

# Billiard Algorithm for the K-Coverage Deployment of Sensor Networks

Rabie A. Ramadan\* , Toqeer Ali†, Hatem M. EL-Boghdadi‡

**Abstract**— Wireless Sensor Networks (WSNs) became one of the important networks on that many state-of-the-art applications are established. Sensors used to monitor traffic, borders, and products. In fact, the base for Internet of Things (IoT) applications is WSN. However, deployment of the sensors is still a challenging problem, especially in large-scale networks. Large-scale sensor networks mean a network with large number of sensors to be deployed on a wide monitored field. The main concern of the sensor deployment is coverage of the monitored field. This paper introduces Billiard algorithm to handle the deployment problem of large-scale WSN. Most importantly, we introduce the Billiard algorithm for the K-Coverage problem. Through a set of experiments, we examine the performance of the two algorithms.

**Index Terms**— Wireless Sensor Network, Deployment, Coverage, Billiard Algorithm, K-Coverage, Large Scale Network, Intrusion.

## 1 INTRODUCTION

THE Wireless Sensor Network (WSN) is one of the emergent networks that has been involved in many of our life activities. It has been used in many critical applications such military monitoring, health care, transportation, border monitoring, firefighting, building monitoring, intruder detection, and many others. Such applications require reliable monitoring and stable network. However, sensing nodes suffer mainly from limited energy sources, sensing ranges, and limited communication ranges, in addition to its limited processing and storage capabilities. Therefore, sensors have to be carefully deployed in the monitored area. One of the efficient sensors deployment is K-coverage deployment in which sensors are deployed to cover the monitored field K times. This should maximize detection activity to any event occurs in the monitoring field. However, K-coverage deployment with limited/optimal number of sensing devices is NP Hard problem. In addition, deployment methods used in this regard are either analytical solutions or heuristics. Analytical solutions are not suitable for large scale problems while current heuristics are taking too much time since they have to scan the monitored field at least k-times.

The rest of paper is organized as follows: the problem is stated in the following section; the related work is described in section III; section IV explores the proposed solutions; section V depicts the experimental results; finally, the paper concludes in section VI.

## 2 PROBLEM STATEMENT

The problem of sensors node deployment is an old problem; however, it is still a challenging problem, especially in large scale deployment and large size of the area to be covered. In addition, the k-coverage problem became harder than the one coverage problem. The main idea behind the k-coverage is: Given an Area (A) with length (L) and width (W) and a num-

ber of sensor nodes (S) to be deployed for the purpose of k-coverage. We assume that the area could be in any shape including a square, rectangle, circle, or any shape. However, for simplicity, in this paper, we deal only with squares or rectangles. Each sensor has a sensing range ( $S_s$ ) and communication range ( $S_c$ ). These sensing and communication ranges could be homogenous or heterogeneous. Heterogeneity, in this context, means sensors might be different in their sensing ranges, communication ranges, and their initial energy. We are interested in sensors sensing ranges where we assume that sensors could have a variable sensing range that could be adapted based on the deployment requirements. In other words, a sensor might have different power level to help in communication and sensing, e.g.  $p_1, p_2, p_3, \dots, p_n$  where n is the maximum sensing level per sensor. N could also differ from one sensor to another. Certainly, large sensing ranges affect the lifetime of the nodes; therefore, sensors sensing ranges have to be small as much as possible. In addition, to reduce the sensor network cost, the number of sensors have to be minimum. Therefore, the problem is how to deploy the given number of sensors S on the area A given with almost full coverage. In addition, another question need to be answered which is "Is it possible to cover the monitored field K-times?"

## 3 RELATED WORK

The deployment problem is not a new problem, but it is still one of the complicated problems of Wireless Sensor Networks. There are many variations to the deployment problem including line coverage, border coverage, full coverage, and K-coverage. This also goes for homogenous and heterogeneous version of the problem. In homogenous sensor deployment, sensors are assumed similar in every detail. On the other hand, homogenous sensors mean sensors may differ in one or more of their parameters such as initial energy, sensing rang-

\* Rabie Ramadan, Associate Prof., Hail University, SA. Email. [rabie@rabieramadan.org](mailto:rabie@rabieramadan.org)  
† Toqeer Ali, Assistant Prof., Islamic University, SA. Email. [toqeer@iu.edu.sa](mailto:toqeer@iu.edu.sa)  
‡ Hatem El-Boghdadi, Professor, Islamic University, SA. Email. [helboghdadi@iu.edu.sa](mailto:helboghdadi@iu.edu.sa)

es, communication ranges and processing capabilities.

Loukas et al. in [9] considered the problem of heterogeneous sensor deployment for mobile target detection. The problem is formulated to a set line intersection problem; the problem is evaluated analytically, and it turns out that target detection probability is increasing exponentially with increasing the network size. The same problem is solved by Huang et al. in [8] where the purpose of deployment was to provide efficient and reliable data collection. The authors divided the monitored area into a set of grid cells where the sensors are placed on corners of the squares, squares, correspond to "tilings" of equilateral triangles and hexagons, or grid "tiling". Communication range and sampling rate and their values turned on to be the sensor deployment limitations. Another deterministic deployment method is utilized by Liu et al. in [6]. The authors formulated the sensor placement for enhancing the sensor lifetime, network cost, and maximizing the monitored area coverage in a form of an optimization problem. In most cases the optimal solution provides the optimal number of sensors to be used [10] [11].

However, in a random deployment method, the sensors are sprayed using a flight robot or helicopter. One of the random deployment methods is proposed by Clouqueur et al. in [12]. The purpose of deployment problem is for target detection. In fact, the authors to deploy the sensors at different intervals where some of the sensors initially deployed in the first step followed by another set of sensors and so on till the required detection probability is achieved. The algorithm presented in [14] entitled virtual force algorithm is used to reorganize the deployed nodes assuming that mobile sensors in the field to be monitored. The authors assumed two types of forces which are attractive and repulsive forces. The algorithm goes in rounds until sensor movement stability occurs followed by ordering the sensors to move to their decided positions. The same idea is extended in [13] where the network coverage holes are identified by Voronoi diagram after sensors initial random deployment. Mobile sensors that are close to the network holes move to cover these holes. As can be seen in [14] and [13], the deployment process was based on the assumption of mobile sensors are available; this assumption is hard assumption and it might not be practical in many of the sensor networks. Our work in this paper, assumes only virtual mobile sensors that are utilized only in the offline deployment.

Homogenous sensor deployment [15][14] could be efficient in some applications; however, recently some research is done on using heterogeneous sensors instead. Heterogeneous sensors, in this context, means sensors with different sensing ranges, communication ranges, and initial energy. The authors in [15], for example, introduced an optimal solution based on a math-

ematical formulation for the deployment to a set of homogeneous sensors based on a linear topology. One of the problems in linear topology is that sensors close to the sink node got depleted earlier. Another greedy algorithm is proposed based on the concept of sending the data as far as possible to reach the farthest neighbor.

Unfortunately, heterogeneous deployment problem was not taken that much attention. For instance, for full coverage deployment, the authors in [19] introduced a solution based on integer linear programming for heterogeneous sensor deployment. In addition, the authors in [16] proposed an optimal solution to heterogeneous sensor deployment for the purpose of minimizing the deployment cost when different communication modes are used. Their proposal clustered the sensors into a set of clusters taking into consideration single-hop, multi-hop, and hybrid of communication modes. Lee et al. in [17] investigated the effect of the heterogeneity on the network lifetime and aging process. The authors concluded that a mix of high and low capability sensors could extend the lifetime of the overall WSN. The same problem is handled in [20] with reconfigurable and reprogramming capability are used. The authors came to the same conclusion where mixing of low and high capability sensors prolongs the WSN lifetime. Along with the coverage and lifetime as objectives to the deployment, connectivity and effective routing is another objective for the WSN deployment. For instance, Zhou et al. [15] studied the k-connected coverage problem in which k sensors are to be connected for effective routing.

K-coverage problem is still a hot problem in the field of Wireless Sensor Networks. For instance, Peng et al. in [18] proposed an algorithm for an event K-coverage for under water sensor networks. It uses a mix of Pareto set and Genetic Algorithm. In [19], the authors use the K-Coverage probability in a finite wireless sensor networks. This is the inverse of finding the longest k-uncovered segment in the monitored field. They also considered the detection model of the sensor as a binary model, as we do in this proposal. The authors in [20], proposed both centralized and distributed protocols for the K-Coverage problem in WSNs. The proposed approach is based on Coverage Contribution Area where the sensors energy is taking into consideration during the deployment. A genetic algorithm is also used by the authors of [21] for the k-coverage problem. After deployment, we will be using a modified version of billiard algorithm [22] for measuring the longest k-uncovered line segment [23].

#### 4 PROPOSED SOLUTIONS

In this section, we describe our proposed solution algorithms for the given problem. We start by the billiard algorithm that it is inspired from [24]. Afterwards, we describe how K-Coverage is accomplished using a variation of Billiard algo-

rithm.

#### 4.1 BILLIARD ALGORITHM

The main idea behind the Billiard algorithm is handling the collision among moving objects. The algorithm consists of a sequence of steps as shown in Figure 1. As can be seen in Figure 1, the algorithm consists of five steps as follows:

##### STEP 1: SENSOR DEPLOYMENT

In this step sensors are deployed into the monitored field using one of two methods, deterministic or random. In both cases, sensors positions are captured. Sensors sensing level are assumed initially at the minimum level,  $p_1$ .

##### STEP 2: SHAKING PROCESS

After the initial deployment, sensors are shaken in which they are moved in certain speeds. Based on their positions and generated speeds, sensors will collide and move to other positions. The shaking process tries to limit sensors' overlap. Therefore, the algorithm is basically designed to reduce the overlap which serves our purpose of increasing the coverage.

##### STEP 3: COLLISION DETECTION PROCESS

The collision detection process is basically based on sensors overlap. The overlap between two sensors is determined by the distance between their centers compared to the current sensors radii. If the distance is less than the sum of the radii of the two sensors, then they are overlapped which means that the two sensors have a collision.

The collision of certain objects is handled using equation (1).

$$m_i x_i(t) = C(t, x_i(t), v_i(t)) x_i(0) = q_i, x_i(0) = p_i, i = 1, \dots, n. \quad (1)$$

Where  $x_i \in R^d (d > 1)$  and  $m_i$  are both position and mass of object  $i$ , respectively. Also,  $C(t, x, v)$  represents the object external force located at  $x$  with velocity  $v$  and time  $t$ . Any collision between any two objects such as objects  $i$  and  $j$  at time  $t_c$  is subject to equation (2).

$$\|x_i(t_c) - x_j(t_c)\| = r_i + r_j \quad (2)$$

Where the radii of both objects  $i$  and  $j$  are  $r_i$  and  $r_j$ , respectively.

In order to simulate the objects collisions, we assume that objects will move in straight lines after the collisions. The reason

behind this assumption is that, it simplifies the process of collision detection and to be able to determine the collision times before its actual occurrences.

##### STEP 4: COMPUTING SENSORS NEW DISPLACEMENTS AND POSITIONS

Computing the new displacements and positions is straight forward. As an example, let's assume that the positions of two objects  $i$  and  $j$  are given by equations (3) and (4).

$$x_i(t) = q_1 + v_1 t \quad (3)$$

$$x_j(t) = q_2 + v_2 t \quad (4)$$

Where  $q_1$  and  $q_2 \in R^d$  are the positions of both objects  $i$  and  $j$  respectively.

##### STEP 5: INCREASING SENSORS RADII

After each iteration, sensors radii are increased by  $\Delta$ , where  $\Delta$  is a constant identified at the deployment designer. This process is called "Radii Grows" or "Grows", for simplicity, making sure that sensors radii is not increased beyond its maximum sensing level  $p_n$ .

##### STEP 6: INCREASING SENSORS RADII

The algorithm stops in one of two cases which are either reaching the total number of iterations or when sensors do not move. When the algorithm stops, sensors sensing levels will be adjusted to the nearest sensing level.

#### 4.2 K-COVERAGE SOLUTION

For K-Coverage, there are few changes in the billiard algorithm. In the first step, we assume that the given number of sensors are enough to cover the monitored field  $K$ -times. This is possible through a good estimation to the sensors given their maximum and minimum sensing ranges. Therefore, in this step, sensors are divided into  $k$ -sets and each set is used for a separate run. We realize that for a given set, a point in the monitored field could be covered more than once. However, due to the limited space, this paper covers only the basic solution for  $k$ -coverage and another paper will be followed that contains a parallel version of the billiard algorithm as well as the parallel  $k$ -coverage version of the billiard algorithm. In addition, we will be presenting other variations of the billiard algorithms. Therefore, the only modification to the billiard algorithm to fit the  $k$ -coverage is a step before the deployment process, we call it "K-Set Division".

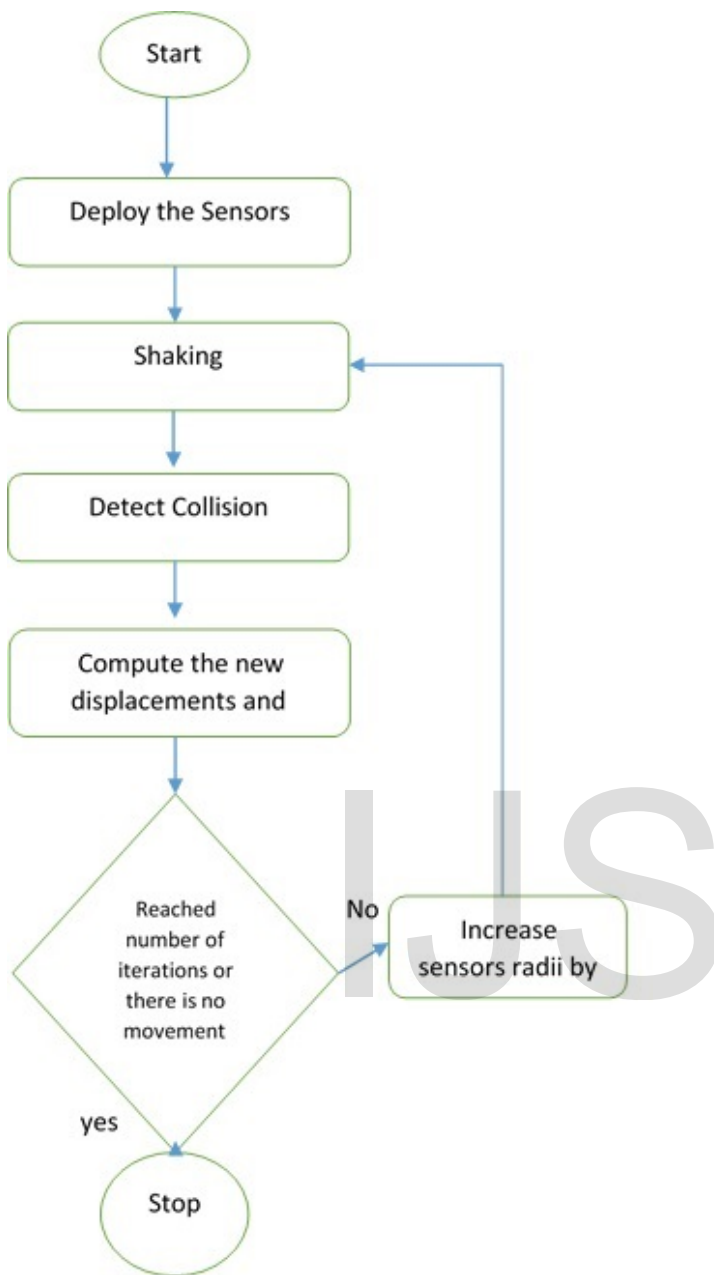


Figure 1. Sequential Billiard algorithm

In the following section, we try to experiment with both versions of the billiard algorithm and examine their performance.

## V. EXPERIMENTAL RESULTS

In this section, we experiment with the two previous algorithms for sensor deployment. A set of experiments are designed to test the performance of the previous algorithms including:

- Running time for full coverage
- Number of iterations for one coverage
- Expansion percentage level for one coverage
- Running time for full K-Coverage

### - Number of Iterations Vs. K-Coverage

A common hardware platform is used for all of the previously mentioned set of experiments. We utilize HP ProBook 450 G4 Notebook PC with Intel Core i5-7200U Intel HD Graphics 620, 2.5 GHz, 3 MB L3 cache, 2 cores, 500 GB (5400 rpm) hard drive, and 8 GB total system memory (4 GB x 2).

### A. Running Time for Full Coverage:

In this set of experiments, we tried with different problem settings to examine the time billiard algorithm takes for full coverage to the monitored field. The idea is to let the billiard algorithm run without identifying the number of iterations. The stopping criteria is changed to no movement/displacement to the deployed sensors. In other words, sensors having no collision any more. We base our experiment on trial and error as well as estimation to the number of sensors to fully cover an area. Since, we are basing our assumption on a circular/disk sensing range, certainly, full coverage where each point in the monitored field is covered will be impossible. However, a good approximation is to have the sensors distributed in the field where no more sensor can be added.

In this set of experiments, 1000m X 1000m monitored area with different sets of sensors, 400, 450, 500, 550, and 600. Sensors are initially deployed with four power levels and the maximum sensing range is assumed to be 50m. In addition, the expansion ration is set to 1m. As can be seen in Figure 2, the average time taken to fully cover the monitored field is 7200ms and the number of iterations is 10000.

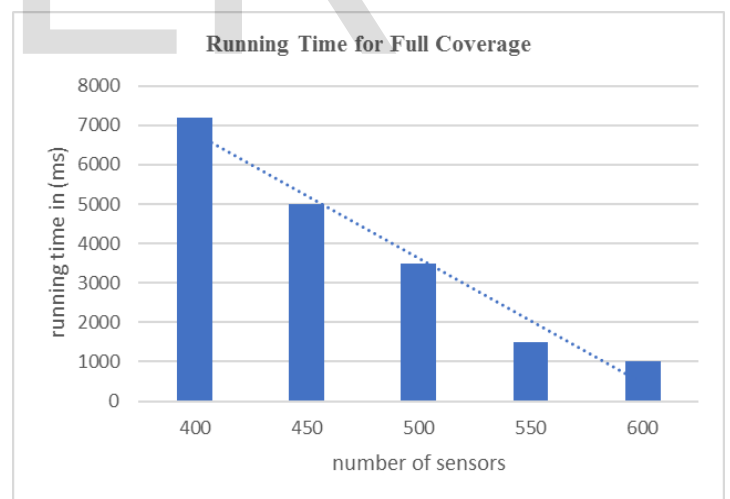


Figure 2. Running Time for Full Coverage

### B. Number of Iterations for One Coverage

In this set of experiments, we examine the required number of iterations for a full coverage. These set of experiments use the same settings used in the previous experiments. As shown in Figure 3, the number of iterations seem linear with different number of sensors.



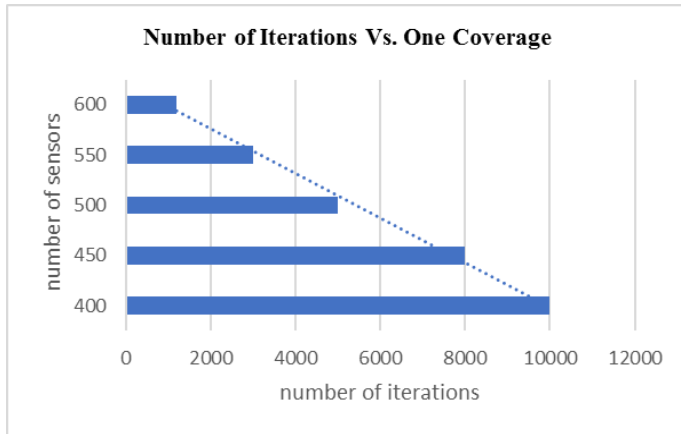


Figure 3. Number of iterations vs. for One Coverage

**C. Expansion Percentage for One Coverage**

Here, we examine the effect of the radii expansion percentage on the coverage percentage. The expansion percentage is varied from 1% of the maximum sensing level to 20%. The monitored area is assumed 1000m X 1000m and the number of sensors is assumed 550 sensors. 2000 iterations are used as a stopping criterion for this set of experiments. As shown in Figure 4, the average results over 10 runs for each case, show that the coverage percentage becomes much better if the expansion ratio. Is kept small.

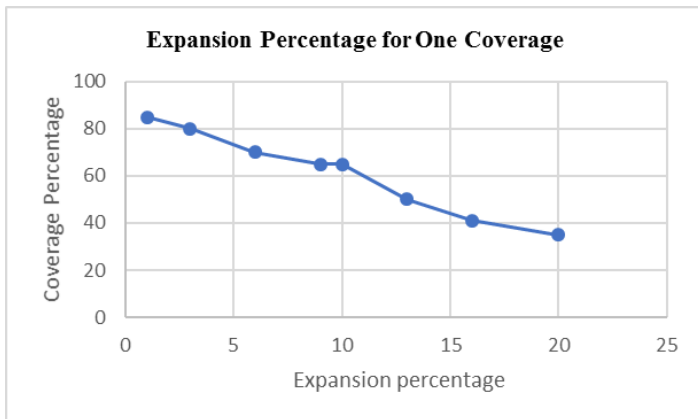


Figure 4. Expansion Percentage for One Coverage

**D. Running Time for full K-Coverage**

In this set of experiments, we examine the running time for the k-coverage problem based on the proposed algorithm. We use the same settings used in experiment A. However, K set to 2. In addition , the running time for the 2-coverage is compared

to the one coverage running time. As can be seen in Figure 5, the running time, in most of the cases is almost double the running time of the 1-coverage algorithm.

**VI. Conclusion:**

Sensor deployments are still a problem for most of the wireless sensor networks, especially for inaccessible places as well as when there is a large number of sensors to be deployed. In addition, K-coverage problem is much more complex than the 1-coverage problem. In this paper, we proposed a variant Billiard algorithm for 1-coverage and K-coverage deployment problems. Based on a number of test cases, we conclude that the billiard algorithm could be a suitable algorithm for both 1-coverage and k-coverage problems. However, the running time for both still large. Therefore, our future work will target the parallel billiard algorithm for both 1-coverage and k-coverage problems.

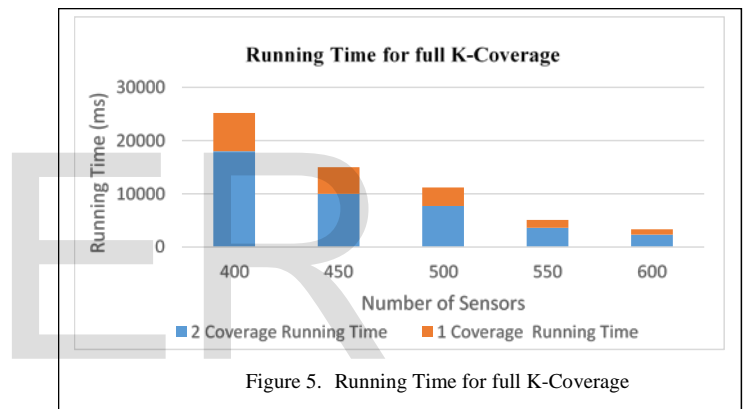


Figure 5. Running Time for full K-Coverage

**ACKNOWLEDGEMENTS**

This work is done under the grant received (400285/25/39-2017) by Deanship of research at Islamic University of Madinah (IUM) for applied research. We would like to give our regards and special thanks to the administration of IUM for their support in completion of this research.

**REFERENCES**

- [1] Jaja J (1992) An introduction to parallel algorithms. Addison Wesley, Redwood City, CA, USA.
- [2] Leighton FT (1992) Introduction to parallel algorithms and architectures: arrays, trees, hypercubes. Morgan Kaufmann, San Mateo.
- [3] Bondalapati K, Prasanna VK (2009) Reconfigurable computing: architectures, models and algorithms. Curr Sci 78:828-837
- [4] Gordon D, Koren I, Silberman GM(1984) Embedding tree structures in VLSI hexagonal arrays. IEEE Trans Comput C-33:104-107
- [5] Vaidyanathan R, Trahan JL (2004) Dynamic reconfiguration: architectures and algorithms, vol 13. Springer Series in Computer Science. Kluwer Academic/Plenum Publishers, New York

- [6] Subbaraman C, Trahan JL, Vaidyanathan R (1993) List ranking and graph algorithms on the reconfigurable multiple bus machine. In: Proceedings of international conference on parallel processing, (St. Charles, IL, August 1993), vol III, pp 214-247
- [7] Wang BF, Chen GH, Lin F (1990) Constant time sorting on a processor array with a reconfigurable bus system. *Inf Process Lett* 34:187-192
- [8] C. Huang, Y. Tseng, The coverage problem in wireless sensor network, *Mob. Netw. Appl.* 10 (4) (2005) 519-528.
- [9] Loukas Lazos, Radha Poovendran, and Jim Ritcey, "On the Deployment of Heterogeneous Sensor Networks for Detection of Mobile Targets ", to appear in the International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), April 2007.
- [10] S. Shakkottai, R. Srikant and N. Shroff, "Unreliable Sensor Grids: Coverage, Connectivity and Diameter," Submitted for journal publication, 2003.
- [11] Zhang, H. Hou, J. C. , "Is Deterministic Deployment Worse than Random Deployment for Wireless Sensor Networks?," INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 2006 ..
- [12] Guiling Wang, Guohong Cao and Tom La Porta. "Movement-Assisted Sensor Deployment," *IEEE Transactions on Mobile Computing*, Vol. 5, No. 6, pp. 640 - 652, June 2006
- [13] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces", *Proc. IEEE nfocom Conference*, pp. 1293-1303, 2003. X. Liu and P. Mohapatra, "On the Deployment of Data Backhaul Nodes for Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, April 2007.
- [14] V. Mhatre and C. Rosenberg, "Design guidelines for wireless sensor networks: communication, clustering and aggregation," *Ad Hoc Networks Journal*, Elsevier Science, vol. 2, pp. 45-63, 2004.
- [15] Jae-Joon Lee, Bhaskar Krishnamachari, C.C. Jay Kuo, "Impact of Heterogeneous Deployment on Lifetime Sensing Coverage in Sensor Networks," *First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON)*, Santa Clara, CA, October 2004.
- [16] Z. Zhou, Samir R. Das, Himanshu Gupta, "Connected K-Coverage Problem in Sensor Networks," *Proc. of the 13th Int. Conf. on Computer Communication and Networks (IC3N)*, Chicago, IL, 2004 .
- [17] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Transactions on Computers*, vol. 51, pp. 1448-1453, Dec. 2002.
- [18] Falko Dressler, Isabel Dietrich, "Lifetime Analysis in Heterogeneous Sensor Networks," *dscd*, pp. 606-6169, th EUROMICRO Conference on Digital System Design (DSD'062006)
- [19] Jiang, P., Xu, Y. and Liu, J. (2017) 'A distributed and energy-efficient algorithm for event K-coverage in underwater sensor networks', *Sensors (Switzerland)*, 17(1). doi: 10.3390/s17010186.
- [20] Afshang, M. and Dhillon, H. S. (2017) 'k-Coverage probability in a finite wireless network', *IEEE Wireless Communications and Networking Conference, WCNC*. doi: 10.1109/WCNC.2017.7925966.
- [21] Yu, J., Wan, S., Cheng, X. and Yu, D. (2017) 'Coverage Contribution Area based k -Coverage for Wireless Sensor Networks', 9545(c), pp. 1-14. doi: 10.1109/TVT.2017.2681692.
- [22] Elhoseny, M., Tharwat, A., Farouk, A. and Hassanien, A. E. (2017) 'K-Coverage Model based on Genetic Algorithm to extend WSN lifetime', *IEEE Sensors Letters*, 1(4), pp. 1-1. doi: 10.1109/ISENS.2017.2724846.
- [23] Gryazina, E. and Polyak, B. (2014) 'Random sampling: Billiard Walk algorithm', *European Journal of Operational Research*, 238(2), pp. 497-504. doi: 10.1016/j.ejor.2014.03.041.
- [24] Dash, D., Gupta, A., Bishnu, A. and Nandy, S. C. (2014) 'Line coverage measures in wireless sensor networks', *Journal of Parallel and Distributed Computing*. Elsevier Inc., 74(7), pp. 2596-2614. doi: 10.1016/j.jpdc.2014.03.004.
- [25] Bilal Al-Qudah and Rabie A. Ramadan: Sensor Deployment Using Billiards Algorithm, 4th International Computer Engineering Conference Information Society Applications in the Next Decade (ICENCO), 2008.