjoint SSCs Problem

*A. **Problem Parameters***

Let us assume that n sensors $s_1, ..., s_n$ are randomly deployed in a region of interest to monitor m targets $z_1, ..., z_m$.

We consider Z_0 as the initial set of targets, and S_0 the initial set of sensors. Each sensor has a predefined sensing range called r_s, and a sensor covers a target if the Euclidean distance between sensor and target is less than r_s. Also we consider that the base station dispose sensors and targets coordinates, and can compute for each target z_j the subset of sensors covering it, called $S(z_j)$. For each sensor s_i, we designate by $Z(s_i)$ the subset of targets covered by s_i.

B. *Definition*

The main objective of maximal disjoint SSC problem can be addressed as follow:

How well to partition sensors initial set (S_0) in multiple disjoints SSCs $C_1, ..., C_K$ such that each SSC C_k satisfies complete targets coverage?

Centralized algorithm are executed on the base station, and the results are communicated directly to sensors for execution. They usually take as input S_0 and Z_0, as well as $S(z_1)$,...,$S(z_m)$ and $Z(s_1)$, ..., $Z(s_n)$. The output can be a collection of SSCs $C = \{C_1, ..., C_K\}$. Each SSC C_k is a subset of S_0 and the union of targets covered by sensors in C_k is equal to Z_0. When producing disjoint SSCs, one sensor can participate to at most one SSC. Centralized algorithms try to maximize the total number of SSCs $|C|$ in order to extend the network lifetime as much as possible. Since SSCs are successively activated, then the network lifetime can be multiplied by a factor equal to $|C|$.

C. *Sensor Selection Strategy*

Let us call z_{min}, with ($z_{min} \in Z_0$), the more sparsely covered target after the deployment scenario; z_{min} is called the critical target [11]. Then $|S(z_{min})|$ represents the cardinality of the subset of sensors covering z_{min} and is called the theoretical maximum (*max_theo*). This latter designates theoretically the maximum number of computable disjoint SSCs; this is due to two conditions: the first is that each SSC guarantees complete coverage of targets and the second is each sensor cannot be part of more than one SSC due to the disjunction.

In order to compute the closest number of SSCs to *max_theo*, an efficient sensor selection strategy must be adopted for a well repartition of sensors through set covers. An inefficient sensor selection strategy affects more particularly sensors covering the critical target(s) and those which covered targets are sparsely covered. Furthermore it will compromise the ability of the algorithm to produce a near optimal number of SSCs. For example if k ($k \geq 2$) sensors covering z_{min} are included simultaneously in one SSC, then the number of produced sets is reduced from k.

D. *Weight-Based Profit Function*

Recall that to maximize the number of disjoint set covers, sensors must be strictly selected, in order not to include simultaneously more than one sensor covering the critical target(s) in one set cover. Classical greedy algorithms select at each stage, the sensor covering the largest number of sparsely covered targets and remove them from the targets set. Then the selected sensor is added to the current set cover and the same process is iterated until targets set is empty. However, when several sensors cover the same number of targets, then the node with the lowest index is chosen. This way of selecting node with lowest index, is a random choice. For example, suppose that we have in one stage during set covers generation $|S_{cur}|$ remaining sensors, and $|Z_{cur}|$ the number of remaining targets. If there is at least one target in Z_{cur} being covered by one sensor s_i from S_{cur}, then s_i must be selected during this stage, independently of the number of targets covered by s_i. The probability of selecting a sensor during one stage must depend both on the number of targets covered by this sensor and the way these targets are covered (sparsely or frequently). Sensors covering the more sparsely covered targets must be selected first. For example, if one target z_0 is covered by only one sensor, then this latter will be selected during this stage. If k ($k \geq 2$) sensors cover z_0, then the selection will depend on the number of already covered and uncovered targets that each of these sensors covers. In this case, if a sensor covers only uncovered targets during this stage, then the probability of selecting this sensor must be greater. A suitable solution is to compute for each sensor, a weight-based coverage measure W_{Cov} which estimates its selection likelihood considering Z_{cur}, the set of targets to be covered. For this, we establish first for each sensor s_i and target z_j the coverage relationship function $Cov(s_i/z_j)$; this latter determines how well each target is covered by a given sensor, and its relation with other sensors in the field covering this target. The coverage relationship function is defined by equation 6:

$$Cov(s_i/z_j) = \frac{I(s_i, z_j)}{\sum_{t=1}^{n} I(s_t, z_j)} \tag{6}$$

Where $I(s_i, z_j)$ represents the characteristic function which is equal to 1 if s_i covers z_j, and 0 or else.

As illustrated in equation 6, the coverage relationship function between s_i and z_j is equal to the ratio between the characteristic function and the number of sensors covering this target. The value of this function is equal to 0 if z_j is not covered by the given sensor. The highest value of the coverage relationship function (1), corresponds to the case where z_j is covered by only one sensor s_i.

Based on equation 6, we define the weight-based coverage measure considering Z_{cur}, i.e $W_{Cov}(s_i/Z_{cur})$. This latter determines the coverage relationship between one sensor s_i and the subset of targets that are included in Z_{cur}. W_{Cov} is defined by equation 7:

$$W_{Cov}(s_i/Z_{cur}) = (1/m) / \sum_{z_j \in Z_{cur}} Cov(s_i/z_j) \tag{7}$$

Where m is the total number of targets. Equation 7 is applied to define a new algorithm called WGA, which uses as profit function the weight-based coverage measure considering Z_{cur}. WGA performs in multiple stages and greedily selects a sensor with the maximal coverage measure, until all targets are covered.

IV. ALGORITHM DESCRIPTION

A. *Setup*

Before WGA starts its execution, input data are built during the setup phase. The performed task during this phase is done by the base station which computes for each sensor s_i, the subset of neighboring targets it covers, and for each target z_j the neighboring sensors which cover it. We define the input parameters of WGA by the following variables :

$Z_0 = \{z_1, ..., z_m\}$: the initial targets set.

$S_0 = \{s_1, ..., s_n\}$: the initial sensors set.

$\Delta_{S_0} = \{Z(s_1), ..., Z(s_n)\}$: the collection of subsets of neighboring targets that each sensor s_i covers.

$\Omega_{Z_0} = \{S(z_1), ..., S(z_m)\}$: the collection of subsets of neighboring sensors covering each target z_j.

$max\_theo = min\{|S(z_j)| : S(z_j) \in \Omega_{Z_0}, j = 1, .., m\}$: the theoretical maximum.

Once input data calculated, WGA is executed with the objective to produce a maximum number of disjoint SSCs from S_0, such that each SSC satisfies coverage of all targets in Z_0. The pseudo-code of WGA is presented below.

B. *Pseudo-code of WGA*

The algorithm maintains a list of available sensors S_{cur}, initialized to the initial set of sensors (S_0). The first "while" loop (line 10) tests the availability of sensors in S_{cur}, and checks whether all targets in Z_0 can be covered by sensors in S_{cur}. If this is the case then a new SSC may be constituted. An empty collection is created to build a new SSC, and the current target list Z_{cur} is initialized to Z_0. The algorithm passes the control to the second "while" loop, which is responsible of the new SSC construction. A list of not yet covered targets is kept by the second loop, which tries to select a sensor having the greatest profit function (W_{Cov}) during this stage. The weight-based profit function is used, to determine which of these sensors that belong to S_{cur}, has the maximum profit to cover targets in Z_{cur}. This sensor is called $s_{selected}$. Once selected, all targets covered by $s_{selected}$ are removed from the uncovered targets list (Z_{cur}); $s_{selected}$ is also removed from S_0, and is added to the SSC under construction (C_{cur}). The same process is iterated until Z_{cur} becomes empty; meaning that there is no more target to cover. The whole sub-collection is added to the ultimate collection, C_{total}. At this stage of the execution, the algorithm checks whether the theoretical maximum has been reached or not. If this is the case, no further SSC can be generated: the collection is returned and the algorithm terminates. Otherwise, a new round of SSC construction will be started.

Algorithm 1: Weight-based Greedy Algorithm

input : $S_0, Z_0, \Delta_{S_0}, \Omega_{Z_0}, max\_theo$
output: C_{total}

// **Initialization Phase**
1 $S_{cur} = S_0$;
2 $Z_{cur} = \phi$;
3 $k = 0$;
4 **foreach** *target* $z_j \in Z_0$ **do**
5 | *compute* $den(z_j)$;
6 **end**
7 **foreach** *sensor* $s_i \in S_{cur}$ **do**
8 | *compute* $W_{Cov}(s_i/Z_0)$;
9 **end**
// **SSCs Generation Phase**
10 **while** $S_{cur} \neq \phi$ **do**
11 | $C_{cur} = \phi$;
12 | $Z_{cur} = Z_0$;
13 | $k = k + 1$;
14 | **while** $Z_{cur} \neq \phi$ **do**
15 | | *Select sensor with max* W_{Cov} *to cover* Z_{cur} ;
16 | | Z_{cvd} = *Targets covered by* $s_{selected}$;
17 | | $Z_{cur} = Z_{cur} - Z_{cvd}$;
18 | | $S_{cur} = S_{cur} - s_{selected}$;
19 | | $C_{cur} = C_{cur} \cup s_{selected}$;
20 | **end**
21 | $C_k = C_{cur}$;
22 | $C_{total} = C_{total} \cup C_k$;
23 | **if** $|C_{total}|$ == *max_theo* **then**
24 | | STOP;
25 | **end**
26 **end**
27 $C_{total} = C_1, ..., C_K$;

C. *Analysis*

WGA algorithm is constituted of two phases : an initialization phase and a SSCs generation phase. During the initialization phase, the density of each target is defined as being the number of sensors covering it. The density is given by equation 8:

$$den(z_j) = \sum_{i=1}^{N} I(s_i, z_j) \tag{8}$$

Thus the coverage relationship function between sensor s_i and target z_j ($Cov(s_i/z_j)$) is equal to the ratio between 1 and the density of this target if z_j is covered by s_i, and 0 otherwise. This value is calculated once at the beginning of the algorithm for each sensor, and does not need to be recalculated for each iteration. Therefore, the profit function $W_{Cov}(s_i/Z_{cur})$ can be computed once during the initialization phase, and updated at each iteration by subtracting the coverage relationship function between s_i and the subset of already covered targets. Thus $W_{Cov}(s_i/Z_{cur})$ during one step in the second phase of the algorithm is obtained by equation 9:

$$W_{Cov}(s_i/Z_{cur}) = \sum_{k=1}^{|Z(s_i) \cap Z_{cur}|} Cov(s_i/z_k) - \sum_{r=1}^{|Z(s_i) - Z(s_i) \cap Z_{cur}|} Cov(s_i/z_r) \tag{9}$$

With $|Z(s_i) \cap Z_{cur}|$ denoting the number of targets in Z_{cur} covered by s_i, and $|Z(s_i) - Z(s_i) \cap Z_{cur}|$ the number of targets covered by s_i and already covered by the sensors contained in the current SSC. The sensors selection process during one round is made on the basis of a comparison of all available sensors profits. This means that the algorithm tests in the first iteration all available sensors in S_{cur} and chooses the one with the maximal profit. If a sensor is selected, all targets covered by that sensor are removed from the set of remaining targets Z_{cur}. For each iteration, the number of sensors (cardinality of S_{cur}) is decreased by 1, and targets set Z_{cur} is reduced by the number of targets already covered by the selected sensors. When Z_{cur} becomes empty, the algorithm resets Z_{cur} with the initial target list Z_0, and rebuilds a new set cover. The algorithm terminates when the list of available sensors S_{cur} is empty or the theoretical maximum is reached. In the worst case, all available sensors are used. Considering that the algorithm takes as input n sensors and m targets to produce set covers, the process is as follow :

- First iteration: n sensors are tested for m targets.
- ...
- N^{th} iteration: $n-(n-1)$ sensors are tested for $m-(n-1) mod\, m$ targets.

The consumed time by WGA for SSCs generation is equal to:

$$T(n) = \sum_{i=0}^{n-1} (n-i)(m - i mod\, m)$$

We conclude that the worst case complexity of WGA is equal to $O(n^2 m)$.

V. PERFORMANCE EVALUATION

To evaluate the performance of WGA algorithm, we simulate a stationary network with sensor nodes and target points randomly located in a 500 m X 500 m area. An implementation prototype of WGA is developed in MATLAB, version 7.10.0.

For topology generation, targets are spread first in the area by generating random coordinates within the area, sensors are then randomly deployed in proximity of targets. We first compare the number of SSCs computed by our heuristic with the number of SSCs computed by MCMC-H proposed in [11]. Secondly, we compare WGA with B{GoP} proposed in [12].

In the first set of experiments, we vary both the number of sensors between 50–150, and the sensing range between 100–300 in order to study the impact of sensors density and sensing range in WGA. For every sensors number value, we repeat the experiment 5 times, for different sensors random positioning, and the average of computed SSCs is output.

In figure 1, we present the average number of SSCs computed by the WGA algorithm, depending on the number of sensors. As the transmission range increases, redundancy also grows and therefore increasing targets coverage likelihood in the network. This results in more disjoint SSCs generation.

Figure 2 compares the average number of SSCs computed by WGA and MCMC-H [11]. We vary the number of sensors from 100 to 500 and we fixed the number of targets, in order

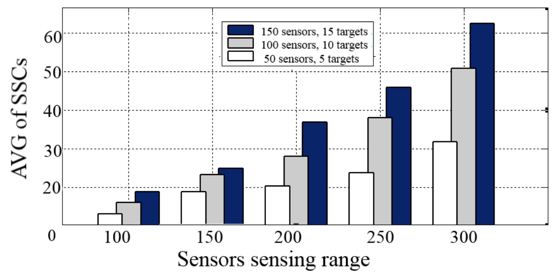

Fig. 1. Average number of SSCs computed by WGA, depending on the number of sensors and the transmission range.

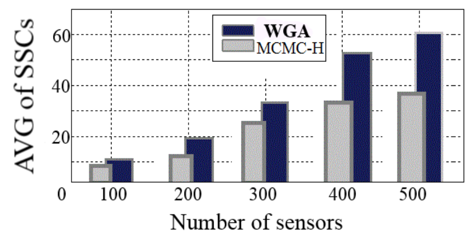

Fig. 2. Average number of covers computed by WGA and the MCMC-H [11].

to study the effect of nodes density on these two algorithms. As the number of sensors increases, the number of disjoint SSCs increases too, since every target would be covered by more sensors. Our algorithm produces constantly more covers, and in some extent a greater number of covers compared with [11].

Figure 3 compares the average number of SSCs computed by WGA and B{GoP} [12]. We vary the number of sensors from 50 to 600 with a fixed number of targets, in order to study the effect of nodes density on the abilities of these two algorithms. The general remark is that the number of SSCs computed by WGA get larger, as the number of sensors becomes greater. These results emphasize the efficiency of the selection strategy driven by the probability function during SSCs construction phase.

In table I, we present a comparison of WGA, MCMC-H and B{GOP} with the theoretical maximum. We can see a very

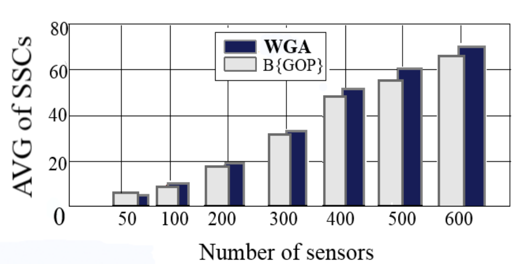

Fig. 3. Average number of covers computed by WGA and B{GoP} [12].

close similarity between the results provided by WGA and the theoretical maximum.

In most cases, the number of SSCs calculated by the WGA algorithm is equal to the theoretical maximum. Considering B{GoP} [12], the theoretical maximum is reached with 50 sensors, whereas for MCMC-H [11] the theoretical maximum has not been reached for all sensors number values. This can be explained consistently by the fact that the sensor selection strategy is more stringent with WGA than with the other two. When the selected sensor during one stage covers k critical targets, where $k > 1$, then theoretical maximum will not be reached. The profit function based on coverage relationship calculation enables to formalize clearly each sensor's contribution, allowing thus a more systematic sensors selection during SSCs generation phase.

TABLE I
AVERAGE OF THE NUMBER OF SSCS COMPUTED BY WGA, B{GOP} [12] AND THE MCMC-H [11], COMPARED WITH THE THEORETICAL MAXIMUM.

Nb.Sensors	*Theo. Max.*	WGA	Zorb.	Slijep.
50	**2**	2	2	1.6
100	**5.4**	5.4	5	3.8
200	**13.6**	13.6	12.2	8.6
300	**14**	14	13.4	9.2
400	**25.4**	25.2	24.2	20
500	**28.4**	28.4	27.2	20.8

VI. CONCLUSION

To summarize, this work address the problem of the maximization of wireless sensor network lifetimes in the context of target coverage applications. Managing sensors energy dissipation through efficient power saving techniques is useful as long as sensors nodes remain battery powered. The approach most frequently used in the literature is based on scheduling sensors activities, a well-known power saving technique which allows them to switch between active and sleep mode. We model this problem by means of cover set generation approach and we propose an algorithm which try to maximize the number of sensors set covers. The proposed algorithm is greedy and centralized, and is based on the calculation of the probability of coverage between each sensor and the set of uncovered targets. Performance evaluation results driven by our simulation show significant results, in term of generated sensors set covers, compared to existing works in this area.

REFERENCES

[1] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[2] Zheng Fang and Jie Wang. Convex combination approximation for the min-cost wsn point coverage problem. In *Wireless Algorithms, Systems, and Applications*, pages 188–199. Springer, 2008.

[3] Xiaochun Xu and Sartaj Sahni. Approximation algorithms for sensor deployment. *Computers, IEEE Transactions on*, 56(12):1681–1695, 2007.

[4] Jie Wang and Ning Zhong. Efficient point coverage in wireless sensor networks. *Journal of Combinatorial Optimization*, 11(3):291–304, 2006.

[5] Jae-Hyun Seo, Yong-Hyuk Kim, Hwang-Bin Ryou, and Seok-Joong Kang. A genetic algorithm for sensor deployment based on two-dimensional operators. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1812–1813. ACM, 2008.

[6] Qishi Wu, Nageswara SV Rao, Xiaojiang Du, S Sitharama Iyengar, and Vijay K Vaishnavi. On efficient deployment of sensors on planar grid. *Computer Communications*, 30(14):2721–2734, 2007.

[7] İ Kuban Altınel, Necati Aras, Evren Güney, and Cem Ersoy. Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks. *Computer Networks*, 52(12):2419–2431, 2008.

[8] Krishnendu Chakrabarty, S Sitharama Iyengar, Hairong Qi, and Eungchun Cho. Grid coverage for surveillance and target location in distributed sensor networks. *Computers, IEEE Transactions on*, 51(12):1448–1453, 2002.

[9] Maulin Patel, R Chandrasekaran, and S Venkatesan. Energy efficient sensor, relay and base station placements for coverage, connectivity and routing. In *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, pages 581–586. IEEE, 2005.

[10] Mihaela Cardei and Ding-Zhu Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3):333–340, 2005.

[11] Sasha Slijepcevic and Miodrag Potkonjak. Power efficient organization of wireless sensor networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 2, pages 472–476. IEEE, 2001.

[12] Dimitrios Zorbas, Dimitris Glynos, Panayiotis Kotzanikolaou, and Christos Douligeris. B {GOP}: An adaptive algorithm for coverage problems in wireless sensor networks. In *13th European Wireless Conference, EW*, 2007.

[13] Mihaela Cardei, My T Thai, Yingshu Li, and Weili 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*, volume 3, pages 1976–1984. IEEE, 2005.

[14] Dimitrios Zorbas, Dimitris Glynos, Panayiotis Kotzanikolaou, and Christos Douligeris. Solving coverage problems in wireless sensor networks using cover sets. *Ad Hoc Networks*, 8(4):400–415, 2010.

[15] Mihaela Cardei, Jie Wu, Mingming Lu, and Mohammad O Pervaiz. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In *Wireless And Mobile Computing, Networking And Communications, 2005.(WiMob'2005), IEEE International Conference on*, volume 3, pages 438–445. IEEE, 2005.

[16] Neeraj Jaggi and Alhussein A Abouzeid. Energy-efficient connected coverage in wireless sensor networks. In *Proceedings of 4th Asian International Mobile Computing Conference, Kolkata, India*, pages 77–86. Citeseer, 2006.

[17] Dimitrios Zorbas and Christos Douligeris. Satisfying coverage and connectivity in bandwidth constrained sensor networks. In *Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on*, pages 390–395. IEEE, 2009.

[18] Qun Zhao and Mohan Gurusamy. Lifetime maximization for connected target coverage in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 16(6):1378–1391, 2008.

[19] Maggie Xiaoyan Cheng, Lu Ruan, and Weili Wu. Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2638–2645. IEEE, 2005.

[20] Maggie X Cheng, Lu Ruan, and Weili Wu. Coverage breach problems in bandwidth-constrained sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 3(2):12, 2007.

[21] Mingming Lu, Jie Wu, Mihaela Cardei, and Minglu Li. Energy-efficient connected coverage of discrete targets in wireless sensor networks. In *Networking and Mobile Computing*, pages 43–52. Springer, 2005.